TECHNICAL UNIVERSITY MUNICH

TUM School of Life Science

Influence of the Standard Genetic Code on Overlapping Genes: A Study of Multiple Properties

Stefan Werner Wichmann

Vollständiger Abdruck der von der TUM School of Life Sciences der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

genehmigten Dissertation.

Vorsitzende:	Prof. Dr. Caroline Gutjahr
Prüfer der Dissertation:	1. Prof. Dr. Siegfried Scherer
	2. Prof. Dr. Dmitrij Frishman

Die Dissertation wurde am 02.03.2022 bei der Technischen Universität München eingereicht und durch die TUM School of Life Sciences am 23.06.2022 angenommen.

Table of contents

Zusammenfassung

Die Triplettstruktur des genetischen Codes in seiner Standardform (SGC) zusammen mit der Doppelstrangstruktur der DNA erlaubt es, dass Nukleotidsequenzen als drei unterschiedliche Aminosäuresequenzen auf jedem der beiden Stränge gelesen werden können, indem man die Startposition um ein oder zwei Nukleotide verschiebt, was man als alternative Leserahmen bezeichnet. Theoretisch kann dadurch in jedem Leserahmen einer Nukleotidsequenz ein Protein kodiert werden. Eine einzelsträngige Nukleotidsequenz kann damit drei unterschiedliche proteincodierende Gene tragen, die als überlappende Gene (OLG – OverLapping Genes) bezeichnet werden. Wegen der gemeinsamen Nutzung derselben Nukleotiden sind OLGs anfälliger gegenüber Mutationen und schränken sich gegenseitig in ihrer Kodierungsfreiheit ein, was möglicherweise zu einem niedrigeren Grad an Optimierung der Proteinfunktion führt. OLGs werden bei der Genomannotation oft nicht in Betracht gezogen mit Ausnahme von Viren, bei denen OLGs zuerst entdeckt worden sind. Man nimmt an, dass OLGs bei Viren mit ihren konstanten Kapselgrößen, welche nur schwer zu ändern sind, eine höhere Anzahl an codierten Proteine ermöglichen. Nichtsdestotrotz sind OLGs inzwischen in zahlreichen Prokaryoten und Eukaryoten gefunden worden, was die Frage aufwirft, welche Funktionen diese Konstrukte erfüllen. Das Ziel dieser Dissertation ist es, die Bedeutung von OLGs bezüglich des Ursprungs des Lebens, möglicher biologischer Funktionen und ihrer evolutiven Entstehung zu erforschen. Indem man den SGC mit evolutionär sinnvollen Alternativen vergleicht, kann man seine Optimierung in Bezug auf bestimmte Eigenschaften analysieren. Viele Eigenschaften wurden auf diese Weise bisher untersucht, wobei die bekannteste die Robustheit des SGCs gegenüber Mutationen ist. In dieser Arbeit werden verschiedene Eigenschaften des SGCs erforscht, welche die Entstehung von OLGs begünstigen. Dabei wurde aufgrund der Ähnlichkeit der Konservierung verschiedener Leserahmen eine neue Optimalität des SGC gefunden. Ein Modell zur Analyse des Sequenzraums wurde entwickelt, um die Balance zwischen Sequenzkonservierung und Kodierungsflexibilität zu beschreiben, die sowohl für die Entstehung als auch für die Erhaltung von OLGs notwendig sind. Schließlich wird die Schwierigkeit der evolutionären Entstehung von OLGs abgeschätzt, indem durch einen neuartigen Algorithmus künstlich erzeugte OLGs untersucht werden. Die künstlichen OLGs werden in silico aus zufällig gewählten Proteindomänen gebildet, um die durchschnittlich nötige Veränderung eines Gens für diesen Prozess abzuschätzen. Ausgewählte OLGs von verschiedenen Reportergenen werden experimentell auf ihre Funktionalität getestet.

Auch wenn die Abhängigkeiten von unterschiedlichen Eigenschaften des genetischen Codes eine Analyse erschwert, ist die Optimalität des SGC selbst eine robuste Eigenschaft. Als ein Beispiel wird die “Konservierung” alternativer Leserahmen, definiert als die durchschnittliche Effektgröße einer Mutation, untersucht. Die Ähnlichkeit verschiedener Leserahmen in dieser Eigenschaft legt die Deutung nahe, dass ein spezifischer Mittelwert zwischen Sequenzkonservierung und Kodierungsflexibilität optimal ist. In der Evolution wird der Sequenzraum durch Mutationen „abgetastet“, um entweder neue funktionale Sequenzen zu finden oder bereits bestehende funktionale Sequenzen zu optimieren. Ein einfaches Model, dass diesen Prozess simuliert, hat gezeigt, dass ein spezieller durchschnittlicher Einfluss einer Mutation die zu erwartende Fitness einer Sequenz in einer rauen Fitnesslandschaft optimieren

kann. Die Balance zwischen Erkundung und Konservierung ermöglicht es Sequenzen, niedrige Fitnessoptima zu überwinden und zu höheren Fitnessoptima zu gelangen.
Um die Schwierigkeit der OLG-Entstehung abzuschätzen, werden künstlich konstruierte OLG Sequenzen mit natürlich vorkommenden Homologen verglichen. Dabei zeigt sich, dass die nötige Veränderung von biologischen Proteindomänen zur Erzeugung von OLGs vergleichbar ist mit der Variation von Homologen in einer Genfamilie. Das gilt für Hidden-Markov-Model Wertungen, Aminosäureübereinstimmung und Ähnlichkeit, sowie Sekundärstruktur. Manche OLG Paare können erzeugt werden indem man nur 1.8% der Nukleotide in der überlappenden Region verändert und könnten damit durch zufällige Mutationen erreichbar sein.
Während Viren als taxonomische Gruppe mit den meisten OLGs gilt, zeigen sich Hinweise, dass pro- und eukaryotische Gene möglicherweise viel besser geeignet sind, um OLGs künstlich zu erzeugen. Trotz ihrer hohen Ähnlichkeit mit natürlich vorkommenden Genen konnte die Funktionalität von konstruierten OLGs bisher nicht experimentell nachgewiesen werden.
Die Ergebnisse dieser Dissertation weisen auf eine mögliche Rolle von OLGs in der de novo Entstehung neuer Gene hin, was auf einer inhärenten Eigenschaft des SGC beruhen könnte. Die Kodierungsflexibilität des SGC und die Flexibilität von genetischen Sequenzen ist ausreichend für die Erzeugung von OLGs und damit für die Evolution neuartiger Gene. Möglicherweise liegt hier eine plausiblere Hauptfunktion von OLGs im Vergleich zu Genomkompression bei Viren. Weitergehende Studien, die diese Hypothesen prüfen, könnten helfen, einen fundamentalen Aspekt der Evolution, nämlich die de novo Entstehung von Genen, zu verstehen.

Abstract

The triplet structure of the standard genetic code (SGC) and double-stranded nature of DNA together allow for nucleotide sequences to be read as three different sequences in each of the two strands, called reading frames, by shifting the starting position by one or two nucleotides or reading from the opposite strand. In theory a nucleotide sequence can encode a gene in every reading frame, which therefore use the same nucleotides. These constructs are called overlapping genes but no more than two genes encoded parallel to each other have been observed in nature. Due to their simultaneous use of the same nucleotides, OLGs are more susceptible to mutations and restrict each other in the coding flexibility, redistricting their possible degree of optimisation. For these two reasons, OLGs are often not considered in genome annotations outside of viruses, where OLGs were first discovered. In viruses OLGs are considered to facilitate a compression of the genome size, which is a limiting factor due to small capsule sizes. Nevertheless, OLGs have been found all over the tree of life, raising the question which functions these constructs fulfil. The aim of this study is to examine the theoretical foundation of OLGs regarding their importance for early life, possible functions and the difficulty of creating OLGs.

Comparing the SGC with evolutionarily sensible alternatives, the level of its optimisation can be determined regarding a chosen property. Many properties have been tested this way with the most prominent being the mutational robustness of the SGC. Here different properties of the SGC are studied with special focus on properties facilitating the existence of OLGs. Studying various methods of combining different properties into a single test, a new optimality of the SGC is found in its similarity between the conservation of alternative reading frames. A toy model of sequence space exploration is studied to determine the function of such a tradeoff value between sequence conservation and coding flexibility, which are both necessary to evolve and maintain OLGs. Finally the difficulty of evolving OLGs, which is a fundamental question for the existence of OLGs, is estimated by studying artificially created OLGs using a recently published algorithm. The artificial OLGs are designed using arbitrarily chosen protein domains to estimate the average change inflicted on a gene to create an overlap. OLGs constructed from different reporter genes are experimentally investigated for function.

While many difficulties are encountered in combining multiple properties in optimality tests due to interdependencies between different properties, the optimality of the SGC itself is found to be a very robust feature. As one example, we investigated the “conservation” property of alternative reading frames, defined as the average effect size of a mutation. The similarity of this property between reading frames appears to be optimal in the SGC, as if it were optimised for a specific tradeoff value between sequence conservation and coding flexibility. In evolution, mutations help explore sequence space to either find or optimise functional sequences. A toy model simulating this process showed that a specific average mutation step size can optimise the average fitness of a sequence in a rugged fitness landscape. The balance between exploration and conservation helps sequences to escape small fitness peaks and be conserved in larger ones.

Estimating the difficulty of evolving OLGs by comparing artificially constructed OLGs sequences to naturally occurring homologs shows that the necessary change to natural protein domains inflicted by constructing OLGs is on the same level as variations between homologs within a

gene family. This is true for Hidden-Markov-Model scores, amino acid identity and similarity, and secondary structure. Some OLG pairs can be created by only changing 1.8% of the nucleotides in the overlapping region and are therefore accessible through random mutations. While viruses are thought to be the most likely taxonomic group to carry OLGs, eukaryotic genes are in our analysis actually much more suited for designing OLGs. Despite their high similarity, the function of the constructed OLGs could so far not be verified in experiments, but which are not conclusive yet.

The results hint at a possible role of OLGs in *de novo* gene creation which is also indicated by the SGC. The coding flexibility of the SGC and the flexibility of genetic sequences is sufficient to enable creation of OLGs. This is a much more plausible central function of OLGs compared to genome compression, which is reinforced by viruses genes being much less suitable for creating OLGs compared to other taxonomic groups. Further studies challenging these hypotheses could help understanding this essential aspect of life - *de novo* gene creation.

1. Introduction

The standard genetic code has been an object of fascination since it was discovered soon after the discovery of the molecular structure of the means of inheritance. It has been studied from various angles but continues to reveal new insights. Here I specifically study the standard genetic code in relation to overlapping genes. Historically, the genetic code was the last missing piece to understand the century old question of inheritance, as necessary for Darwin's theory of evolution [1] and first observed by Mendel in 1865 [2], on a molecular level. The “genes”, which were thought as a unit of information that can be inherited to offspring first, were later found to be contained in the deoxyribonucleic acid (DNA) molecule [3]. With the discovery of ribosomes [4], their function [5] and transfer ribonucleic acids (tRNAs) [6], [7], which carry an amino acid (AA) and bind to the ribonucleic acid (RNA) inside the ribosome, the translation of RNA to protein was understood. Finally, the full deciphering of the genetic code [8] completed the molecular basis of inheritance, namely molecules that carry the genetic information and a way to express the information in proteins that in turn express a certain trait in the organism. The details on which parts of the DNA are genes, which genes are expressed and how the expressed genes interact with each other is still not completely understood [3].
Even though the genetic code was fully deciphered in 1963 there is still a lot of current research on it. Searching google scholar for “standard genetic code” results in 3.000.000 entries of which 264.000 entries are from the last 20 years. Overlapping genes (OLGs), the second topic of this dissertation, on the other hand have been known to exist in viruses since 1976 [9] but only 26.200 entries can be found on google scholar of which 17.500 are from the last 20 years making it a much younger and less prominent field than the genetic code. OLGs are only possible due to the structure of the genetic code and both are therefore strongly linked. In this dissertation I explore the relationship between the genetic code and OLGs in more detail in order to better understand the strong relation of the two - this study has potentially important implications for understanding the origin of the code, the evolution of new proteins, and for synthetic biology.

1.1. The Genetic Code and its Importance for Life

The basic functional building blocks of a cell are the DNA and various cellular organelles, which differ depending on the species. While organelles are vital for creation of a new cell in e.g. cell division, all proteins used to build the organelles are encoded in DNA, either the organelles’ own DNA as in the mitochondria and chloroplasts in eukaryotes or the chromosomal DNA, arguably making the DNA the most central structure of life as we know it. In a sense the DNA is the hard disk of life containing its information written with the nucleotides adenosine (A), cytosine (C), guanosine (G), and thymidine (T) (or uridine (U) in RNA) just as the information on a hard disk in a computer is written with 0s and 1s. The translation from nucleotide sequences to AA sequences (polypeptides), which subsequently fold into proteins, is dictated by the genetic code. This has been coined the central dogma of molecular biology by Francis Crick in 1958 [10] even before the details of the genetic code were known, and still holds true today with only minimal refinements [11] despite different challenges [12].

Without the genetic code only RNA molecules would be possible as functional information rich macromolecules so the code opens up the many possibilities of the protein world for life. Thus, unsurprisingly, the genetic code is very old and probably had the same structure (i.e. mapping between codons and AAs) in the last universal common ancestor (LUCA) as today. Since it is the same in most organisms [13], it is called the standard genetic code (SGC). The various known exceptions to the SGC appear to be much younger (i.e. derived from the SGC), on account of being highly taxonomically restricted, and differ only very little compared to the SGC [13].

1.1.1. Structure of the SGC

The codon to AA translation of the SGC is summarised in Fig. 1.1 and it has multiple layers of conceptual structure. The most basic is the set of elements that can be coded for, which includes 20 different AAs in the SGC. This set appears to be highly nonrandom as it is highly evenly distributed across three important chemico-physical properties: size, charge and hydrophobicity [14], [15]. Every tRNA binds to a nucleotide triplet, which is called a codon, so with four different nucleotides the SGC consists of 64 potentially different codon to AA bindings. Since the SGC only contains 20 different AAs plus a termination signal marking the end of a gene, which stops translation in the ribosome and is therefore also called a stop codon, some have to be coded for by multiple codons. This degeneracy is the second layer of structure of the SGC, namely how many codons encode each amino acid. In the SGC this varies from only one codon for Methionine to six codons for Leucine, Arginine and Serine. The third layer of structure consists of which specific codons code for which AA, since codons can be related in different ways, which in turn determines the evolutionary relationships between the AAs they are coding for. One such connection between codons is due to mutations in DNA/RNA and misread errors in the ribosome, for example due to faulty bindings of tRNAs [16]. In case of single mutations or misreads, this connects codons which differ only by one nucleotide. The SGC often codes for chemically similar AAs in these ‘one step distant’ codons [17], which results in mutations and misread errors having a reduced influence on the final protein. Most of the second and third layer of structure is due to the wobble binding rules [18], [19] of tRNAs, see Table 1.1. These rules originate from the observation that the third nucleotide in the codon-anticodon binding of the tRNA to the RNA inside the ribosome does not always have to be the exact nucleotide base pairing but can have some more freedom. This leads to a block-like structure of the SGC in which the first two nucleotides matter more than the third. In 8 out of 16 possible combinations of the first two nucleotides the third nucleotide does not matter as all four possibilities code for the same AA. Of the remaining 8 blocks 6 only distinguish between a pyrimidine (C, T/U) and a purine (A, G) bases and 2 further distinguish between the two purine bases while the pyrimidine bases still code for the same AA. Consequently, tRNAs for each codon are not needed and 31 different tRNAs are sufficient to create the SGC, two for each block but only one for the UAN block with N being any nucleotide as stop codons do not need tRNAs. For example, a study of 11 very different eukaryotes from yeast to humans found between 41 and 55 tRNAs with different anticodons [20]. Despite the Wobble binding rules and a surplus of tRNAs, the codon to AA translation is unambiguous, in the absence of errors, so the SGC is a true code [21].

Table 1.1: The wobble binding rules. Possible binding partners are marked with an “x”. Besides adenine (A), guanine (G), cytosine (C) and uracil (U), a fifth nucleotide, namely idenine (I) can be found in tRNAs and mRNAs resulting from a post transcriptional modification of adenine [22].

mRNA \ tRNA	U	C	A	G	I
U			X	X	X
C				X	X
A	X				X
G	X	X			

The structure of the SGC creates strong differences in the biochemical effects on the AA level for mutations in different nucleotide positions of the codon. The third nucleotide position is the least affected position due to the wobble binding rules. The first position is slightly less affected than the second position due to mutationally close AAs being more similar [23], [24].

	U		C		A		G	
U	UUU	Phe	UCU	Ser	UAU	Tyr	UGU	Cys
	UUC		UCC		UAC		UGC	
	UUA	Leu	UCA		UAA	STOP	UGA	STOP
	UUG		UCG		UAG	STOP	UGG	Trp
C	CUU	Leu	CCU	Pro	CAU	His	CGU	Arg
	CUC		CCC		CAC		CGC	
	CUA		CCA		CAA	Gln	CGA	
	CUG		CCG		CAG		CGG	
A	AUU	Ile	ACU	Thr	AAU	Asn	AGU	Ser
	AUC		ACC		AAC		AGC	
	AUA		ACA		AAA	Lys	AGA	Arg
	AUG	Met	ACG		AAG		AGG	
G	GUU	Val	GCU	Ala	GAU	Asp	GGU	Gly
	GUC		GCC		GAC		GGC	
	GUA		GCA		GAA	Glu	GGA	
	GUG		GCG		GAG		GGG	

Figure 1.1: The standard genetic code. The colour scheme groups AA by their properties, namely small and nonpolar (*orange*), hydrophobic (*green*), polar (*pink*), negatively charged (*red*) and positively charged (*blue*). The full names of the AAs can be found using table 1.2.

Table 1.2: AA one letter and three letter symbols. Only the AAs included in the SGC are listed here.

A	Ala	Alanine	M	Met	Methionine
C	Cys	Cysteine	N	Asn	Asparagine
D	Asp	Aspartic acid	P	Pro	Proline
E	Glu	Glutamic acid	Q	Gln	Glutamine
F	Phe	Phenylalanine	R	Arg	Arginine
G	Gly	Glycine	S	Ser	Serine
H	His	Histidine	T	Thr	Threonine
I	Ile	Isoleucine	V	Val	Valine
K	Lys	Lysine	W	Trp	Tryptophan
L	Leu	Leucine	Y	Tyr	Tyrosine

1.1.2. Properties of the SGC

The structure of the SGC gives rise to different properties the SGC excels in, regarding the biological usefulness of its arrangement. This can only be measured relative to alternative ‘possible’ genetic codes, which are usually artificial codes not realised in nature. Depending on the approach these artificial codes reflect either what is possible in a search for an optimum or what is likely by randomly selecting a set of codes under some restrictions. These properties could be an important contributing factor in the subsequent evolution of life as for example a sufficient amount of error correction is needed in translation such that larger and more complex proteins can exist [25], which build the foundation of life as we know it. If the SGC did not have any error correction properties evolution might have stopped at simpler life forms so this property could partly explain why complex life is possible. Properties of the SGC could also give hints regarding its evolution as it could have acquired advantageous properties through selection amongst competing alternative codes. This hypothesis becomes especially interesting if the properties of the SGC are rare compared to a set of artificial codes reflecting plausible alternatives according to a possible path of evolution for the SGC. Assuming that the SGC has rare and advantageous properties just by chance is a possible but scientifically unsatisfactory explanation. From the point of view of selective evolution via random mutations in the SGC, rare properties in a set of alternative codes reflect an optimised (selected) value even if they are not a local or global maximum, since better codes become increasingly less likely to be found by chance, slowing down evolution. That is, a selection hypothesis predicts some degree of optimization, but not necessarily a global optimum. Therefore properties which are rare are called ‘optimal’ or ‘optimised’ in this study. A maximum of 5% of codes in the comparison code set performing better in a measure has been used as a cutoff for optimal properties in the

literature [26], but is an arbitrary threshold. In the following the remarkable range of different properties of the SGC that have been found to be optimal in this sense are reviewed. Whether they have really been subject to selection is a separate question.

1.1.2.1. Mutational Robustness

The most prominent property is the mutational and misread error robustness of the SGC [27], which brings a clear advantage to any organism as faulty translation not only wastes resources but the resulting protein can also be toxic to the cell. It is defined as the average change a mutation infers on the AA coded for by any mutated codon. The differences in AAs can be defined in two different approaches: either by physicochemical differences in the AAs or a statistical approach determining how often each AA is exchanged by each other in naturally occurring homologs. While any or any combination of physicochemical properties can be used (see see Table 4 of [26] for a summary), one common measure is polar requirement [28], [25], [29], which measures the mobility of AAs in a water-nucleobase-solution with paper chromatography and is similar to the AA's hydrophobicity. Alternatively, a statistical approach uses AA exchange matrices (e.g. the BLOSUM62 matrix [30]), which are thought to be a measure for how similar AAs are in real proteins [31] as it is not clear which physicochemical properties actually matter for evolution. While neither of these approaches can resolve the dependency of AA similarity on specific genes and positions in those genes, the statistical approach additionally suffers from influences of the SGC, as mutationally close AAs will be replaced with each other more frequently; this effect can be reduced by only using more distant homologs but not removed entirely.
Results of optimality analyses rely strongly on the restrictions on the alternative codes which serve as a comparison set, with the probability to find a better code than the SGC in the property of mutational robustness ranging in published analyses from 1 in 5 [32] to 1 in 10^8 [33]. But even for alternative codes which also follow the wobble binding rules and have the same degeneracy on the third nucleotide in the codon, the SGC is as rare as one in a million [34]. In approaches searching for the best possible code for mutational robustness using genetic algorithms, the SGC is far from the best codes found [35] and it is not even a local maximum [36]. From an evolutionary point of view this is not surprising, however, as will be discussed below after introducing some different evolutionary hypotheses.

1.1.2.2. Frameshift protection

Besides mutations and misreads, a shift of the ribosome on the translated messenger RNA (mRNA) is another possible error leading to faulty translations. If the ribosome shifts by a multiple of three nucleotides, one or more AAs are left out of the protein, but the protein still has a real chance to be functional as it consists of hundreds of AAs and removing a few is not a big change. On the other hand if the shift does not consist of a multiple of three a completely different chain of AAs is produced by the ribosome and usually renders the translated protein non-functional or even toxic. The same kind of frameshift appears for deletion or insertion mutations. While a mutation or misread at most changes a single AA in the gene, a frameshift changes all codons downstream of the error and is thus far more deleterious for an organism. A possibility to reduce energy and therefore fitness costs [37] of such an event, the genetic code could be designed in such a way that frequently used codons create stop codons when

frame-shifted in any way. This would stop the faulty transcription shortly after the frameshift and save resources for the organism. This property does not only depend on the code but also the codon statistics used in the genome and is therefore harder to maintain across different organisms with different genome compositions. Nevertheless this property, as instantiated in the SGC, has been found to be rare among the set of codes maintaining the mutational error robustness of the SGC [38], [39].
A different approach to solve the problem of frameshifts is to code very similar proteins in the frame-shifted reading frames. This seems almost impossible but has surprisingly been found to be true for the hydrophobicity structure of genes, which correlates strongly between the original gene and the frame-shifted sequences, in a recent study [40]. While the frame-shifted sequences are not the same gene, they could be similar enough in their physicochemical properties in order to maintain function. This only makes sense if the frame shifted reading frames rarely contain stop codons, otherwise the gene is only partly translated. Since stop codons are very common in alternative reading frames [39] and the conservation of AA properties in the frameshifted codons correlate strongly with the mutational robustness [41], it is more likely that the similarity of the frameshifted reading frames has a different function. In [40] it is hypothesised that this could be used to explore variants of the original gene to find a more optimal sequence [40].

1.1.2.3. Finding functional sequences by random mutations

In order to optimise genes, random mutations must first find functional variants of an existing gene before the best variant can be selected. The fundamental difficulty of finding functional sequences is the astronomical number of different sequences and the rarity of functional ones. There are more possible sequences of 62 AAs using only the 20 AAs included in the SGC, than the number of atoms in the observable universe (~10^{80}), as estimated from the cosmological parameters determined by the Planck Collaboration [42], while it has been estimated that perhaps only as few as 1 out of 10^{77} sequences have a particular enzyme function [43]. There are various other estimates available, with some folds being much more accessible but still very rare (e.g. the small WW domain is constituted by perhaps one out of $2.9 \cdot 10^{24}$ sequences [44]) and some perhaps even rarer in sequence space. It is questionable whether the time since the formation of earth ($4.55 \cdot 10^{9} a$) [45] or even the age of the universe ($13,7 \cdot 10^{9} a$) [46] would be enough for us to expect to find particular functional domains by random mutations. Our own laboratory experiments, as described in chapter 5, support this general picture of the rarity of specific functions, as despite careful design of new sequences with the intention of matching a known Pfam domain, using highly informative restrictions on the sequence space, we could not successfully create functional proteins with our initial attempts.
To solve this problem, biological mechanisms to accelerate finding of new genes are likely needed. Since it is very hard to study this problem due to the small probabilities, simpler but similar systems have been studied. In [47] exploration of combinations of mutations in 4 AA positions of a gene was studied using different genetic codes. Of the 194,481 different AA combinations including stop codons only 1659 are functional but mutations have to scan 16,777,216 nucleotide sequences to find them. How many of the nucleotide sequences translate to the small set of functional AA sequences depends on the genetic code. Starting from a functional sequence, the structure of the SGC has been shown to be optimal for finding

functional protein variants by random mutations for intermediate time scales [47]. This is very surprising as higher sequence space exploration would be associated with higher impact of each mutation, which reduces the mutational error robustness. A trade-off between the two properties has been hypothesised before [48] and could partly explain this observation.

1.1.2.4. Facilitating coding of additional information

DNA and RNA carry much more information than just the genes, like binding sites for regulatory proteins, histone binding sites, splicing signals and ribosome binding sites (RBS) to just name a few. Sometimes it is necessary or advantageous to encode this information alongside a protein sequence. Testing whether the SGC facilitates this possibility well has shown that it is exceptional at including random amino nucleotide n-mers parallel to an existing sequence on the same strand without destroying it by stop codons [38].

1.1.2.5. Using the antisense strand for proteins

The antisense strand to an existing gene in the DNA is similar to a frameshift as it usually translates to a completely different AA sequence. Interestingly, these sequences mostly have a complementary hydrophobicity profile compared to the sequence on the sense strand [49], which could lead to special interactions of the two proteins if both sequences are translated [50]. The reading frame on the antisense strand which is directly complementary to a ‘reference’ coding sequence (-1 frame) is especially well conserved if synonymous mutations in the reference sequence, i. e. mutations that do not change the amino acids, are considered [51]. Such a case of actual bidirectional coding would be called an OLG, which is introduced in detail later.

1.1.3. Evolution of the SGC

In order for any feature to be expected to be maintained after it evolved by chance, it needs to offer a selective advantage for its replicating system. The genetic code can only bring a selective advantage if the translated proteins do, which means that functional genes encoded into DNA or RNA must exist as well as the translation system. Therefore the evolution of the SGC is strongly connected to the evolution of genes and a translation system. Since neither of the three systems can bring any selective advantage without the other two,they must have evolved simultaneously. For this reason, the evolution of the SGC has been labelled a ‘notoriously difficult problem’ in 1976 [52] and remains so today [53]. Forming hypotheses for the evolution of any of those complex and sophisticated systems independently is very challenging, so most focus only on one system. Therefore most hypotheses on the evolution of the SGC assume that genes exist and can be translated, so a reasonably sophisticated organism or replicator carries the SGC.

In this case, a change in the translation of any codon in the SGC would impact every gene carrying this codon, which most likely is expected to cause an overall deleterious effect, given the rarity of functional sequences. Further, all else remaining equal, it can be assumed that more complex organisms have larger genomes and that code evolution becomes progressively more difficult with increasing system complexity [54]. As a consequence it has been thought that the SGC was a ‘frozen accident’ [55] since it cannot change after organisms reach a limiting

level of complexity. This would also explain why the SGC is so universal across the tree of life, if all living organisms descend from this early system with the ‘frozen’ code. However today we know that variants of the SGC exist, so the genetic code can change in at least minor ways. One theory is that some codons might be very rare in the genome and could therefore be changed much more easily [56]. Another, separate, explanation for the universality of the SGC assumes that horizontal gene transfer [57] is an essential mechanism for early life and yields a strong fitness advantage [58]. Different organisms would have been required to have the same genetic code in order to exchange genes and they showed that such a system eventually leads to a single genetic code [58]. So there are at least two good arguments why the evolution of the SGC stopped - either due to horizontal gene transfer or a too large genome size and complexity. The evolution before such a ‘freeze’, however, is still strongly debated.

The most popular theories for the evolution of the SGC are the stereochemical hypothesis, stating that stereochemical interactions determine the SGC; the coevolution hypothesis, stating that AAs were added to the SGC as they became biochemically available to the organism due to pathways of biosynthesis; and the optimization hypothesis**,** which suggests that the code was optimised for certain properties. While none individually gives a satisfactory explanation for the origin of the SGC, they do not exclude each other and could be different forces that acted on the SGC simultaneously or in different stages [59]. Further below, I will briefly summarise the arguments for and against these three hypotheses as they are used to define the restrictions for hypothetical alternative genetic codes in the optimality calculations of this study.

An evolutionary hypothesis for the SGC, which also takes the formation of genes into account, assumes that translation was ambiguous in the beginning with the ambiguity being reduced in the course of evolution [60]. The proteins translated from genes are then ‘statistical proteins’ [61] as no exact translation is possible. Some of these proteins are probably deleterious and some advantageous for the replicator, but if the produced proteins have a net positive effect the genetic code can be selected for. In this scenario, changing the code would shift the rates of each protein produced from a gene and would not have such a strong deleterious effect as in theories assuming an unambiguous translation. The production of statistical proteins would also strongly increase the number of explored proteins compared to the scenario of an exact translation in which each protein needs its respective genetic sequence so genome size is a strong limiting factor. The problem of finding functional sequences would be a little less severe in the ambiguity-reduction theory.

A feature of the SGC that in my view most likely never changed was the number of nucleotides in each codon (despite some theories assuming otherwise [62]). Changing the number of nucleotides in a codon would probably destroy start and stop signals of genes, change their length and affect each AA in the gene, thus such an impact would most likely destroy every gene. Since a genetic code cannot be selected for without useful genetic material, it is more likely that the structure of the ribosome or its precursor defined how many nucleotides are read at a time. If this is true, an interesting consequence is that overlapping genes, investigated later in this thesis, were possible from the beginning of the code.

1.1.3.1. The Stereochemical hypothesis

The most straightforward idea is that the codon to AA mapping of the SGC is due to a direct stereochemical interaction between nucleotides and amino acids [29]. While this is not true

today as the anticodon on the tRNA that binds to the mRNA is independent from recognition by the tRNA-synthetase, it could have been true before tRNAs existed, with this role later taken over by tRNAs. While aptamers can be artificially evolved by introducing codons and anticodons at the right positions to bind at least 11 out of 20 different AAs [17], all tRNAs that do have an unexpectedly high amount of codons or anticodons in their AA binding site [63] are not prebiotically available or have high energy costs in synthesis [64]. This further solidifies that the SGC is not determined stereochemically and other forces are needed to explain its modern form. The stereochemical hypothesis has been proposed to give a starting point for a primordial genetic code that can then be expanded or optimised a different way [65]. This common suggestion is problematic however in light of the above observation that the amino acids most likely to have been available early (less energetically expensive and/or prebiotically available) show no stereochemical affinities with their modern codons or anticodons - a hypothetical completely different early code does not help much in explaining the code's modern structure.

1.1.3.2. The coevolution hypothesis

The observation that only a few AAs can be formed in the hypothesised conditions of the abiotic Earth [66] as well as findings of a AAs in meteorites [67], [68], led to the hypothesis that in the beginning only a few AAs of the 20 in the standard code were part of the genetic code [69]. This hypothesis states that the genetic code only coded for four different AAs in the beginning, which were distinguished only by the first nucleotide of the codon. Whenever new AAs could be synthesised by the organism they were introduced into the code by consecutively splitting the existing blocks into subgroups determined by the second and third nucleotides in the codon. From alternative genetic codes observed in nature it can be seen that deviations from the SGC are due to small mutations in the tRNAs [13]. Assuming a similar process for the coevolution hypothesis, primordial, tRNA-like adapter molecules could mutate and bind physicochemical similar AAs. This way new AAs would always bind to neighbouring codons of similar, already existing AAs. This theory could partly explain the mutational error robustness in a stereochemical way, since it explains why AAs with similar physicochemical properties are in a close mutational distance as well as the degree of degeneracy in the standard code. But only in a very specific order of adding new AAs at specific codons is the observed mutational robustness fully explained [32]. If the first tRNA-like adapter molecules formed before the genetic code, this theory could explain the beginning of the SGC without the need of direct codon to AA binding as suggested in the stereochemical hypotheses.

1.1.3.3. The Optimization hypothesis

The many observed biologically relevant properties of the SGC strongly suggest that the SGC has been optimised before it reached its current form. Selection is the only known reliably optimising force, and positing this mechanism implies that different genetic codes existed and competed in the same environment. The organisms must have been very simple, otherwise the constant change in their genetic material due to changes in their genetic codes would stop evolution as discussed above. The number of generations needed in order to get a code as good as the SGC strongly depends on the restrictions on alternative codes [70], besides typical evolutionary factors, e.g. population size. Assuming no restriction on genetic codes in their

evolution but the 20 AA used today and stop codons results in $4.2 \cdot 10^{84}$ different genetic codes of which only about 10^{50} have a similar degeneracy structure on the third codon position as the SGC [71]. Finding one of such codes, not even one that is as optimised as the SGC among the codes with the same block structure [71], by chance is very unlikely and only worsens the problem of the unlikely code. This theory can therefore not explain the structure of the SGC alone and needs restricting factors on the evolution of the SGC. Using a combination of the stereochemical and a special case of the coevolution hypothesis, resulting in the 2-1-3 model [72], can explain the mutational and misread robustness of the SGC without the need of optimization but any more general model yields a strong optimality of this property [34]. Other properties have not been considered in this model yet so it is not possible yet to rule out the optimization theory in this model completely.

1.2. Overlapping Genes

In the SGC three nucleotides form a codon, which is translated into an AA. As a result each sequence of nucleotides can be translated to at least three different AA sequences by shifting the start of the first codon by one or two nucleotides. If this happens unintended as described above in the section about frameshift errors, this is highly deleterious as most of the AA sequence will be changed. But this property can also be useful, since each nucleotide sequence can contain up to three different genes at the same locus and read in the same 'frame' i.e. triplet sequence with the same offset relative to an arbitrary point. If two genes share nucleotides in the genome, they are called overlapping genes (OLGs) and can either overlap partially or with one gene completely embedded into the other (c.f. Fig. 1.2). Depending on the number of shared nucleotides, also called the overlap length or size, it is either a trivial or non-trivial overlap. Trivial overlaps have a length of less than 90 nucleotides [73], and can arise as the result of a mutation in the stop codon of a gene, which extends it to the next stop codon downstream lying inside another gene. This is an example of a general phenomenon of 'overprinting' [74] and adds some AAs to the first gene which are not part of its functional domains at first and often does not impair function. A non-trivial overlap on the other hand is longer than 90 nucleotides [73] and can potentially include essential parts of both genes. Since the average open reading frame (ORF) length, which is usually defined as the distance between two stop codons [75], is very short in overlapping regions if they are taken as a chance event, long overlaps that are maintained in the genome must have some function otherwise they would be lost in the course of evolution. While the majority of overlaps are trivial [76], very long and non-trivial overlaps have been found [77], [78]. When speaking of OLGs here, it is implicitly meant that it is a non-trivial overlap if not stated otherwise.

The first OLGs were discovered in the bacteriophage φx174 [9], [79]. While it is often assumed that OLGs only exist in viruses, today, besides viruses [80], [81], [82], [83], many OLGs are known [84],[85] in prokaryotes [86], [87], [88], [89], [90], [91], eukaryotes [92], [93] and highly complex organisms like vertebrates [94], [95] to only name a few examples. While most known OLGs seem to be young [83], two classes of aminoacyl-tRNA synthetases can be encoded in an almost fully overlapping manner [96], [97], [98], which would be very unlikely as a chance event, and these two classes of aa-tRNA synthetase are therefore considered ancient OLGs.

So, it appears that life has most likely always been using OLGs or at least as far back as the last universal common ancestor.

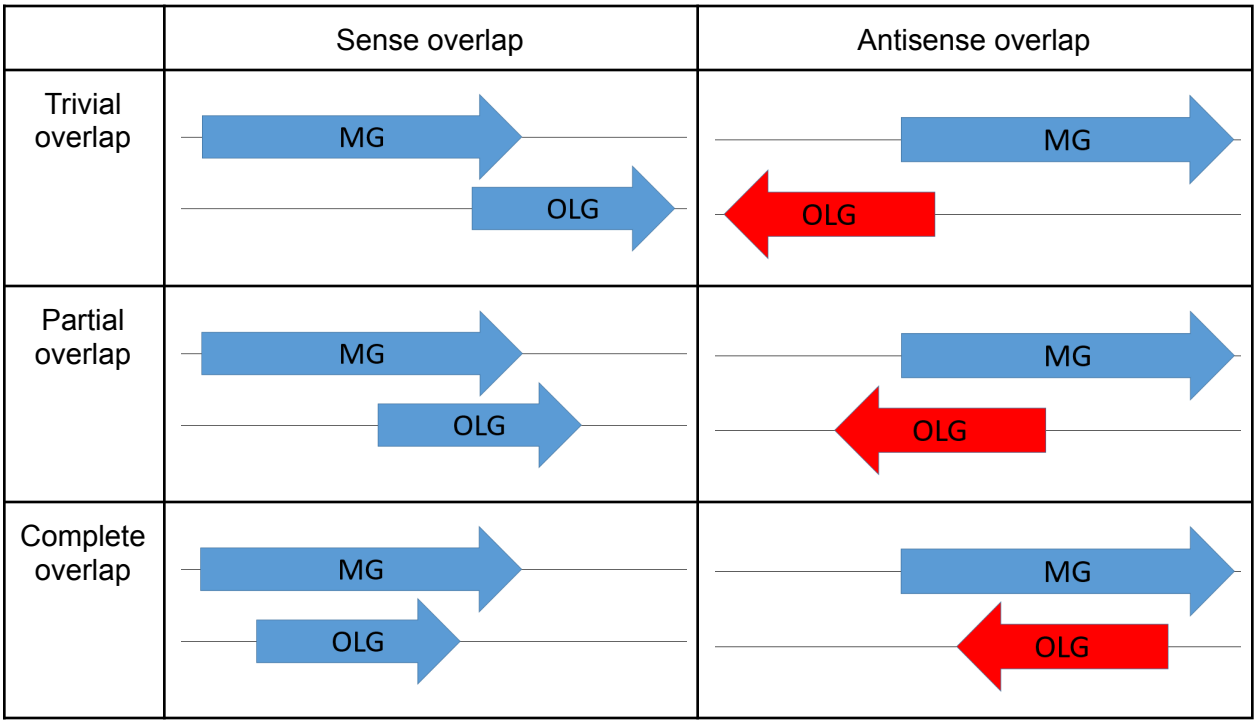

Figure 1.2: Different types of overlaps. Trivial overlaps share only very few nucleotides between MG and OLG. Partial overlaps share a longer sequence of nucleotides but both genes have a non-overlapping part. In complete overlaps one sequence is completely embedded into the other gene.

Despite the vast amount of proof for the existence of OLGs, NCBI does not annotate prokaryotic OLGs without individual justification [99]. This is presumably at least partly due to the perceived evolutionary difficulty of developing and maintaining OLGs by an organism. A nucleotide region that is already used in an existing gene, also called the mother gene (MG), is restricted in its changeability if the gene is not to be lost. This makes it more difficult to encode another gene at the same position. In the case that an OLG pair could be formed nevertheless, mutations in that region will impact two genes at once and are therefore something like twice as deleterious as normal mutations on average, if both genes are functional. There is also an energetic cost, and hence also fitness cost, to expressing any protein [37], [100]. Thus, OLGs bring along a fitness burden to the organism subject to mutations and should be removed by selection if this is not counteracted by some advantageous functions. In viruses this function is often thought to be genome compression as viruses have a limited genome size due to their envelope [101],[102] - although this is disputed [103], while no consensus is reached for non-virus organisms. For this reason many genome annotation programs exclude OLGs from the start [104] thus missing out on an old and universal feature of life.

1.2.1. Reading Frame Properties

An OLG can be shifted by one or two nucleotides relative to the MG if they are on the same DNA strand, so including the antisense strand a total of five alternative reading frames exist ‘parallel’ to every gene in the DNA. Due to the structure of the SGC, the alternative reading frames differ strongly in their properties. In order to speak about them, the naming definitions in Fig. 1.3 are used, where the ‘+1’ frame refers to the MG, the ‘+’ frames to the reading frames on the same strand as the MG and the ‘-’ frames to the antisense strand of the MG.
Perhaps the most interesting property varying across different reading frames is their flexibility of encoding different sequences without changing the MG. This is a result of the degeneracy structure of the SGC, for example all codons that code for Valine have the same first two nucleotides but the third is arbitrary, which translates to a fixed first nucleotide and an arbitrary second nucleotide in the ‘+2’ frame. This results in different numbers of combinatorial restrictions in the different reading frames [105]. The ‘-3’ has no specific restrictions and is therefore the most flexible followed by the ‘+2’ and ‘+3’, which both have the same but reversed restriction [105]. The ‘-2’ frame is the most restricted with nine restrictions and the ‘-1’ has four restrictions [105]. Since flexibility is the counterpart to conservation this also reflects the expected order of conservation for the different reading frames with the ‘-2’ the most conserved and the ‘-3’ the least conserved. So far it is not clear whether more OLGs would be expected to naturally occur in any reading frame as both flexibility and conservation are important. Flexibility is needed to create OLGs and conservation to maintain them.

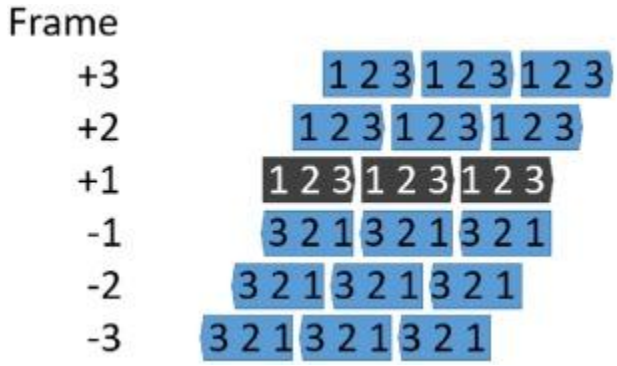

Figure 1.3: Illustration of the alternative reading frames. The ‘+1’ frame is the standard or reference reading frame and ‘+2’/’+3’ the sense overlaps, while frames ‘-1’ to ’-3’ are on the antisense strand. Figure taken from [106].

1.2.2. Possible Functions of OLGs

The most straightforward effect of OLGs is the genome compression as mentioned before and could explain the existence of OLGs in viruses but not the remaining part of the tree of life. While a shorter genome always decreases replication times in any organism, OLGs are not prevalent enough to significantly reduce genome size outside viruses. A more general applicable function is expression regulation. Same strand OLGs are transcribed together on the same mRNA and are therefore more likely to be expressed simultaneously. Antisense strand OLGs on the other hand potentially form “noncontiguous operons” [107] leading to

complementary mRNAs, which influence each other's expression (transcription and translation) in many ways [108], [109].
ORFs parallel to an existing gene are less variable due to the conservation of the MG, which is usually interpreted as a hurdle for the formation of OLGs, but could just as well be a positive factor in the evolution of OLGs, if the restrictions make the formation of functional genes more likely. Two observations hint at this conclusion. First, same strand ORFs parallel to genes have an astounding propensity to have the same hydrophobicity profile as the MG [40]. Second, antisense ORFs in the '-1' frame have a complementary polarity structure to the gene on the antisense strand [49], [50], which could result in interesting interactions between the MG and the OLG [110]. Both properties are a consequence of the structure of the SGC but have not been tested on their optimality. As a result, it has been hypothesised that OLGs are a place of *de novo* gene creation [74], [111], [112], [113], [114]. New genes which are functional would likely be eventually copied out of the OLG afterwards in order to be optimised more freely, perhaps partly explaining why most OLGs appear to be young.

1.3. Research question

The goal of this study is to broaden the theoretical basis of OLGs in order to facilitate better oriented research in the future. The structure of SGC reflects its evolutionary history. Since it is strongly linked to OLGs, the SGC is studied for optimalities with special interest in properties beneficial for OLGs to determine how important OLGs were in the evolution of the first cells. If the SGC is optimised for OLGs in any sense it can be inferred to have played an important role for early life and possibly today. But even if no optimality can be found for OLGs, the mere existence of these complex structures, which are expected to be lost in the course of evolution if they are only a chance event, raises the question of what their purpose is for an organism. Possible functions have been hypothesised before, but the list is not exhaustive so new functions of OLGs are explored here and also tested for optimality in the SGC. Before OLGs can perform any function they must first exist and one of the main counterarguments of OLGs is that genes are much too complex for two genes to be encoded in an overlapping manner. This has only been studied from an information content point of view, which revealed that OLGs are very difficult but not impossible [115]. So the argument of too high complexity in genes has not been quantified in a more biological approach and is only an assumption. One goal of this research is to quantify the change inflicted on known genes in order to create artificial overlapping genes. This can clarify whether the combination of the flexibility in the genetic encoding using the SGC and the flexibility of genes, as observed in naturally occurring homologs, is sufficient for the creation of OLGs.

2. Code Optimality for Multiple Properties

The different putatively optimal properties of the SGC have been tested in very different evolutionary hypotheses restricting the artificial code set. It has been shown that the optimality calculated strongly depends on the evolutionary hypothesis and can be removed if the right hypothesis is chosen, e.g. the mutational robustness of the SGC is not a rare feature among codes created using the 2-1-3 model [32]. Optimality can only be determined for a specific evolutionary hypothesis and while single properties can be explained without optimization by choosing specific evolutionary (historical) hypotheses, it is not clear whether any hypotheses can explain all properties without additional selection for certain properties of the code. Since it is not clear which evolutionary hypothesis is the correct one, determining the optimality of the SGC for every sensible hypothesis can show whether an optimising phase (selection) is necessary in the evolution of the SGC. Current evolutionary hypotheses use the observed structures of the SGC as a starting point, so by creating alternative codes by fixing different layers of structure in the SGC a more general answer can be found. In this study multiple properties are tested on different alternative code sets with an escalating amount of structure in order to find the robustness of the optimality in the SGC. The results have been published in [106].

Many properties have been found to be optimal in the SGC in different evolutionary hypotheses, but they are not independent of each other. In a competitive environment, every property yields a different fitness advantage that accumulates to a total fitness function. Any optimising change to the genetic code has to increase the total fitness, but it is not clear which property is improved and some could even be worsened in the process. In order to study a realistic scenario a fitness function should be constructed from different properties and tested on optimality instead of testing each property independently. Since the contribution of each property to the total fitness varies it is not clear how to construct such a fitness function, but properties can nevertheless be combined and tested to determine tendencies and conditional optimalities. Different approaches and their difficulties are discussed here.

2.1. Methods

In this study, optimality is always determined with the more dated ‘statistical approach’, determining the rarity of a property in comparison to an artificial code set, opposed to the more recently developed ‘engineering approach’, which tries to construct the best possible code for a property using genetic algorithms. These algorithms mimic evolution by starting with a random code and subject it to cycles of variation and selection until the property reaches a local extremum (minimum or maximum). Initialising the genetic algorithm with different starting codes, global extrema can be approached. The level of optimization of the SGC is determined as the percentage of the distance between the worst (minimum) and best (maximum) it covers. Also the number of variations needed in order to get to the level of the SGC, starting out from a random code, can be tracked, which is an indication of how difficult it is to develop the property. While both are interesting properties, they cannot determine whether a selective phase in the evolution of the SGC took place. If the codes that can be explored by the genetic algorithm do not form a symmetric distribution with respect to a certain property, most codes could be much

closer to either the minimum or the maximum, so a percentage of optimization does not reflect the rarity of a property. In the following chapters a detailed description of the tested properties and the code sets used is discussed.

2.1.1. SGC Properties

In this study four properties are tested, which all provide a straightforward fitness advantage for an organism. The first two properties, namely the mutational robustness and the frameshift protection, reduce energy costs of faulty translations and are therefore advantageous for the translation of all genes and have been discussed above. The remaining two properties are useful for the formation and maintenance of OLGs, namely the average ORF length in alternative reading frames and the conservation of AAs in alternative reading frames.
While the mutational robustness is often calculated for mutations on different codon positions, only the combined effect is calculated here as mutations in DNA do not know about codon position and the total mutational robustness should be optimised for in the SGC. The frameshift protection is calculated for each frame individually as well as a combination of the “+2” and “+3” reading frame as this is the most relevant for non-OLGs.
An ORF of the length of a small gene is a prerequisite in order to form an OLG and can be achieved by pairing rare codons in the MG with stop codons in the alternative reading frame. Here pairing means that two nucleotides of the two codons can overlap in the chosen reading frame. This property opposes the frameshift protection property as that property is optimised by having many stop codons in frame-shifted reading frames. While both properties cannot be realised optimally in a single reading frame, different reading frames could be used for different purposes, so both properties can be realised in a single genetic code. The average ORF length has been studied on the antisense strand before and not found to be optimal in an artificial code set maintaining the mutational robustness of the SGC [39]. Here analysis of this property is extended also to the sense strand and different alternative code sets.
A conservation of alternative reading frames due to the SGC could explain how OLGs are maintained in the genome despite being more susceptible to mutations and is a crucial property for the existence of OLGs. This property is defined as the difference between possible AAs on an alternative reading frame due to synonymous mutations on the MG and has been found optimal in the “-1” frame [51]. Since non-synonymous mutations in the MG are usually selected against anyway, reducing the influence of synonymous mutations can additionally increase the conservation of OLGs. This can be achieved by pairing similar AAs in an alternative reading frame to codons coding for the same AA in the MG. Here this property is studied in all reading frames. In the following chapters, the details of calculation for each property is discussed.

2.1.1.1. Mutational and misread error robustness

This property is defined as the average effect of a point mutation. It is calculated with (1), which has already been used in [71]. The function $d\left(a_i, a_j\right)$ returns a numerical value indicating the difference between the two AAs a_i and a_j before and after the mutation and is called the AA distance function. Since there are many possibilities to define its values, a closer discussion can be found below in chapter 2.1.1.5. Squared AA distance values are chosen so that very different AAs have a stronger impact as is expected in nature. Using squared values instead of any other

exponent greater than one is arbitrary but using different values has little effect on the result [31]. Since an exponent of two has always been used in the literature, it is also used here in order to create comparable results. The sum of the squared AA distances is averaged by dividing over the number of codon pairs n_{pairs} of original and mutated codon. Since every codon can mutate to nine different codons, is at most 576, but depending on the number of stop codons and how mutations from and to stop codons are managed, this number can vary.

$$D_m = \frac{\sum_{i,j} d^2\left(a_i, a_j\right)}{n_{pairs}} \tag{1}$$

Mutations do not occur with the same rates from every nucleotide to any other. Mutations from a pyrimidine (c,t/u) to a pyrimidine or a purine (a,g) to a purine base are called transition and occur with a different rate than a pyrimidine to purine base or vice versa [116], [117], [118], [119], [120], which is called a transversion. Unfortunately, the ratio of these mutation rates is very different in these studies, likely due to different organisms being studied. Extrapolating from this data to differences in transition and transversion rates in organisms and environments even before LUCA is not very reliable.

Similarly, misread errors do not occur with the same frequency on all codon positions [23], [24]. The most error prone position is the third followed by the first position. The second position is the least affected by misread errors. Interestingly, this reflects exactly the structure of the SGC as a change in the second position always leads to a change in AA, which is less often the case for a change in the first position and even less frequent for a change in the third position.

Both observations have been used in the literature by weighting different mutations [34], [121], [31], which lead to an increase in the optimality of the mutation and misread error robustness to previous studies [27]. The optimality of this property in itself, namely the fact that it is incredibly rare, on the other hand is very robust to variations in its calculation. Since the rates discussed above do not seem to be independent of the specific organism or the environment it is in, they are not used here in order for the results to be as general as possible.

2.1.1.2. Frameshift error abortion time

After a frameshift event, the average number of translated codons before a stop codon is encountered is defined as the frameshift error abortion time T_A [39]. It is calculated as the average absorption time of a Markov chain with the stop codons as absorbing states and any other codon as a transient state. It can be calculated from the conditional probabilities $P\left(c_j|c_i\right)$ of codon c_j following codon c_i, following equations (2)-(4) [39]. In (2), n_t is the number of transient states. The entries of the matrix $\hat{Q}_f$ in (4) are the conditional probabilities of transient states of codons.

$$T_A = \frac{1}{64} \sum_{i=0}^{n_t} \left[\vec{t}_f\right]_i \quad (2)$$

$$\vec{t}_f = \left(\hat{1} - \hat{Q}_f\right)^{-1} \cdot \vec{1} \quad (3)$$

$$\left[\hat{Q}_f\right]_{i,j} = P\left(c_j | c_i\right) \quad (4)$$

This property is calculated for all five alternative reading frames, which are assumed to be independent of each other and are therefore treated as five different properties. While reading frames on the sense strand are straightforward, reading frames on the antisense strand only make sense if an OLG on the antisense strand is translated and frameshift errors affecting this are considered. The average T_A value of both sense reading frames is the most relevant as translation can frameshift to both reading frames and will therefore also be considered in this study.

2.1.1.3. Conservation in alternative reading frames

Originally, this property was calculated in a stochastic approach [51]. Random AA sequences without stop codons were generated and the possible AAs of that sequence in the '-1' frame determined. At every position i, two AAs in the '-1' frame, here a_i and b_i, were randomly selected and their difference measured with an AA distance function $d\left(a_i, b_i\right)$. The average AA distance over the whole sequence was defined as the conservation D_c of the '-1' frame. A large D_c value reflects low conservation and vice versa. Equation (5) summarised its calculation.

$$D_c = \frac{1}{L} \sum_{i=0}^{L} d\left(a_i, b_i\right) \quad (5)$$

The stochasticity vanishes for large L and D_c converges to a fixed value. The limit of $L \to \infty$ can be calculated analytically, which allows a computationally more efficient calculation of D_c.

Before taking the limit of $L \to \infty$ the sum over L in (5) must be rewritten as a sum over the 20 different AAs and a sum over n_a, which is the number of occurrences of the AA a in the sequence.

$$D_c = \sum_{a=1}^{20} \frac{n_a}{L} \sum_{k=0}^{n_a} \frac{1}{n_a} d\left(a_k, b_k\right) \tag{6}$$

In (6), a limit of $L \to \infty$ also results in a limit of $n_a \to \infty$. For random AA sequences, as used in [51], $\frac{n_a}{L}$ converges to $\frac{1}{20}$ for $n_a, L \to \infty$. In a more general approach $\frac{n_a}{L}$ converges to P_a, which is the usage percentage of the AA a in the sequence. In nature not all AAs occur with the same frequency. Using AA usage percentages of real organisms is expected to result in more realistic results compared to an even usage of all AAs. The second sum in (6) converges to the average distance d_a of all AAs parallel to the AA a for $n_a \to \infty$, see equation (7). Using P_a and d_a in (6), the limit $L \to \infty$ results in equation (8), which is an analytic expression for D_c.

$$\lim_{n_a \to \infty} \sum_{k=0}^{n_a} \frac{1}{n_a} d\left(a_k, b_k\right) = d_a \tag{7}$$

$$D_c = \sum_{a=1}^{20} P_a d_a \tag{8}$$

In [51] only different AAs in the '-1' frame of the same AA in the '+1' frame were considered, meaning that instances of conservative mutations in the '+1' frame that are also conservative in the '-1' frame were excluded. This reduces the calculated conservation without any biological meaning, so mutations in the '-1' frame that do not change the AA and result in a AA distance of 0 are included in this study. Equation (9) is used to calculate the mean distance d_a, where i and j are two codons in the '-1' frame that code for the same AA in the '+1' frame and a_i and a_j are their respective AAs as translated by the genetic code. Here the squared AA distance is used in order to account for large differences having a much higher probability to disrupt a gene, just as in the calculation of the mutational robustness. N_a is the number of codons which code for the same AA a, so the expression $\frac{1}{2} N_a \left(N_a - 1\right)$ is the number of different codon pairs in the antisense frame.

$$d_a = \frac{2}{N_a\left(N_a - 1\right)} \sum_{i,j,i \neq j} d^2\left(a_i, a_j\right) \tag{9}$$

While the different AA usage is already incorporated in D_c as a factor, the number of codons coding for the same AA is a mostly independent property, which should also be taken into account. AAs with more codons allow for more conservative mutations resulting in a higher variability in the alternative reading frame, which should be accounted for. Here the weight $\frac{N_a - 1}{\sum_b N_b - 1}$ is introduced, using the number of conservative mutations for each AA $N_a - 1$. As a consequence, AAs encoded by only one codon do not influence this property. In the standard genetic code this is only methionine, but in alternative genetic codes this can be much more prevalent. Equation (10) is the final equation to calculate for the conservation of alternative reading frames D_c.

$$D_c = \sum_a \frac{N_a - 1}{\sum_b N_b - 1} P_a d_a \qquad (10)$$

Alternative reading frames besides the '-1' frame, have no one to one codon translation from the MG to the OLG. Instead two codons in the MG are needed to define one codon in OLG. Equation (10) is adapted to this scenario by summing over dipeptides instead of single amino acids. N_a and the group of possible AAs in the alternative reading frame are determined from all possible codon combinations maintaining the dipeptide in the MG, but only the codons that mutate in a nucleotide included in the codon on the alternative reading frame are considered in order to reduce double counting. Fig. 2.1 illustrates the calculation of d_a in the '-1' and the '+2' frame.

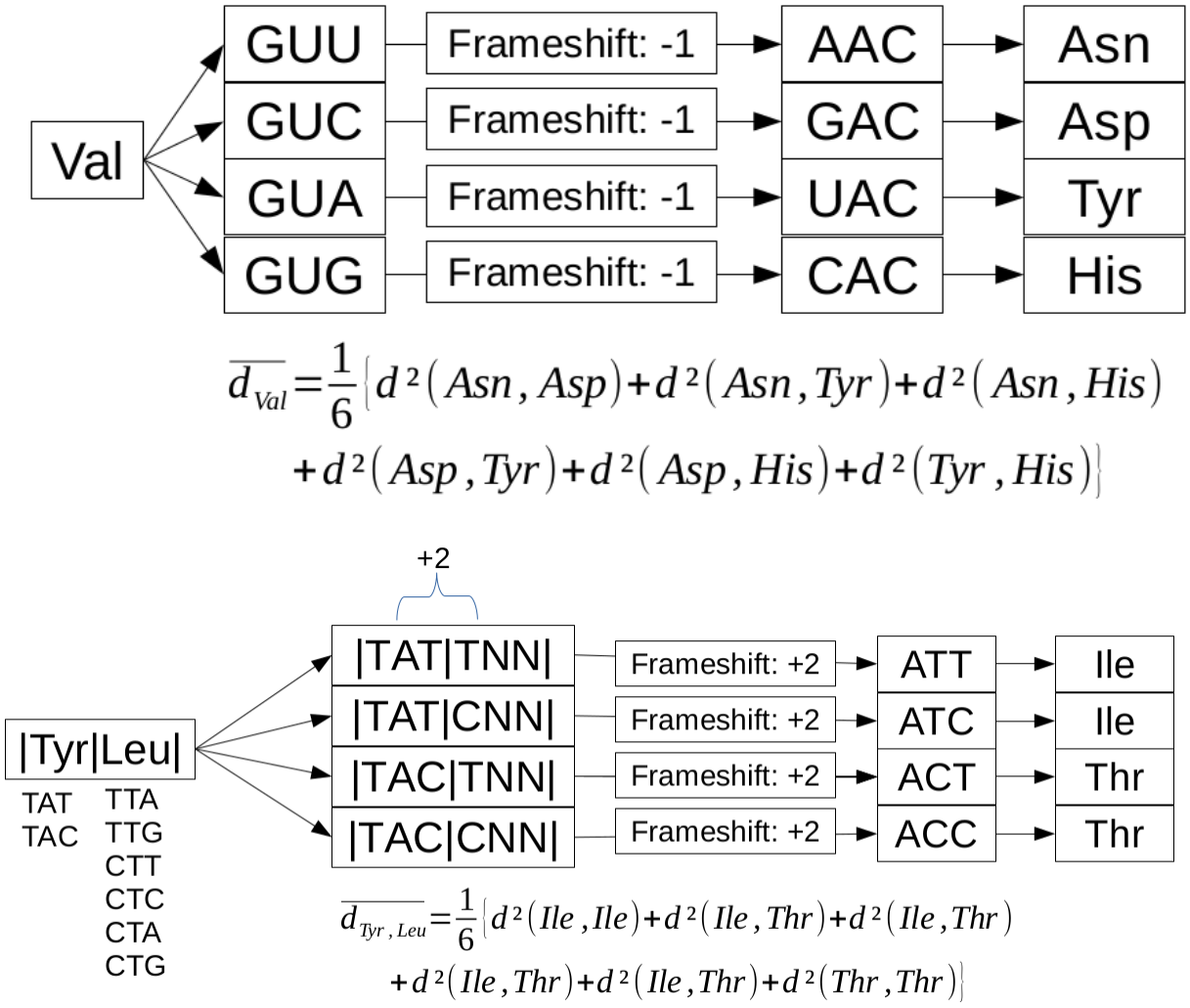

Figure 2.1: Example calculation schematics for $\bar{d}_a$. *Top:* Calculation of the average distance between AA in the '-1' frame to Valin $\bar{d}_{Val}$. *Bottom:* Calculation of the average distance between AA in the '+2' frame to the dipeptide Tyrosine-Leucine. Figure taken from [106].

Just as in the frameshift error robustness, different reading frames are assumed to be independent of each other in the conservation of alternative reading frames, so this property is also tested as five independent properties.

2.1.1.3. Average open reading frame length

Arguably the most sensible definition of an ORF in this context is the genetic material between two stop codons [75]. Defining a Markov chain, using the conditional probabilities $P(c_j|c_i)$ of

codon c_j following codon c_i, allows the average ORF length to be calculated as the average recurrence time T_R of a stop state [39]. From a stationary distribution of codon probabilities $P(c_i)$ follows the recurrence time of any state as $T^i_R = [P(c_i)]^{-1}$, so the average recurrence time of any stop state is calculated using the following equation (11).

$$T_R = [P_{stop}]^{-1} = \left[\sum_i P\left(c_i^{stop}\right)\right]^{-1} \quad (11)$$

Different reading frames, again, are assumed to be independent of each other in their average ORF length, so this property is tested as five independent properties.

2.1.1.4. Codon probabilities and amino acid usage statistics

The properties of the SGC included in this study are calculated using conditional codon probabilities, codon and AA usage statistics. While AA statistics mostly reflect what resources are available to an organism and what kind of proteins are necessary to its survival and reproduction, all kinds of codon statistics could be adapted, in accordance to the SGC, to improve encoding and translation of genes. Codon statistics are much more adaptable than the genetic code, so it is unlikely that the SGC adapted to the codon usage but vice versa. Since alternative genetic codes, which the codon statistics are definitely not adapted to, are considered in optimality calculations, the SGC may appear to be more optimal if the codon statistics of real organisms are used. In order to remove this effect, just as in [34] codon statistics $P(c_i)$ are derived from AA statistics by assuming every codon coding for the same AA occurs with the same frequency. The conditional probabilities $P^{+1}(c_j|c_i)$ in the '+1' frame are derived from $P(c_i)$ by assuming that consecutive codons are independent of each other, which results in $P^{+1}(c_j|c_i) = P(c_j)$ [34]. The conditional probabilities for alternative reading frames $P^f(c_j|c_i)$, with $f \in \{-1, \pm 2, \pm 3\}$, can also be determined from these probabilities as is derived below following [34]. Here only the formula for the '+2' frame is derived but the final equations are listed for all reading frames.

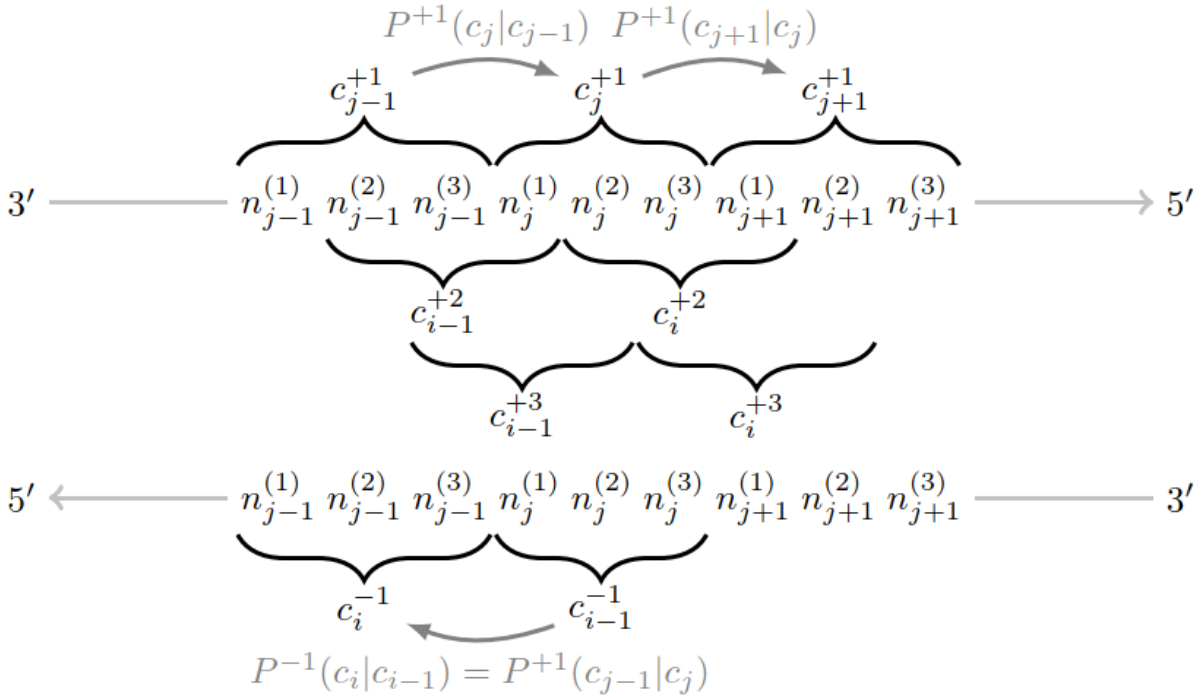

Figure 2.2: Scheme for conditional probability translations. Each codon c_l consists of three nucleotides $n \in \{a, c, g, t\}$. Figure taken from [39].

Starting at the definition of the conditional probability (12), where the index i denotes the position of codon c_i in a nucleotide sequence as shown in Fig. 2.2. Both numerator and denominator can be written using $P^{+1}(c_j|c_i)$ and therefore $P(c_j)$, see equations (13) and (14), which results in the final equation (15) for $P^{+2}(c_i|c_{i-1})$. Equations (16)-(19) are the formulas for the remaining reading frames.

$$P^{+2}(c_i|c_{i-1}) = \frac{P^{+2}(c_i \cap c_{i-1})}{P^{+2}(c_{i-1})} \tag{12}$$

$$\begin{aligned} P^{+2}(c_i \cap c_{i-1}) &= \sum_{n_{j-1}^{(1)}, n_{j+1}^{(2)}, n_{j+1}^{(3)}} P^{+1}(c_{j+1}|c_j) P^{+1}(c_j|c_{j-1}) P(c_{j-1}) \\ &= P(c_j) \cdot \left[\sum_{n_{j+1}^{(2)}, n_{j+1}^{(3)}} P(c_{j+1}) \right] \cdot \left[\sum_{n_{j-1}^{(1)}} P(c_{j-1}) \right] \end{aligned} \tag{13}$$

$$P^{+2}(c_{i-1}) = \sum_{n_{j-1}^{(1)}, n_j^{(2)}, n_j^{(3)}} P^{+1}(c_j|c_{j-1}) P(c_{j-1})$$

$$= \left[\sum_{n_j^{(2)}, n_j^{(3)}} P\left(c_j\right) \right] \cdot \left[\sum_{n_{j-1}^{(1)}} P\left(c_{j-1}\right) \right] \quad (14)$$

$$P^{+2}\left(c_i | c_{i-1}\right) = P\left(c_j\right) \cdot \left[\sum_{n_{j+1}^{(2)}, n_{j+1}^{(3)}} P\left(c_{j+1}\right) \right] \cdot \left[\sum_{n_j^{(2)}, n_j^{(3)}} P\left(c_j\right) \right]^{-1} \quad (15)$$

$$P^{+3}\left(c_i | c_{i-1}\right) = P\left(c_j\right) \cdot \left[\sum_{n_{j+1}^{(3)}} P\left(c_{j+1}\right) \right] \cdot \left[\sum_{n_j^{(3)}} P\left(c_j\right) \right]^{-1} \quad (16)$$

$$P^{-1}\left(c_i | c_{i-1}\right) = P\left(c_{j-1}\right) \quad (17)$$

$$P^{-2}\left(c_i | c_{i-1}\right) = P\left(c_{j-1}\right) \cdot \left[\sum_{n_{j-2}^{(1)}, n_{j-2}^{(2)}} P\left(c_{j-2}\right) \right] \cdot \left[\sum_{n_{j-1}^{(1)}, n_{j-1}^{(2)}} P\left(c_{j-1}\right) \right]^{-1} \quad (18)$$

$$P^{-3}\left(c_i | c_{i-1}\right) = P\left(c_{j-1}\right) \cdot \left[\sum_{n_{j-2}^{(1)}} P\left(c_{j-2}\right) \right] \cdot \left[\sum_{n_{j-1}^{(1)}} P\left(c_{j-1}\right) \right]^{-1} \quad (19)$$

In this study the annotated genes of *E.coli* O157:H7EDL933 (Accession number NC 002655, EHEC) is used to determine the AA usage statistics (as was previously used in [39]). A bacterial organism is used as prokaryotes are a good case study for early life. Only using genes that are supposedly in LUCA could improve the analysis, but the predicted set of genes in LUCA is not well known; current estimates vary between 500 and 1000 genes depending on the method used and many genes might not be recognised due to more dramatic changes in LUCA's descendants [122]. Here it is assumed that annotated genes of current organisms better reflect what is necessary for a living organism.

2.1.1.5. The AA distance function

There have been two fundamentally different methods to create a measure to determine how different AAs are. The straightforward approach is using physical and chemical properties of the AAs to determine how different they are [123], [27]. But it is not known which of these properties are important for proteins and their importance is likely to vary for each gene and AA position therein. A second approach aims to circumvent this problem by using AA exchange statistics taken from homologs as a measure. In this approach it is assumed that the exchange rates reflect all physical and chemical properties important for evolution and hence protein function [124]. But also this approach cannot resolve the gene and position dependence. Furthermore, these statistics partly reflect the structure of the SGC as AAs that have a short mutational distance to each other are exchanged more often. This effect can be reduced but not eliminated

by taking more distant homologs into account so that the sequences reflect what can be exchanged rather than what is likely to be exchanged by chance [125].
Here the physicochemical approach is used, taking polar requirement as the single chemical measure as is common in the literature [26], [124], [34], [121], [126], [33], [71]. Polar requirement is similar to hydrophobicity [25] and has resulted in the highest error robustness optimality when used as a measure [26]. The polar requirement values for all AAs are taken from [28]. The distance between two AAs is defined as the absolute value of the difference in their polar requirement values.

2.1.1.6. Optimising the influence of stop codons

Mutations resulting in a premature stop codon usually render the gene non-functional and therefore have a stronger influence than any other class of mutation. Taking this into account, mutation error robustness calculations would be dominated by differences in stop codons in alternative codes. In this study the focus is on the rest of the code rather than the stop codons, so minimising their influence is important.
Four different approaches to minimise the effect of stop codons have been discussed in the literature [71]. Ascribing a fixed polar requirement value to stop codons or defining a fixed distance value in the distance function whenever any AA is compared to a stop codon are two of the four options. Both values could be optimised in order to minimise the effect of stop codons, but it is not clear whether the optimised values for one property and code set can be extrapolated to other use cases. It is also not clear whether it is possible to minimise the influence of stop codons for sets with many and with few stop codons at the same time.
The third option is to simply treat the code as if the stop codons would be impossible, so there cannot be a mutation towards them and is called the exclusion approach. The last option is called a suppression approach. When translating a gene, stop codons can sometimes be suppressed and instead be read as another codon which has the same first two nucleotides. In all known genetic codes stop codons always end on a Pyrimidine and if only one of the Pyrimidine ending codons code for a stop the other Pyrimidine ending codon can be read instead and thus suppress the stop codon. Here this observation is extended also to all other cases. If both codons with the same first two nucleotides and a Pyrimidine at the end code for stops a random one of the two other codons in this block that is not a stop will be read. In this work hypothetical codes with all kinds of structures are studied, so stop codons can also have a Purine ending, but will be treated respectively. If only one Purine ending codon codes for a stop it is suppressed by the other Purine ending codon in this block and if both Purine ending codons code for stops a random one of the two Pyrimidine ending codons that is not a stop is used. If all four codons of a block code for a stop a random AA is used as a value. It has been suggested in [71] that the suppression approach should minimise the influence of stop codons. In the following this is tested and was confirmed for cases in which there are no more than 6 stop codons in the genetic code.
Creating random codes with a random number of stop codons, as described later in section 2.1.3.1, the influence of stop codons for different approaches can be quantified. The percentage of better codes is calculated as a function of the number of stop codons and normalised by the percentage of better codes among the whole set of artificial codes. This was done for the mutational error robustness and the conservation of alternative reading frames using the

suppression as well as the exclusion approach, c.f. Fig. 2.3. Small numbers of stop codons (0-6) are especially interesting, since naturally occurring variants of the genetic code have only been observed with up to four stop codons, so large numbers of stop codons (>7) are probably unreasonable even in a pre LUCA environment. For small numbers of stop codons, the suppression approach indeed yields a lower influence of the number of stop codons on the percentage of better codes compared to the exclusion approach. While the exclusion approach only outperforms the suppression approach for large numbers of codons in the conservation of alternative reading frames, this property barely depends on the number of stop codons at all. Consequently, the suppression approach is used throughout this study. It is noteworthy that the mutational robustness roughly varies by a factor of two in both directions for small numbers of stop codons in the suppression approach.

.

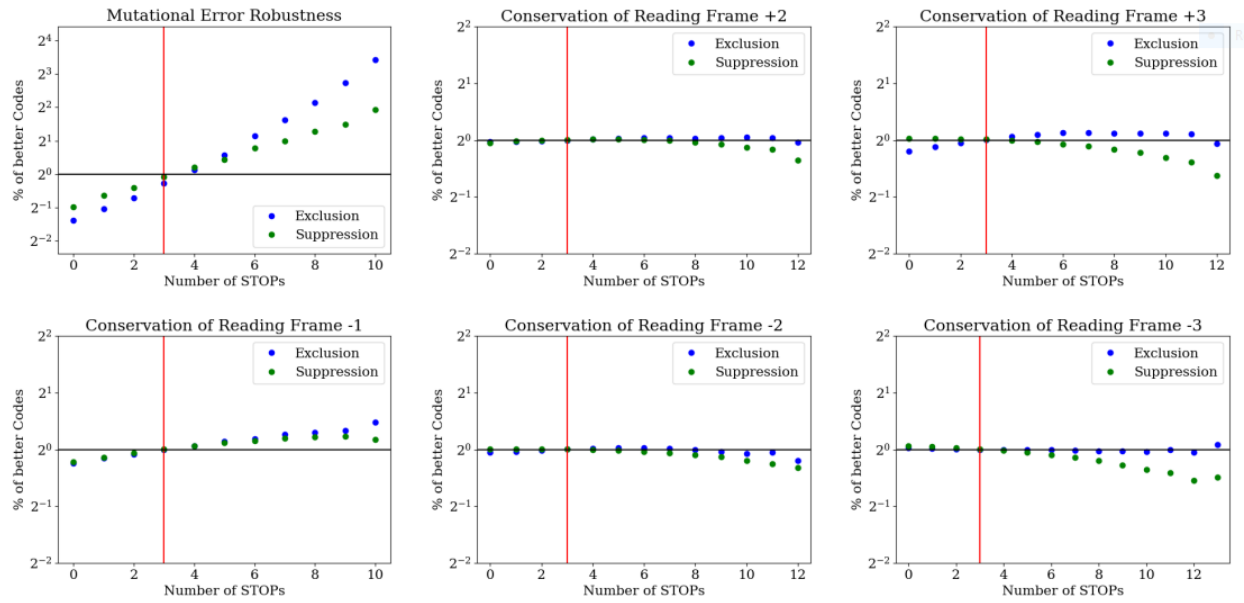

Figure 2.3: The influence of stop codons. The percentage of better codes for each number of stops is normalised by the percentage of better codes of the total set. The red line indicates three stop codons just as in the SGC. For the mutation error robustness (*top left*), the suppression approach has less influence on the results. In the conservation of the alternative reading frames both approaches yield almost the same results for small numbers of stop codons(0-6), while the exclusion approach is slightly better for high numbers of stop codons(>7).

2.1.2. Testing for optimality in multiple properties

Only a combination of different properties of the SGC can rightfully reflect a real optimization process by natural selection. While a sensible fitness function can currently not be reconstructed for the pre-LUCA environment as neither its physical and chemical properties nor the different competing organisms are known. But in order to approach this difficult problem two crude approximations and a combination of both are tested in this study.

The first approach is a consecutive testing similar to [39], where codes that conserve the mutational error robustness are used to test the optimization of the average ORF length on the antisense strand and the frameshift error absorption time. Here, starting with a large set of

alternative genetic codes, all properties are tested for optimality to determine the most optimal property. All codes that perform at least as well as the SGC in this property will be combined to a new set of alternative genetic codes. Repeating this process on the new set of alternative genetic codes until either all properties are tested or no codes are better than the SGC yields a ranking of optimal properties and the conditional optimalities of each property under the condition that all higher ranking properties are at least as good as the SGC in the alternative code set. This approach tries to recreate the importance of each property for the survival of an organism in a pre-LUCA environment by observing its optimization and ranking them against each other, thus shedding some light on a possible contribution of each property in a fitness function. Since only a small set of potentially beneficial properties of the SGC are tested here, the ranking is incomplete and some of the properties might be in a different order or show a different optimality if more properties are tested. Consequently, the focus lies more on the method than on the results in this study.

The second approach creates different approximate fitness functions and tests each one to find the most optimal one. Each fitness function F is a linear combination of different sets of normalised properties ε_i, see Eq. (20). Values x_i of property i are normalised by using the distance to the mean value of the alternative code set μ_i in units of its standard deviation σ_i as a measure, see Eq. (21). Here a positive value indicates a more optimal than average code and a negative value a less optimal one respectively. Since it is not clear a priori which property is more important in a pre-LUCA environment, properties will not be weighted, which is obviously a very rough approximation to a real fitness function but can be considered as an important first step.

$$F = \sum_i \varepsilon_i \tag{20}$$

$$\varepsilon_i = \pm \frac{|x_i - \mu_i|}{\sigma_i} \tag{21}$$

Both approaches, namely the consecutive testing and the combination testing, as described above, can be combined in order to have a fitness function, but are not completely dependent on the equal weighting of the properties therein and should yield more realistic results. This can be done by using the consecutive testing schema but instead of testing single properties in each cycle, a combination testing is used. All properties of the most optimal combination in each cycle are removed in the next iteration. This approach was developed by analysing the problems of the previous two methods and is the third attempt of this study to combine different properties of the SGC in optimality testing.

2.1.3. Artificial Code Sets

All alternative codes will at most contain the 20 AAs used in the SGC and stop codons, but other than that, all three layers of structure in the SGC, as discussed in chapter 1.1.1, will be subject to change. While changes in the first two layers of the structure of the SGC, namely the AAs which are encoded and how many codons code for each AA, are straightforward, changes

in the third layer, namely the relations between codons and therefore the AAs encoded by them, needs further subdivisions.
Usually it is not possible to change only one of the layers. While it is straightforward that higher levels of structure will always be affected by inflicting change on a lower layer, changes in the third level of structure will also change the second layer in most cases. Consequently the codesets are not grouped by the layers of structure they change but structural ideas. Completely random codes without any imposed structure are a focus on the first layer of structure. Codesets focusing on the composition of the codesets focus on the second layer of structure. The third layer of structure is divided into absolute structures and relative structures. The difference between the two is in their construction process. While the absolute structure code sets (such as particular block sets) use templates that are randomly filled, the relative structure code sets take already assigned codons into account before each new assignment in the construction process.
Besides varying the different layers of structure in the SGC, alternative code sets are constructed according to the 2-1-3 Model [72], which has been shown to produce very mutational error robust genetic codes [32], and a specific evolutionary hypothesis, which is a combination of existing hypotheses as envisioned in [59].

2.1.3.1. The random code set

In this code set not all 20 AAs or a stop codon have to be included and therefore reflects the variation in the first layer of structure. Each code is constructed by selecting a random AA or stop codon for each codon. More than $4 \cdot 10^{84}$ different codes exist in this set and it reflects all possible triplet genetic codes using members of the canonical amino acid set. Here it is called the ‘Random’ code set and every other set constructed here is a subset of this set. A variation of this set which fixes the number of stop codons to three is also considered here and is called the ‘Random_fs’ (fixed stops) code set.

2.1.3.2. Composition code sets

Fixing the first layer of structure, namely having all 20 AAs included at least once within every code in the codeset, while maintaining completely random assignment to codons results in the ‘Random_faa’ (fixed AAs) code set. The number of stop codons is arbitrary in this code set. Also fixing the number of stop codons to three results in the ‘Random_fb’ (fixed both) code set, which has the basic first layer structure of the SGC. As a comparison code set, which fixes the second layer of structure, the ‘Degeneracy’ code set is introduced, which is a random code set but the number of codons coding for each AA resembles exactly the numbers in the SGC.

2.1.3.3. Absolute structure code sets

The template that is used for the absolute structure code sets is the block structure of the SGC. Collecting all codons which encode the same AA into groups and shuffling the AAs encoded in each group results in the ‘Blocks’ code set. Here the stop codons build their own group and are also shuffled with the AAs, so there can be one to six stop codons. The second layer of structure is therefore not fixed in this code set and the degeneracy of each AA varies from one to six codons. Since only one group has only one codon in it and only two have 6 codons in it

the variations in the second layer of structure are considered small. Approximately $5 \cdot 10^{19}$ different genetic codes can be constructed this way.
While the 'Blocks' code set maintains the exact absolute structure of the SGC, it could have been slightly different if the tRNA-AA association developed differently. This has been explored in more detail in [71], where all possible blocks of assignments for codons that only differ in the last nucleotide were created using the wobble binding rules, see Table 2.1. Drawing random boxes to first create a new block structure for a genetic code and then randomly filling in AAs results in the 'Random_Blocks' code set. Since many boxes contain stop codons, drawing random boxes would usually result in many stop codons in the resulting genetic codes. In order to fix the first layer of structure, boxes with stop codons will be drawn in such a way that the resulting genetic code always contains three stop codons and can at least have 20 different AAs. Every AA is first added once to the constructed genetic code before randomly adding AAs in the remaining slots. It has been estimated that at most 10^{50} different codes can be constructed this way [71].
A comparison between the results on the 'Random_Blocks' and the 'Blocks' code set can clarify just how important the specific block structure of the SGC is, as it is maintained in many studies on the optimality of the standard genetic code and the evolutionary hypothesis used therein to constructed alternative genetic codes.

Table 2.1: Possible AA assignment patterns for codons differing only on the last nucleotide as constructed in [71]. Boxes with the same letter are assigned the same AA. The weights will only be used in the historical code set.

U	X	X	X	X	X	X	X
C	X	X	X	X	X	X	X
A	X	Y	X	STOP	STOP	STOP	X
G	X	Y	Y	STOP	X	Y	STOP
Weights	22/42	17/42	3/42	3/6	1/6	1/6	1/6

2.1.3.4. Relative structure code sets

Relations between codons can be constructed according to any property of the SGC, but here only the mutational robustness will be used. It is the most optimal property but it also creates the most straightforward fitness advantage for the organism. Thus, the relative structure code sets try to encode similar AAs to 'neighbouring' codons, which are codons that only differ by one nucleotide. This will be done for the 'Random_fb', the 'Degeneracy', the 'Blocks' and the 'Random_Blocks' code set resulting in the 'Random_fb_n', the 'Degeneracy_n', the 'Blocks_n' and the 'Random_Blocks_n' code set ('_n' for 'neighbours'). The '_n' code sets are constructed as the previous ones, but every time a random AA is selected to be added to the code, a similar AA to at least one neighbouring codon that has already been assigned is chosen. First a list of AAs that have at most a difference d in polar requirement values to at least one neighbouring AA is collected and then a random AA from that list assigned to the current codon in the

construction process. If the list is empty a random AA is assigned. A value of $d = 0.25\,\sigma_{PR}$ has been found to create codes with the most similar neighbours (data not shown), where σ_{PR} is the standard deviation of all 20 polar requirement values. A similar approach has been used in [32] and the mutational robustness was strongly enhanced in these codes.

2.1.3.5. The 2-1-3 Model code set

The 2-1-3 Model assumes that not all nucleotides of a codon were recognised in an early stage of a code [72]. First, only the second nucleotide was read in the ribosome with a later addition of the first and third nucleotide. This has been hypothesised due to the increasing optimization of the first and third position in mutational robustness [27]. Consequently, the 2-1-3 model assumes that the SGC started out with only four different AAs in the beginning, namely valine, alanine, aspartic acid and glycine, and became more complex as more nucleotides were recognised. As codons with the same second nucleotide mostly code for similar AAs [24], code extension is assumed to include similar AAs to the AA which was encoded before. It is a remarkable model as it has a high probability to create artificial genetic codes with similar mutational robustness as the SGC if the block structure of the SGC is maintained and a specific scheme of code extension is used [32], see Fig. 2.5. New additions to the code have to be similar to what has been encoded previously on that codon. The similarity criterion was the same as in the relative structure code sets.

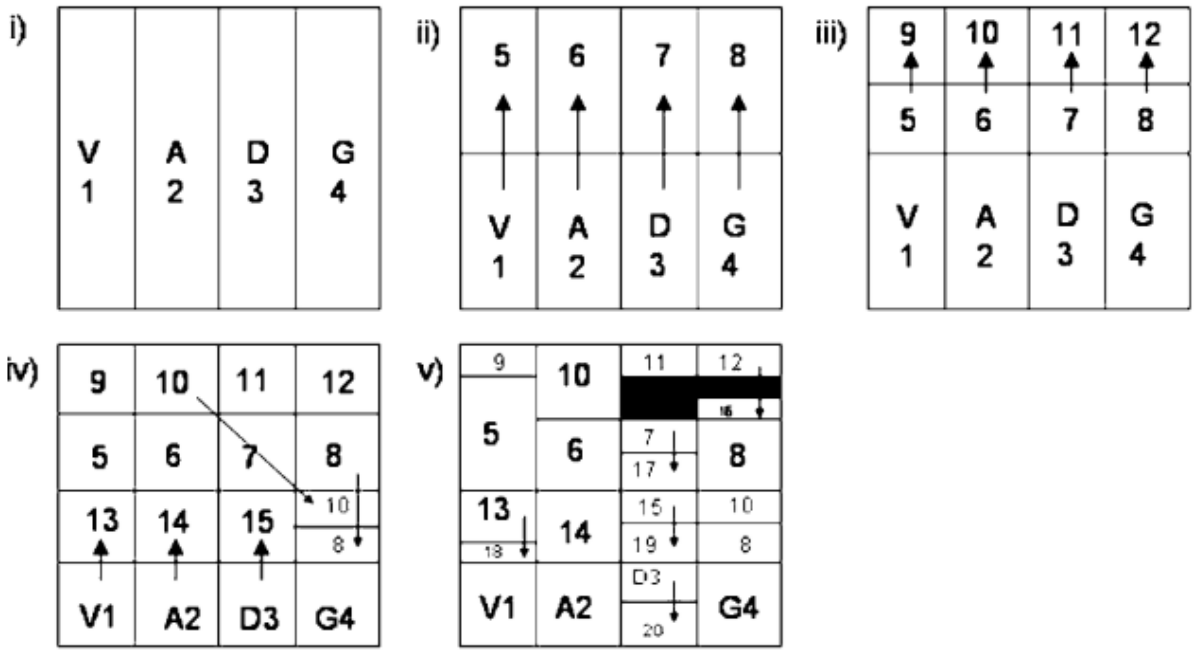

Figure 2.4: Code expansion scheme in the 2-1-3 model. Figure taken form [32].

2.1.3.5. A historical code set

The evolutionary hypothesis used for this code set is the coevolution hypothesis, which has been used before to create alternative code sets [31], [32], [121], [26], but is extended by combining it with a version of the 'Random_Blocks' code set.

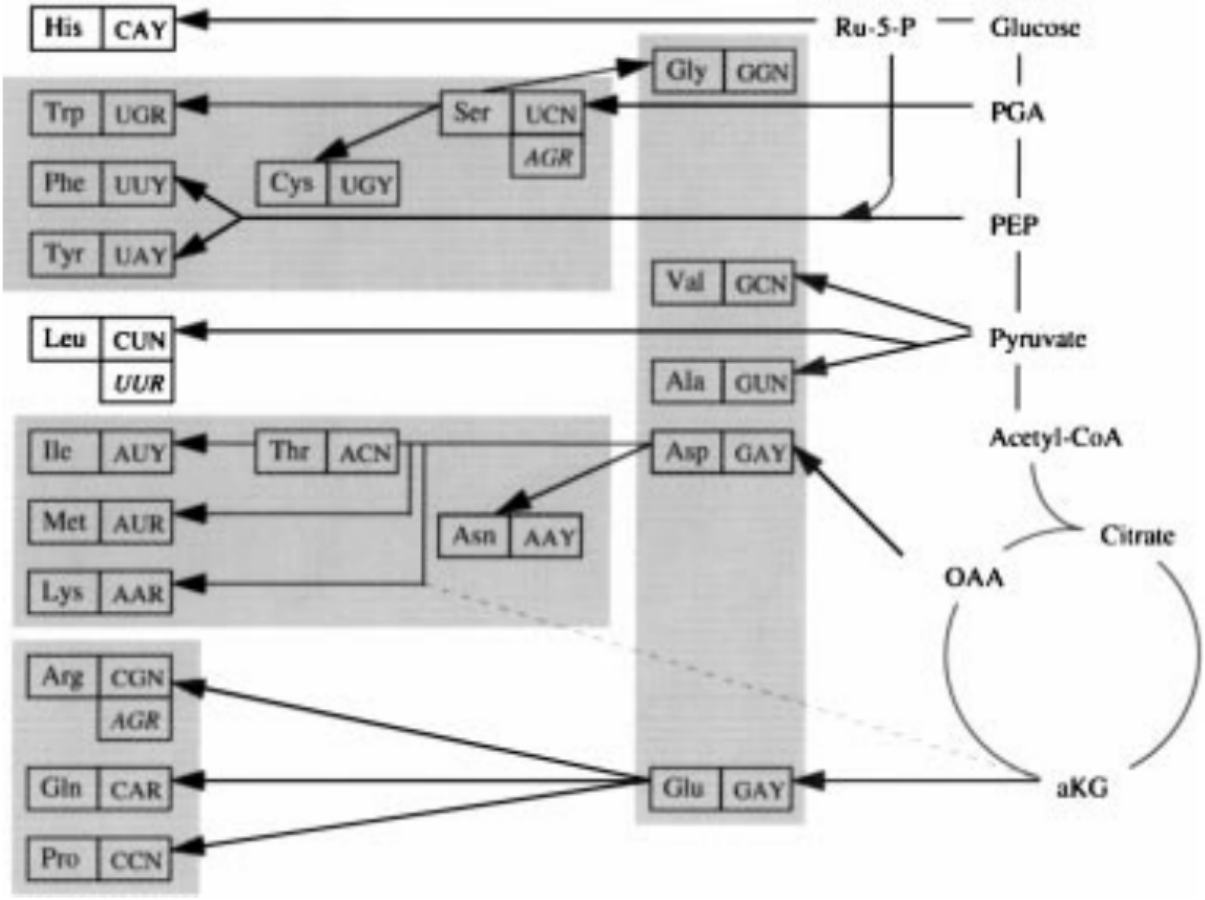

Figure 2.5: Biosynthetic pathways of the 20 AAs used in the SGC. Most AAs from the same pathways share the same first nucleotide in the SGC. Figure taken from [121].

The observation that AAs formed in the same biosynthetic pathways mostly share the same first nucleotide in their respective codons [127], see Fig. 2.5, was later used to create alternative genetic codes [121] by using the blocks code set but instead of randomly assigning AAs to the blocks, AAs are shuffled in such a way that the AA from the same biosynthetic pathway still share the same first nucleotide afterwards. This is done by splitting the 20 AAs into 4 groups with 5 AAs each depending on their first nucleotide [121], namely $U_n \in$ {Phe, Ser, Tyr, Cys, Trp}, $C_n \in$ {Leu, Pro, His, Gln, Arg}, $A_n \in$ {Ile, Met, Thr, Asn, Lys} and $G_n \in$ {Val, Ala, Asp, Glu, Gly} and only shuffling the assigned AAs between codon blocks inside each group, see Fig. 2.6

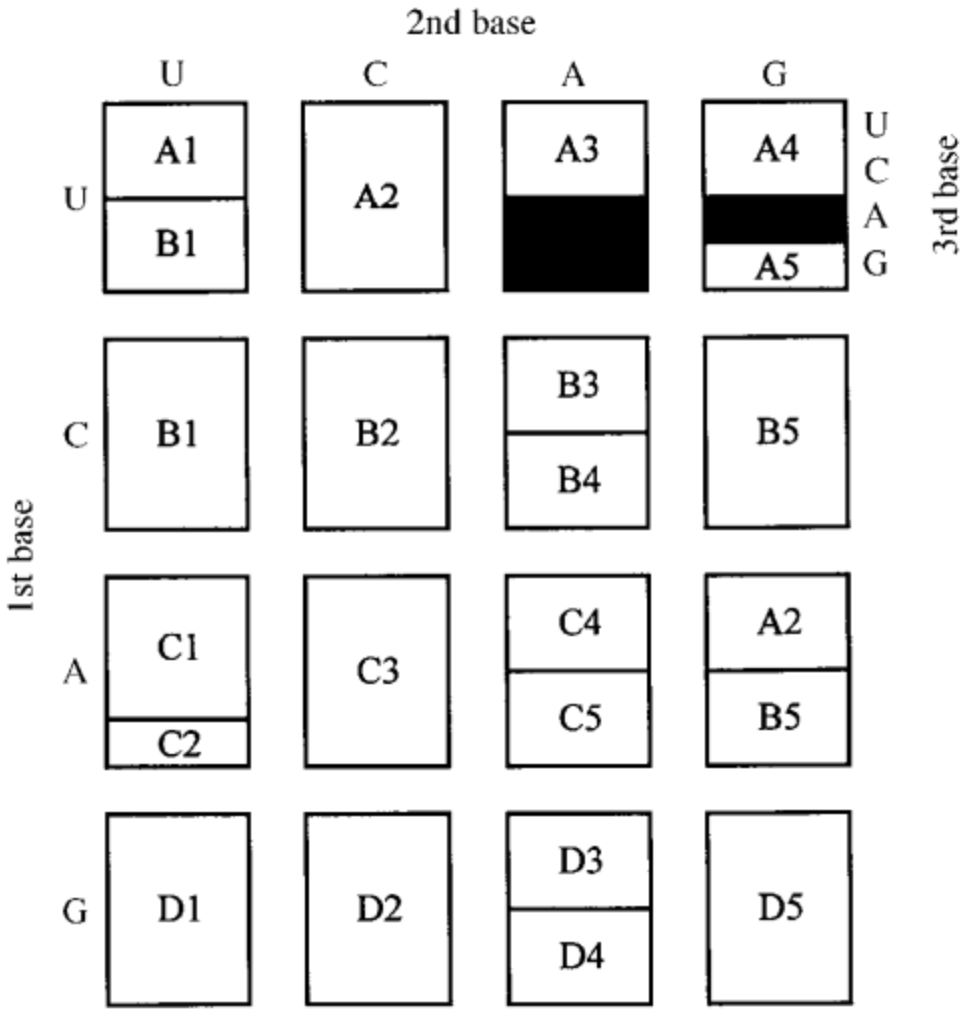

Figure 2.6: Codon blocks divided into four groups A_n, B_n, C_n and D_n according to the first nucleotide of their respective codons. Figure taken from [121].

In order to also include the possibility that different codon blocks could have developed in the evolution of the SGC, randomised block structures as described in the 'Random_Blocks' code set are used for this 'Historical' code set in this study. In order to make the block structures similar to the SGC, two constraints are imposed on the process of drawing the 16 degeneracy boxes from the 7 possible blocks in Table 2.1. Firstly, the number of stop codons can only vary between two and four with an average of three. This way the variation in stop codons is minimised but some variation is still possible, as observed in naturally occurring variants of the SGC. It can be realised by randomly drawing two of the 16 boxes needed for a new code from the set of boxes containing a stop codon with the weights shown in table 2.1. The second restriction is, that the number boxes for each split type in the constructed codes on average is the same as in the SGC. Here three different split types are considered, namely boxes with no split, boxes with a 2-2 split, which is a distinction between a purine and a pyrimidine of the last nucleotide of the codon, and boxes with a 3-1 split, in which only the codon with a G at the last position codes for a different AA. Boxes 1, 5 and 7 go into the no split category and must be

included eight times on average. Boxes 2, 4 and 6 go into the 2-2 split category and must be included seven times on average. Box 3 goes into the 3-1 split category and must be included only one time on average. In order to meet these averages the weights as shown in table 2.1 are used when drawing the remaining 14 boxes of the code from the boxes without stop codons. If all boxes with the same first nucleotide are from the no split category, only four of the five AAs in this group can be added to the constructed genetic code. If two or less boxes are from the no split category, the remaining blocks will be filled with random AAs, first from the group that could not be included in their respective column with the same first nucleotide and then from all 20 AAs. In some cases not all 20 AAs are included into the code, but at least 16 are included in every code.

2.2. Results

The results of the optimality calculations have been published in [106] for all code sets but the 'Historical' code set since the focus of the paper was on structural code sets. Multi property testing was only done on the 'Historical' code set as the calculations are computationally very intensive and the results were not published as the data revealed additional challenges of multiple property testing.

2.2.1. Mutational Robustness

The mutational robustness is optimal in all but the 213-model, see left panel of Fig. 2.7. This is mostly expected, as this property has been reported to be highly optimal in the literature except in the 213-model. In two of the code sets, namely the ‘Degeneracy’ and the ‘Degeneracy_n’ code set, the SGC is even more optimal than ever reported before as no more mutationally robust code could be found in 10^{10} codes. The percentages of better codes than the SGC can be found in Table A.1 in appendix A.

Comparing the distributions of D_m values for alternative code sets, see right panel of Fig. 2.7, fixing the number of stop codons, fixing the number of different AAs included into the code and even arranging similar AAs next to each other has a negligible influence on the average code and only slightly changes the standard deviation of the distribution. The latter is the reason for the big differences in code optimality in these code sets. Conserving the degeneracy as in the ‘Degeneracy’ and the ‘Degeneracy_n’ code set or at least strongly restricting it as in the different (‘Random_)Blocks(_n)’ code sets strongly increases the average mutational error robustness of the artificial codes. Again the optimality difference between the two types of sets originates in their different standard deviation, with the ‘Degeneracy(_n)’ code sets having a much smaller standard deviation. This indicates that the degeneracy structure of the SGC is a very important feature for this property

The most interesting difference is between the ‘(Random_)Blocks_n’ and the ‘213-model’ code sets, as they are very similar but produce very different results. While the former do not include a history of how coevolution took place and just randomly put similar AAs mutationally close to each other, the later is based on a very specific hypothesis of how the SGC developed.

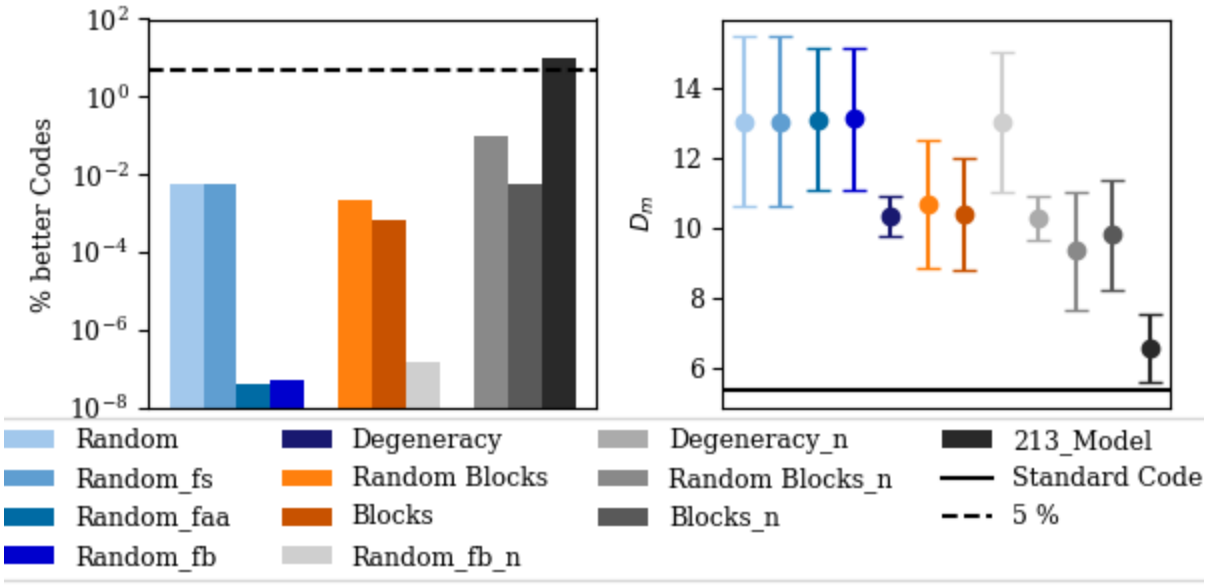

Figure 2.7: Mutational and misread error robustness D_m calculated in different sets of alternative genetic codes. Each set contains 10^{10} codes. *Left:* SGC optimality measured in percentage of better codes. The threshold of 5% for an optimal property is indicated by the dotted line. *Right:* Alternative genetic code distributions summarised by their mean values and respective standard deviations. As comparison the D_m value of the SGC is marked by the horizontal line.

2.2.2. Frameshift error abortion time

In the alternative reading frames on the sense strand, the SGC is only optimal in the 'Degeneracy' and the 'Degeneracy_n' code set in the '+2' and '+3' frame individually, see top row of Fig. 2.8. The average of both sense reading frames on the other hand is additionally optimal in the 'Random_fs', the 'Random_fb', the 'Random_fs_n', the 'Random_fb_n' and the '213_Model' code set. These are all the sets whose codes have exactly three stop codons, which results in lower average distances until a stop codon is encountered and also very small standard deviations in the codes distributions, see bottom row of Fig. 2.8. Interestingly, all code sets in which the SGC is not optimal have higher average number of stop codons but a higher average T_A value. One explanation could be that the impact of each stop codon on T_A decreases with increasing number of stop codons, so really high numbers of stop codons in a single code barely matter, while lower numbers of stop codons have a strong impact. This could partially explain the number of stop codons in the SGC. Other structures of the SGC have no clear impact in one or the other direction.

On the antisense strand the SGC only appears to be optimal in the '213-model' in the '-1' frame, see Fig. 2.9. A clear distinction between coding frames and absorbing frames cannot be made from this data.

The percentages of better codes than the SGC in the sense and antisense reading frames can be found in Table A.1. and Table A.2 of appendix A respectively.

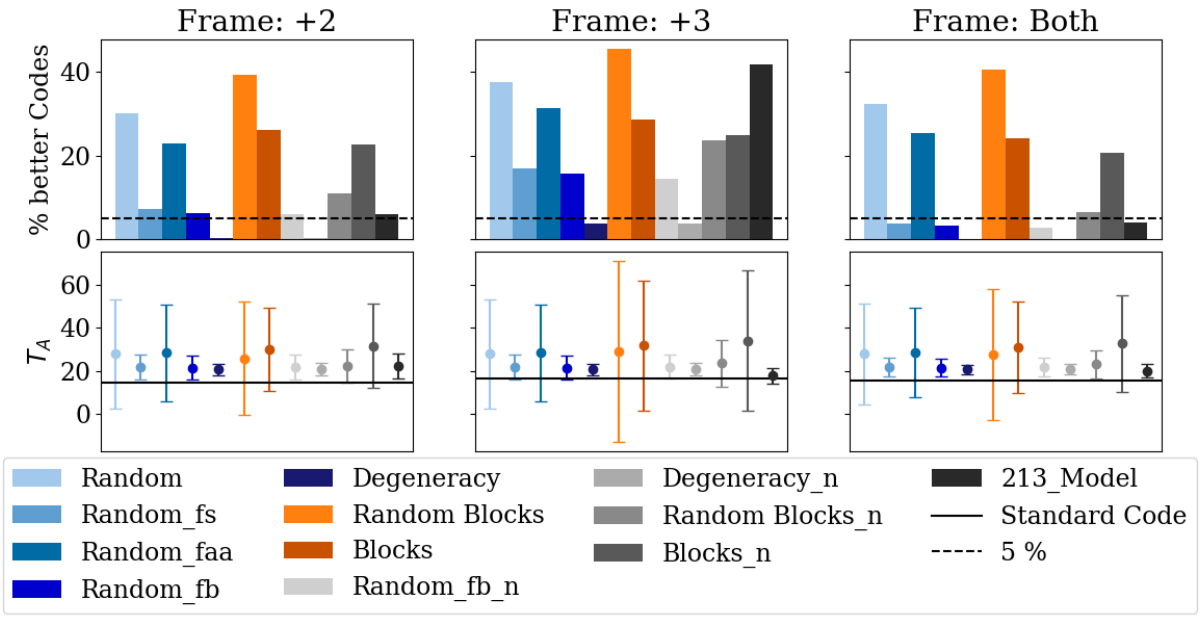

Figure 2.8: Sense frameshift error abortion times T_A in different sets of alternative genetic codes. Each set contains 10^5 codes. Both alternative sense reading frames as well as their average are shown. *Top:* SGC optimality measured in percentage of better codes. The threshold of 5% for an optimal property is indicated by the dotted line. *Bottom:* Alternative genetic code distributions summarised by their mean values and respective standard deviations. As comparison the T_A value of the SGC is marked by the horizontal line.

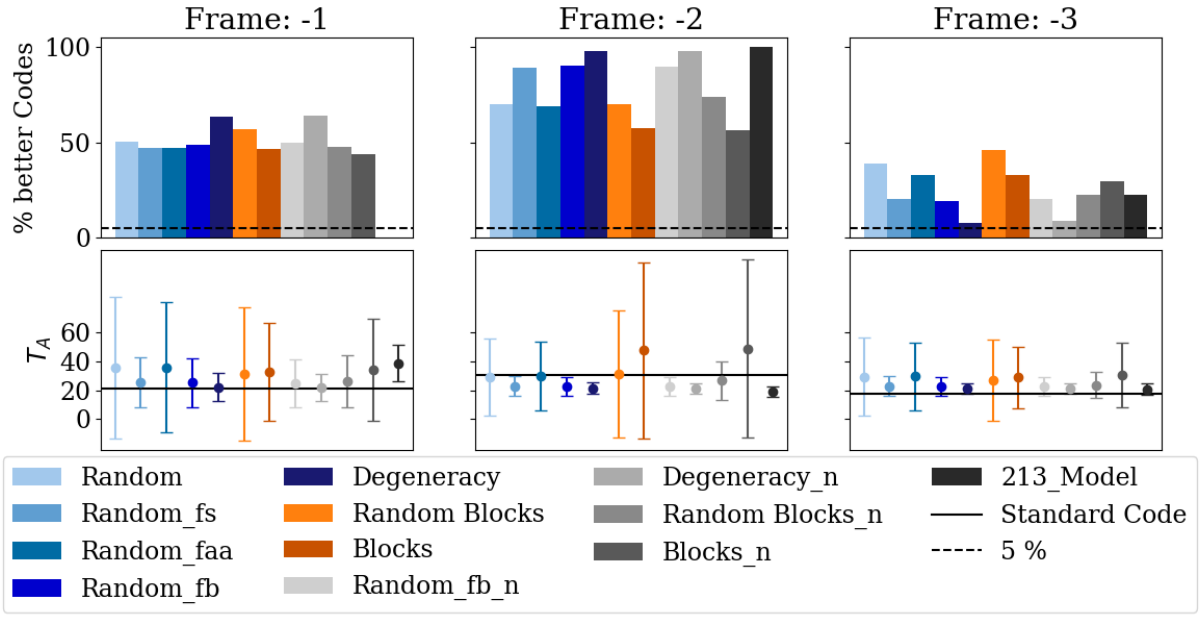

Figure 2.9: Antisense frameshift error abortion times T_A in different sets of alternative genetic codes. Each set contains 10^5 codes. All alternative anti-sense reading frames are shown. *Top:* SGC optimality measured in percentage of better codes. The threshold of 5% for an

optimal property is indicated by the dotted line. *Bottom:* Alternative genetic code distributions summarised by their mean values and respective standard deviations. As comparison the T_A value of the SGC is marked by the horizontal line.

2.2.3. Conservation of alternative reading frames

In the ‘ Degeneracy’ and the ‘Degeneracy_n’ code set the SGC appears optimal in all reading frames, see top row of Fig. 2.10. Additionally, it is optimal in the ‘213-model’ in the ‘+2’,’+3’ and ‘-3’ frame. The ‘-1’ frame shows the most optimalities as the SGC only is not optimal in the ‘Blocks’, the ‘Blocks_n’ and the ‘213-model’. The percentages of better codes than the SGC can be found in Table A.3 of appendix A.
Just as in the mutational robustness and the frameshift error abortion time, the ‘Degeneracy(_n)’ code sets have a very small standard deviation, see bottom row of Rig. 2.10, explaining the optimality of the SGC in these code sets. Only in the ‘-2’ frame can a structural influence of the genetic codes on the conservation of alternative reading frames be observed, namely the block structure creates very low D_c, which means a strong conservation. This is an expected behaviour as the ‘-2’ frame is the combinatorially most restricted reading frame by far, since the third nucleotides of both codons in the ‘+1’ and the ‘-2’ frame overlap [105]. Most strikingly, the SGC, which also has the block structure, does not have an especially low D_c value in the ‘-2’ frame. Comparing D_c values of different reading frames for the SGC, the values are unexpectedly similar, except for the ‘-1’ frame, which has roughly 20 times higher values in all code sets, see bottom row of Fig. 2.10. Since the ‘-1’ frame is averaged over 20 codon groups, one for each AA, and all other reading frames are averaged over 400 dicodon groups, one for each dipeptide, it is expected that this difference is an artefact arising in the calculation and does not mean the ‘-1’ frame is 20 times less conserved than all other frames. Despite all efforts no reason for this factor was found in this study.

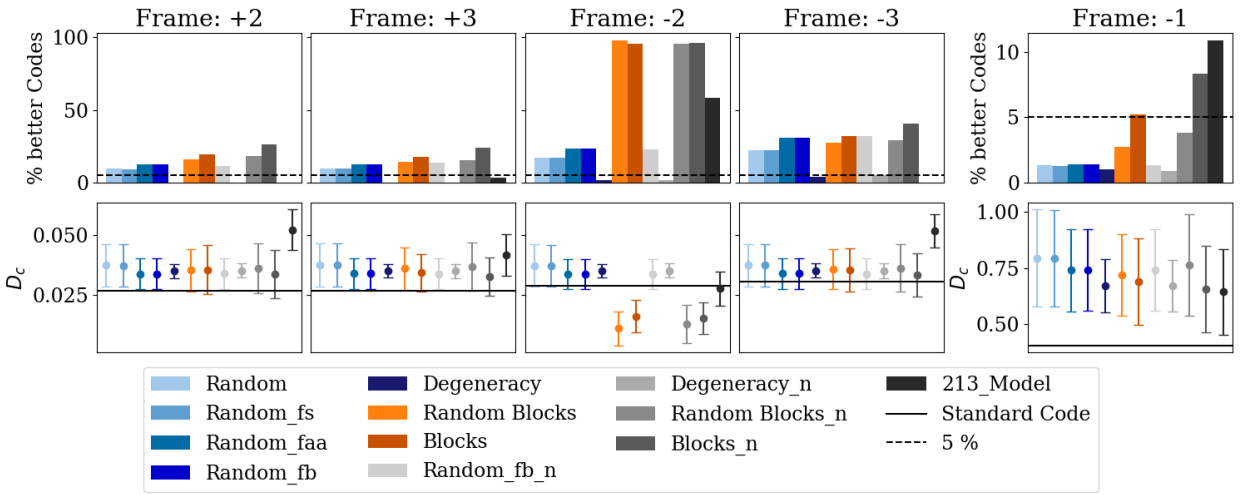

Figure 2.10: Conservation in alternative reading frames D_c calculated in different sets of alternative genetic codes. Each set contains 10^7 codes. *Top:* SGC optimality measured in percentage of better codes. The threshold of 5% for an “optimal” property is indicated by the

dotted line. *Bottom:* Alternative genetic code distributions summarised by their mean values and respective standard deviations. As comparison the D_c value of the SGC is marked by the horizontal line.

It is not clear how strong the conservation of alternative reading frames should be to maximise the number of overlapping genes as flexibility is also crucial to create OLGs that can be conserved afterwards. A trade-off value between conservation and flexibility is most likely if this property has biological significance. All reading frames except the '-1' frame being so similar despite a reasonable expectation of large differences is an indication that a specific trade-off value has been achieved. Calculating the standard deviation σ_D between D_c values of all alternative reading frames except the '-1' frame as a measure of the similarity between reading frames, the SGC appears to be highly optimal compared with codes from the 'Blocks' code set as only 0.26% of codes are more similar in their conservation value across reading frames, see Fig. 2.11.

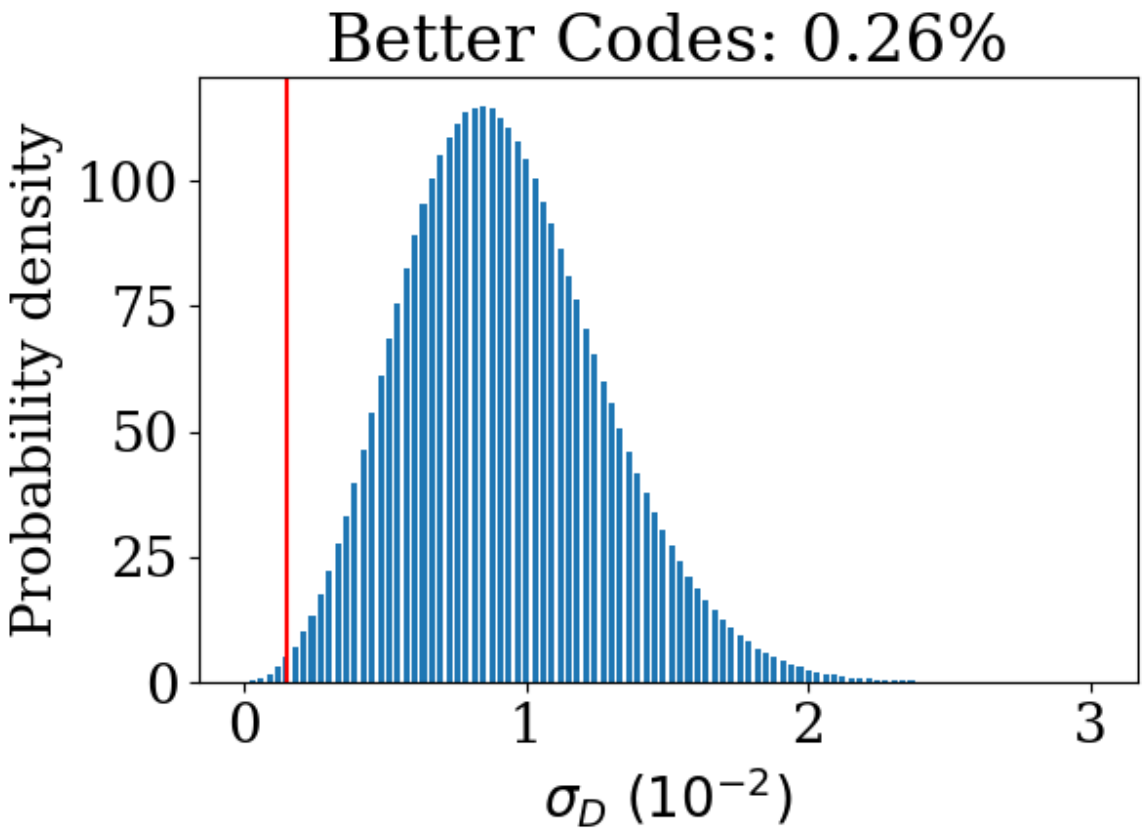

Figure 2.11: Standard deviation of D_c values across all alternative reading frames but the '-1' frame. The red line indicates the value of the SGC. 10^7 alternative codes are used from the 'Blocks' code set.

2.2.4. Average ORF length

The average ORF length is the property with the least code sets in which the SGC is optimal, as it is only optimal in the '-2' frame of the 'Degeneracy', the 'Degeneracy_n' and the '213_Model' code set; see top row of Fig. 2.12. The frameshift error abortion time and the average ORF length are different properties but strongly correlated and just as in the former no clear structural influences can be seen. Code sets with a fixed number of stop codons have very small standard deviations in their distributions compared to code sets with a variable number of stop codons, see bottom row of Fig. 2.12. The percentages of better codes than the SGC can be found in Table A.4 of appendix A.

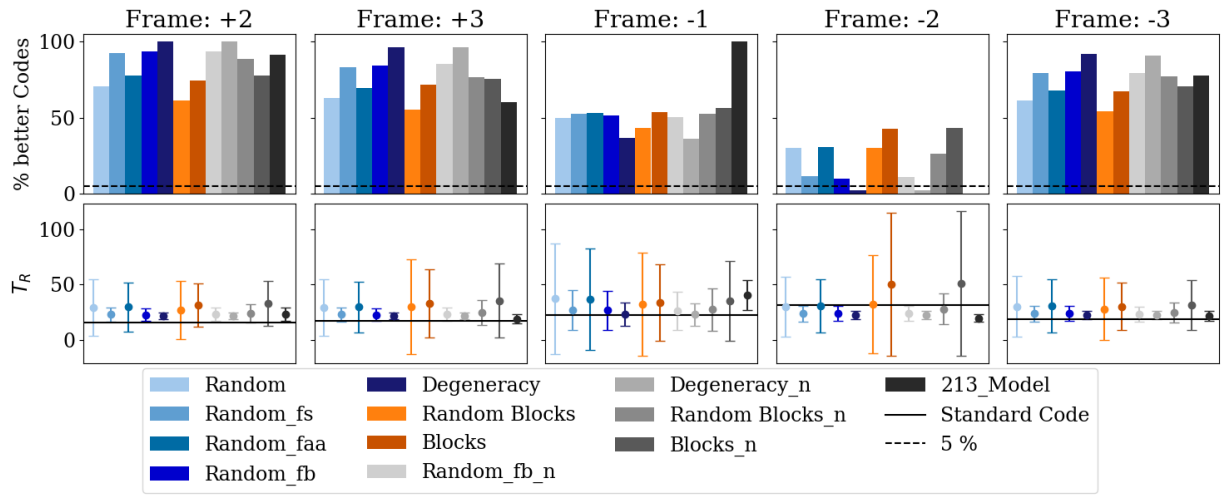

Figure 2.12: Average ORF length T_R calculated in different sets of alternative genetic codes. Each set contains 10^5 codes. *Top:* SGC optimality measured in percentage of better codes. The threshold of 5% for an optimal property is indicated by the dotted line. *Bottom:* Alternative genetic code distributions summarised by their mean values and respective standard deviations. As comparison the T_R value of the SGC is marked by the horizontal line.

2.2.5. Multi property testing

The average ORF length and the frameshift abortion times are very similar properties, as the former is the average number of codons between two stop codons and the latter the average number of codons to the next stop codon starting at a random codon. But the average ORF length is optimal for a large number of codons while the frameshift abortion time is optimal for a small number of codons, so the two properties are almost completely opposite to each other. Therefore it makes no sense for one reading frame to be optimal in both properties at the same time, but it is also important in order to remove artificial optimalities as explained in the following example. If the average ORF length is optimal in some reading frame at some point in the consecutive testing, all codes with a lower average ORF length in this reading frame will be removed for the next testing and the SGC has the lowest average ORF length. Since the frameshift abortion time is optimal for a short distance to the next stop codon, the SGC now has the lowest distance compared to the remaining codes and this property will turn up optimal

afterwards. Consequently, in the consecutive multi property testing whenever one of the two properties turns out to be optimal in a certain reading frame, the other property in this reading frame is also removed from this list of remaining properties to be tested. Similarly, in the parallel multi-property testing for each reading frame only one of the two properties can be included into the fitness function at a time. It is assumed here that properties in different reading frames are mostly independent of each other.

2.2.5.1. Consecutive testing

As expected the mutational robustness is the most optimal property followed by the conservation in the '-2' frame with 15% of the remaining codes being more conserved than the SGC in this reading frame, see Table 2.2. Already at the second position in the priority list, the property is no longer optimal and a selection process cannot be argued for. In positions three to seven an alternating behaviour in the properties absorption and ORF length occurs. The assumption that properties in different reading frames are sufficiently independent of each other does not hold true and must be dropped. Both properties depend too strongly on the number of stop codons, which can vary between two and four in the 'Historical' code set used for this analysis. This analysis shows that $1{,}6 \cdot 10^{-9}$% of codes are better than the SGC in all properties.

Table 2.2: Consecutive testing on the 'Historical' code set.

Rank	Better Codes	Property	Code Set Size
1.	0.00036%	Mutational Robustness	10^{10}
2.	15%	Conservation: +2	36230
3.	30%	Absorption: -3	5284
4.	24%	ORF length: -2	1579
5.	37%	Absorption: +3	375
6.	43%	ORF length: +2	138
7.	42%	Absorption: -1	60
8.	80%	Conservation: -3	25
9.	85%	Conservation: -1	20
10.	94%	Conservation: -2	17
11.	100%	Conservation: +3	16

2.2.5.2. Parallel testing

Testing every possible subset of all properties results in 15551 different combinations. The most optimal subset of properties consists only of the mutational error robustness, see Table 2.3. The next best subset is the combination of mutational error robustness with frameshift error absorption time in the "+3" frame. This result suggests that none of the other properties has been selected for as they cannot increase optimality any further.

Table 2.3: Parallel testing on the 'Historical' code set with 10^{10} codes.

Percentage	Properties
0.00036%	Mutational Robustness
0.0005%	Mutational Robustness, Absorption +3
0.0007%	Mutational Robustness, Absorption -1

2.2.5.3. Combined testing

Table 2.4: Combined testing on the 'Historical' code set.

Rank	Better Codes	Property	Code Set Size
1.	0.00036%	Mutational Robustness	10^{10}
2.	7%	ORF length: -2 Absorption: +2,+3,-1,-3 Conservation: +2	36052
3.	55%	Conservation: -3	2374
4.	68 %	Conservation: -1	1315
5.	96 %	Conservation: +3	881
6.	99 %	Conservation: -2	844

Again, the mutational robustness is the most optimal property followed by a combination of six different properties, see Table 2.4. The combination in the second position has 7% of better codes than the SGC, which is not below the 5% threshold, but much closer than the 15% of the second property in the consecutive testing. The six properties include conservation and frameshift error absorption on the '-2' frame, which is contradictory for optimization for OLGs.

The six properties include frameshift error absorption for all but the '-2' frame, which is again not in favour of optimisation for OLGs. Interestingly, both frameshift error absorption and average ORF length are part of this combination, showing that the two properties are not completely dependent on each other in different reading frames.

2.3. Discussion

While the results of the mutational robustness showed that the details of the chosen evolutionary history of alternative codes matter, the optimality of the SGC in itself is a very robust feature as even the '213-model', which is the only model that in the literature and this study that could explain the mutational robustness of the SGC without a selection process, is optimal in many properties, including the frameshift abortion times, which is a feature that clearly brings a fitness advantage to any organism. This result could be produced by only testing four of the many different purported properties of the SGC and more optimalities could likely be found in a more exhaustive study. The small pool of properties in this study makes the result more convincing as some properties will likely turn out optimal in a larger pool. A clear fitness advantage for an early life form of each property is a crucial requisite in order to prevent misleading results (false positives due to multiple testing).

It is not yet known in which environment organisms existed in the time before LUCA and what functionality the first cells/replicators had, so defining a threshold value for optimality is very difficult. Selection processes are subject to stochastic fluctuations depending on population sizes and a fitness advantage has to be strong enough to overcome those fluctuations in order to get fixed without appealing to chance. Also the degree to which a property has been optimised is very unclear as properties are interdependent and small changes are very difficult to detect. Even though it is difficult to determine which properties have been selected for, this study expands the evidence even further that some kind of optimization was part of the evolution of the SGC.

Consecutive testing, which has only been attempted in the literature by creating artificial codes that conserve the mutational robustness of the SGC [38], [39], is strongly subject to dependent properties. While completely independent properties most likely do not exist, since all have to be realised in the same genetic code, consecutive testing should only be used for 'mostly' independent properties.

While combining, for example, the frameshift error abortion time of the two alternative sense reading frames is straightforward, combining different properties is much more difficult due to the unknown weightings of the different properties in the fitness function. The equal weighting used in this study suggests that the mutational error robustness is by far the most important property as no combination with any other property used in this study could improve the optimality. It therefore makes sense to do the combined testing as it is an approximation to weighting the mutational robustness so strongly that all other properties are negligible and can be tested independently of the former. The combination with the lowest percentage of better codes than the SGC does have a lower value than the second most optimal property in the consecutive testing.

Some code sets properties of the SGC are extremely rare to a point that it is not clear if the SGC could have realistically been found by natural selection. How many codes could have been

tested in such a process has not been studied yet. Nevertheless, in code sets like the 'Degeneracy(_n)' code sets, in which the chance of finding a code similar to the SGC in its mutational robustness is less than 1 in 10^{10}, it makes more sense to rule out the underlying evolutionary hypothesis than to assume such an effective selection process. Such an argument can also be applied to the structure of the SGC. The block structure of the SGC, which is a result of the wobble binding rules of the tRNA, is as rare as 1 in 10^{65} in random codes and cannot be found by a selection process given the biological resources available on Earth and must therefore be explained in any evolutionary (historical) hypothesis. Finding an approximation to how many codes must be tested in order to find a code similar to the SGC could help with designing sensible evolutionary hypotheses for the SGC and could be determined by genetic algorithms.

The results of this study do not clearly support the idea that there is a strong optimization for OLGs, but only two properties have been tested in this study. Only the similarity of the conservation of alternative reading frames turned out to be optimal but its function is not known. As the function and creation of OLGs is only poorly understood yet, it is not clear whether the two properties relevant for OLGs are actually biologically relevant. The average ORF length does not consider that some stop codons can be removed by synonymous mutations in the MG, which is not only a theoretical possibility as very long OLGs exist. Also frameshift abortion conflicts with this property, which is at least obvious in the sense reading frames, and a trade-off between the two properties could be the truly "optimal" value. Similarly, it is not clear whether the conservation of alternative reading frames should have an extreme value or whether an optimal value represents a trade-off between different functions. OLGs must not only be conserved but also be created in the first place, so coding flexibility is also a very important feature, which is an opposing property to the conservation of alternative reading frames. The conservation of alternative reading frames is unexpectedly similar between reading frames, which is a strong hint that a trade-off value has been realised. A follow up study on the conservation versus flexibility trade-off is presented in chapter 3.

The SGC consists of a fixed number of codons and incorporating one property will influence every other property, so strictly speaking, every optimization involves a trade-off. Even the number of different AAs encoded in the SGC is a trade-off to the mutational robustness, which heavily depends on the degeneracy of each AA. Thinking of every property of the SGC in terms of trade-offs offers a different angle on code optimality and should be studied further. A first step has been made in this study by trying to identify which structures influence which properties, identifying dependencies between properties.

3. Flexibility - Conservation Trade-off in the SGC

In order for an optimised tradeoff value to exist on the conservation-flexibility spectrum for alternative reading frames in the SGC, a possible fitness advantage associated with having a particular location within this spectrum must be derived. The first property that comes to mind is the number of existing OLGs in a genome as both properties influence this number. A higher flexibility enables more OLGs to be formed, while a higher conservation maintains those OLGs so that they are not lost due to random mutations. While an optimal trade-off value that optimises the number of OLGs seems natural it is just as likely that the effects cancel each other out. This depends on the specific dependence of OLG creation and loss on this trade-off value, which is not known.

Following the assumptions that OLGs are sometimes involved in *de novo* gene creation and that most OLGs will be copied out eventually, the number of OLGs in the genome is not an important property anymore. In this case a newly formed OLG must only be conserved long enough to be copied out, which could lead to an optimal conservation value as conserving OLGs for longer times is unnecessary.

The conservation of alternative reading frames as a measure is the average effect of a conservative mutation in the mothergene on an alternative reading frame. A trade-off value could also be linked to the evolution of a newly formed gene in sequence space due to random mutations and natural selection. Still following the hypothesis of OLGs as a method for *de novo* gene creation, another possible function of this trade-off value is to optimise gene creation and optimization in a rugged fitness landscape. Mutations with a small effect are better suited to drive a sequence to the top of a fitness peak and conserve it there, according to Fisher's geometric model [128]. Stochastic fluctuations can always drive a gene outside of a local maximum after some time, but the average effect of a mutation sets the time scale for such an event. Even though the time scales of such processes are not known, there is a limit on how long it can take before reasonably speaking of conserving a sequence to a certain fitness peak. The strong mutational robustness, which is a very small average effect of a mutation in a normal gene, as opposed to an OLG, has been argued [126] and shown [129] to be advantageous for gene optimisation, i.e. facilitating the path of a gene in sequence space to the top of its current fitness peak. Larger effects of mutations on the other hand help to scan a bigger fitness landscape in a shorter amount of time, which is especially important for OLGs as the MG limits their evolution. But also after finding a function in sequence space, a newly translated reading frame needs to have enough functionality in order to be subject to purifying selection and not lost due to random mutations in the genome. Considering the gene as situated in a fitness landscape of sequence space, it is essential that the new gene is in a high enough fitness peak in order to be able to pass this threshold rather than being lost to drift. But also after it has enough (selectable) function to be maintained in the genome, in order to optimise this newly formed gene to improve its function in subsequent evolution, a small local fitness maximum will limit its potential. The smaller the mutation effect size the longer it takes for a sequence to leave a fitness peak due to stochastic fluctuations, so a very small mutation effect size can be disadvantageous if the sequence is in a low fitness peak.

The actual fitness landscape of genes is controversial and not well understood, but some evidence suggests that it is rugged and disconnected [130], [131]. In light of this it is a sensible

hypothesis to view gene evolution as a two step process. New genes emerge in OLGs and evolve to a high enough fitness peak due to larger average mutation effect sizes in OLGs. After being copied out and becoming a normal gene with fewer sequence constraints, genes evolve to the top of their current fitness peak in smaller mutational steps.
Another advantage of using OLGs for sequence space exploration is that it constitutes an optimal genome usage. Maintaining a genome is costly [37] as it has to be copied in every cell division. Enlarging the genome with junk sequence in order to be able to evolve new genes is a suboptimal solution to using the alternative reading frames of existing genes for this purpose. This enlarges the space in which new genes can be formed by up to five times of genome size since five alternative reading frames exist. Here this hypothesis will be simplified in a toy model to examine whether choosing a particular average effect size of a mutation in sequence space can optimise finding the highest peak of the system.

3.1. Fitness space exploration model

The goal of this toy model is to determine whether the average step size can optimise finding the biggest fitness peak in sequence space. For such a proof of concept each component of the system will be simplified as much as possible. While sequence space is a high dimensional object, here it will be represented by a 2D surface with periodic boundary conditions. Sequences will be moving around in this space according to two different types of stochastic motion, namely conserving and evolving mutations, see Fig. 3.1. The 'conservative' mutations have a small step size s_c and have a higher chance of moving the sequence to higher fitness values. This represents mutations with small effect size and the result of natural selection favouring the survival of higher fitness mutations. 'Evolving' mutations on the other hand have a large step size s_e and are not influenced by the fitness landscape. These represent rare large effect mutations that by chance are not removed by natural selection. For any given step of the simulation, the probability for a conservative motion is p_c and consequently $1 - p_c$ for evolvative motions.
Motions of sequences in sequence space will always have the full distance $s_{e/c}$ but in a random direction. In order to make movements of the conservative mutations towards higher fitness values more likely, the probability of going into each direction should depend on the relative fitness in relation to other possible directions. For easier calculation, a discretisation into N equiangular directions is used in the simulation. First the fitness f_i value at each possible end position is calculated and the minimum value f_{min} of all positions determined. The probability to go into each direction is then calculated as in eq. (22). The +1 in the numerator ensures that all $p_i > 0$ and the denominator is normalisation such that $\sum_{i=1}^{N} p_i = 1$. The angular resolution will be $N = 100$ throughout this study.

$$p_i = \frac{1+f_i-f_{min}}{\sum_{j=0}^{N}(1+f_j-f_{min})} \tag{22}$$

Initially, the particles will be randomly distributed in sequence space. The fitness landscape consists of two spherical fitness cones with the same size but different heights, see Fig. 3.2. The size of the two peaks is the same in order to have an equal amount of particles in each peak initially. The two cones have their maximal fitness value in the middle and the fitness outside the cones is zero.

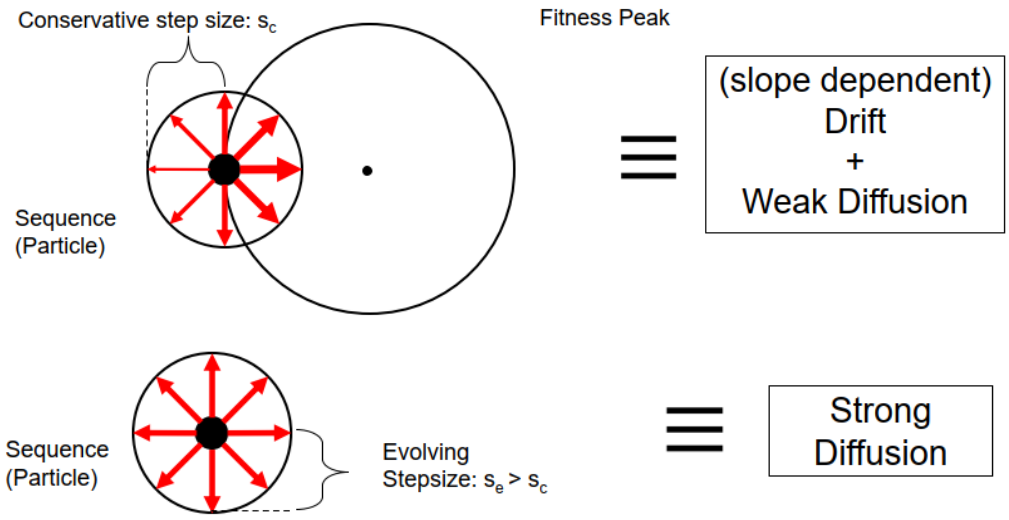

Figure 3.1: Types of motion for sequences in sequence space. Particles depicted as large dots move either by random diffusion with a step size s_e (*bottom sketch*) or by a directed diffusion with step size s_c (*top sketch*). The font weight of the arrows indicate the probability to go in each direction. Directed diffusion has a higher chance to go towards the fitness peak centre.

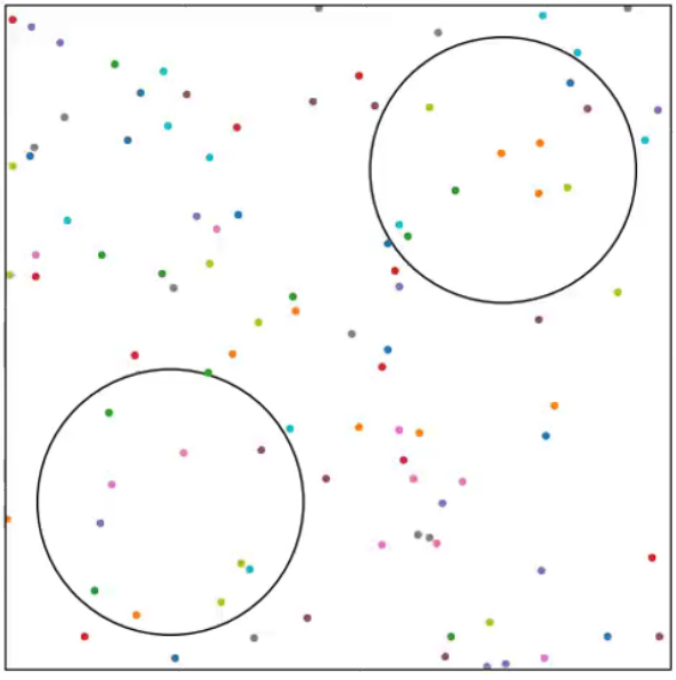

Figure 3.2: Fitness peaks (circles) and initial distribution of particles (dots). Each fitness peak is a symmetric cone with radius 0.2 in a normalised sequence space of size 1x1. The top right peak has a height H (=fitness value) of 160 and the bottom left peak has a height of 40.

3.2. Results

For the right parameter values, the model expresses the expected dynamics of sequences gathering in the larger fitness peak, see left panel of Fig. 3.3. The stochastic fluctuations in the lower fitness peak are much bigger than those in the larger fitness peak, reflecting the capability of both peaks to conserve sequences. Since sequences can leave the smaller peak much easier, it is clear that all particles accumulate in the larger peak eventually. The timescale until all particles end up in one peak can be decreased, e.g. by decreasing p_c, at the cost of larger stochastic fluctuations, a lower proportion of sequences in the larger peak at a time and therefore a lower average fitness value of a sequence, see right panel of Fig. 3.4. Consequently, the average fitness of a sequence after a given evolution time can be optimised by introducing just as much stochasticity into the system as needed for most sequences to reach the higher peak.

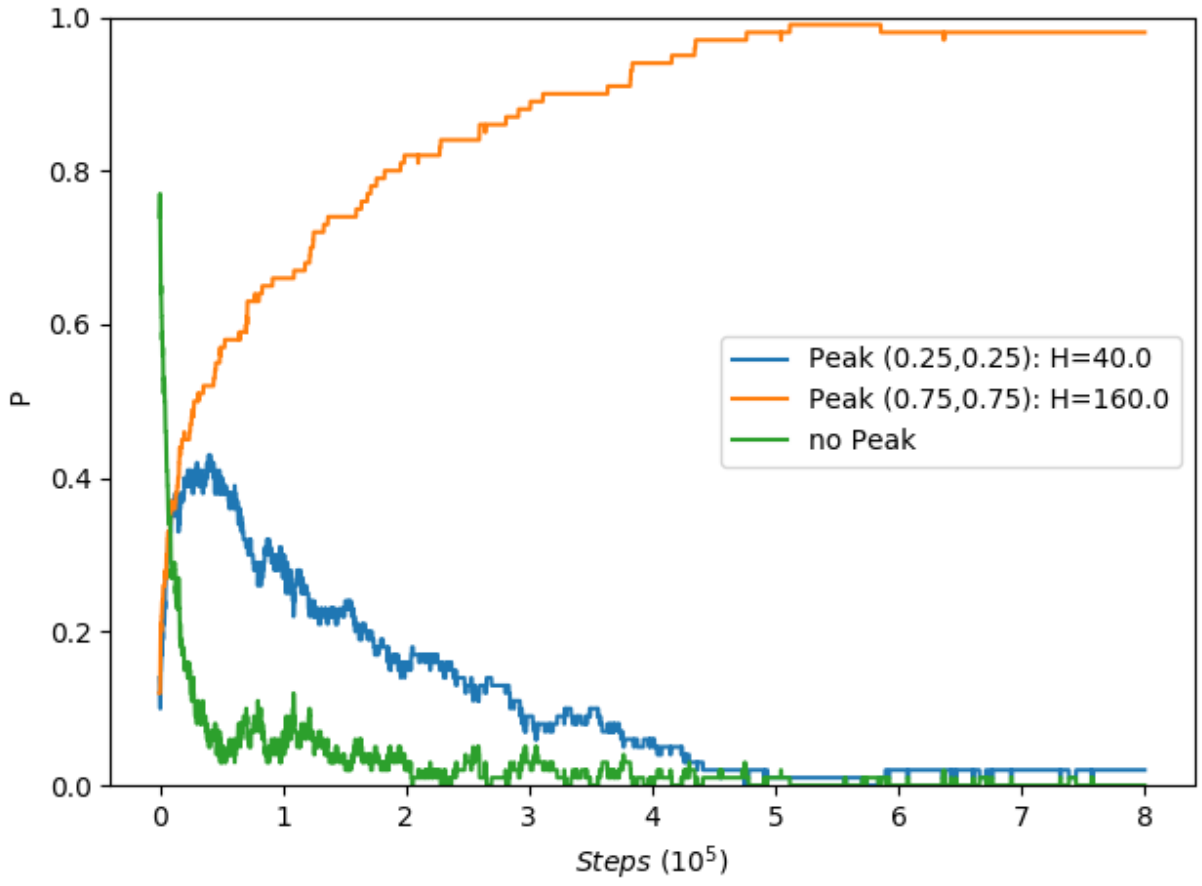

Figure 3.3: Evolution of the proportion P of all sequences in either one of the two peaks or outside any peak over $8 \cdot 10^5$ mutation steps. Most sequences will eventually end up in the higher peak (H=160) with some stochastic fluctuations. The data was created from 100 sequences, $s_c = 0.001$, $s_e = 0.01$ and $p_c = 0.9$.

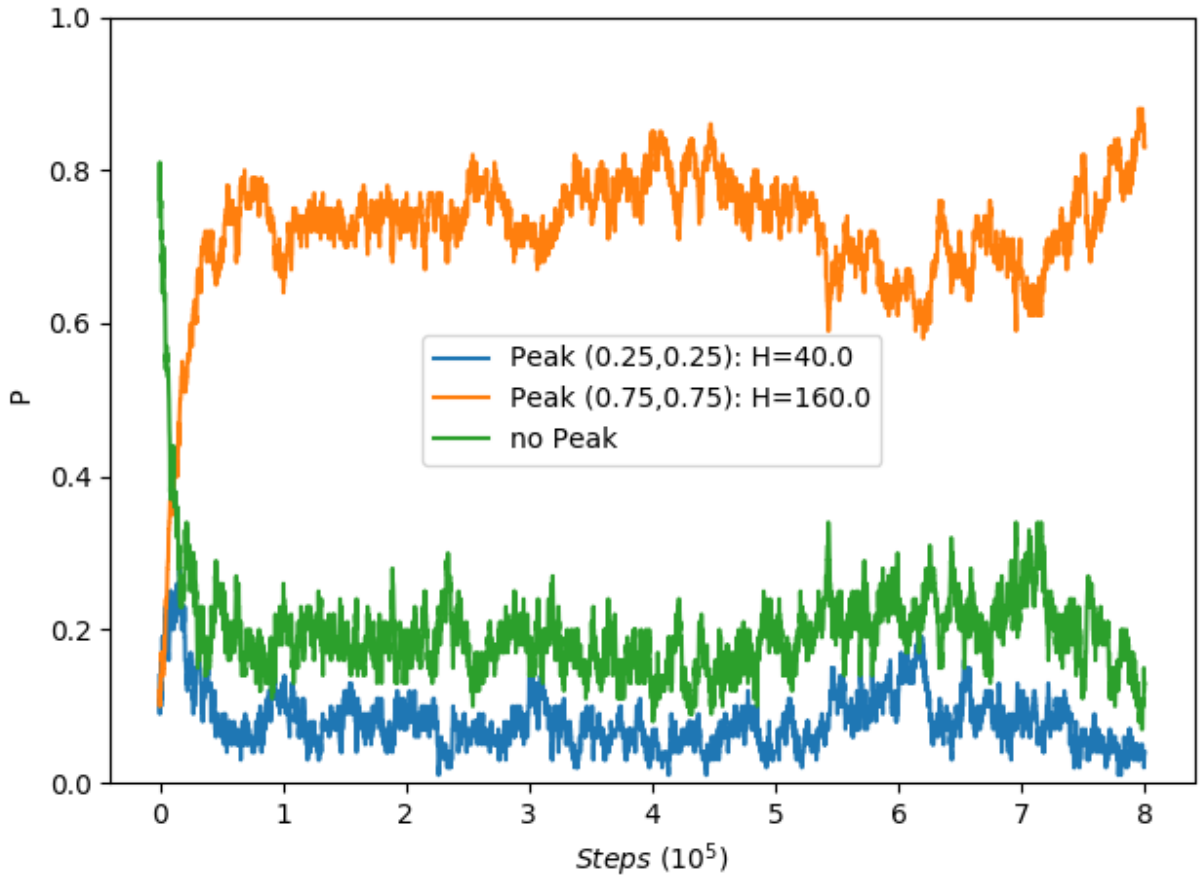

Figure 3.4: Evolution of the proportion P of all sequences in either one of the two peaks or outside any peak over $8 \cdot 10^5$ mutation steps. Most sequences will eventually end up in the higher peak (H=160) with some stochastic fluctuations. The data was created from 100 sequences, $s_c = 0.001$, $s_e = 0.01$ and $p_c = 0.78$.

So far the model uses two different mutation step sizes with the outcome being a ratio of how often each occurs, but in the optimality calculations an average mutational step size was calculated and found to be similar across reading frames. In order to verify whether the average mutation step size can optimise the average fitness of a sequence, average fitness values over a range of parameters are collected, see Fig. 3.5. Three quantitatively different areas can be observed. In area (I), sequences cannot escape either of the two fitness peaks in the given number of mutations. The opposite happens in area (III), where none of the two peaks can conserve the sequence for a long time and stochastic fluctuations dominate the system. In the small area (II) in between areas (I) and (II), only the higher peak can conserve the sequences while the lower peak cannot, just as shown in Fig. 3.3. Fitting the average mutation step size $\bar{s} = p_c s_c + (1 - p_c) s_e$ to the area (II) shows that it is indeed a reasonable approximation.

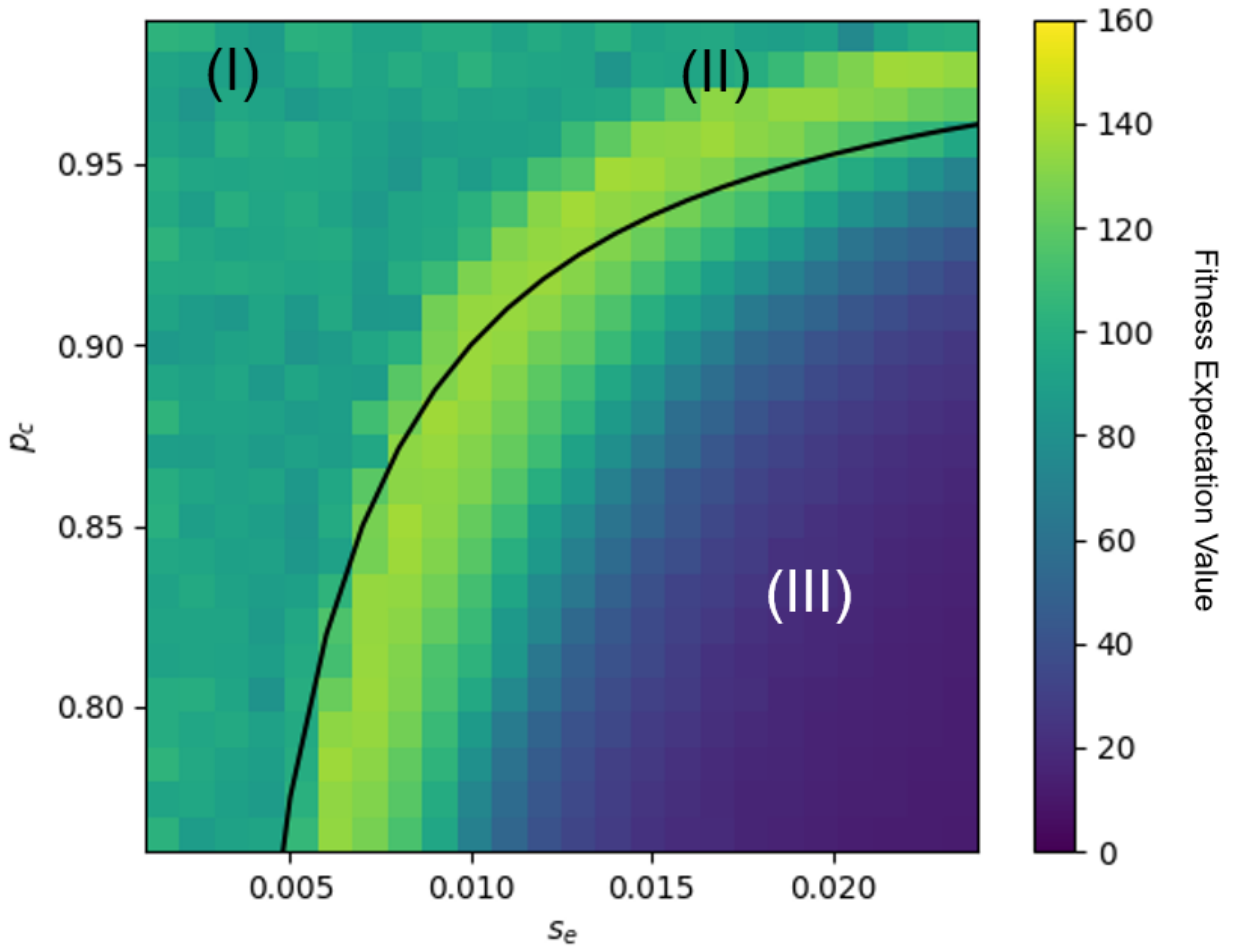

Figure 3.5: Average fitness value of 100 sequences after $8 \cdot 10^5$ mutations for different values of p_c and s_e. The conservative mutation step size is fixed to $s_c = 0.001$. Fitness expectation values fall into three regimes. In the first (I), sequences are conserved in both peaks, in the second (II), sequences are restricted to the higher peak, and in the third regime (III), sequences are conserved in neither peak. Keeping the average mutation effect size at a constant value (black line) we find almost the same functional relation between p_c and s_e, which optimises the expected fitness.

3.3. Discussion

In this model the average mutation step size optimises the fitness of sequences in a search of the sequence space to a good approximation. Optimal fitness values can be found across a wide range of different parameter sets, so within an optimisation framework it is reasonable that this property is optimised in different reading frames with most likely very different distributions of mutation step sizes. This is only a toy model and should only be seen as a proof of concept. Whether a similar result can be obtained from a more realistic model is not clear. For example, natural selection only affecting conservative mutations is a quite arbitrary rule used to incorporate the two forces of conservation and exploration. On the other hand achieving such a

similar function dependency of the parameters which optimise the average fitness value is quite astonishing and appears unlikely to be a chance event.
In this study the area surrounding the two fitness peaks had no fitness value at all, indicating that sequences are not functional in this region, but it could also represent the base level of function around two local maxima. According to this hypothesis, the average mutation step size ensures a minimum fitness advantage to the organism in this first evolutionary step after gene creation.
While it is not yet understood how the rare functional sequences can be found by random mutations in the enormous sequence space, the probability of finding new genes cannot be as small as perceived today. The adaptability and diversity of cellular life makes it quite clear that *de novo* gene creation is a fundamental feature of life [113], [74], [111]. An adaptation of the genetic code to this central property of life could be seen as simply to be expected or even essential as the genetic code might be one of many factors that make *de novo* gene creation possible. Another issue that deserves further attention is the idea that when creating genes parallel to an existing gene, the MG could provide a kind of template for functional genes rather than a restriction. Further, judging the abundance of functional sequences on the rarity of function for modern gene sequences, which are like highly optimised machines, might be the wrong approach. The first proteins were most likely much simpler and these simple functions might be much more abundant in sequence space. Navigating between these barely functional genes might be what the SGC has been optimised for. Much remains to be explored here, but we have laid some reliable groundwork.

4. OLG Construction Theory

The existence of OLGs has long been questioned due to informational constraints [115]. In order to gauge how much of an obstruction the MG is on a potential new gene, this project aims to study how easy it is to encode two arbitrarily chosen genes parallel to each other in different reading frames. If encoding two genes in an overlapping manner without changing their AA sequence, only very short overlaps can be obtained for arbitrarily chosen natural sequences [132] or only very specific genes can be overlapped significantly [133]. Much longer overlaps can be achieved when permitting some change to the genes [132], but it is not clear whether the genes are still functional afterwards. Only in a recent study [134], has the functionality of sequences that had been changed in order to overlap completely, been bioinformatically assessed on a large scale. In that study, protein domains were rewritten by a novel algorithm to completely overlap while minimising the change to each sequence. Of 125,250 protein domain pairs 16% remarkably passed their threshold for functionality. Each pair was tested in two positions and three different alternative reading frames and a pair was labelled as a successful overlap if at least one of the six overlaps passed the threshold. Assuming that OLGs are mostly important for viruses, a taxonomic split revealed that successful virus domain overlaps are much more likely than non-virus overlaps. Overlaps were judged as successful if a BLAST search of the SWISS-PROT database resulted in a hit with at least 85% match length and a maximum e-value of 10^{-10} for both OLGs. The results from this study suggest that it is not nearly as difficult to create functional OLGs as widely assumed. This finding, if reliable, appears to have significant implications for synthetic biology and our understanding of gene origins.

The critical aspect of their analysis is judging whether an artificial sequence is still functional only from its AA sequence, which is a very sought after technique and a very difficult problem. While protein structure can now (thanks to some recent impressive work) be reasonably predicted with much effort [135], [136], [137], it is only the first step of functional prediction as binding sites can be rendered useless by an AA change without changing protein structure. The only fully reliable functional verification today remains real experiments, which are usually much more expensive than bioinformatic studies. Nevertheless the latter can still guide experiment by filtering potential candidates, e.g. by determining their similarity to sequences with known functions as done in [134].

In the study reported here OLGs are created according to the same algorithm as in [134], but the evaluation is improved using Hidden Markov Models (HMMs), which was necessary since their approach created artefacts in the results as will be discussed below in more detail. Another study which we were not aware of until recently independently also did a followup study using HMMs [138] and further took into account intra protein interactions in order to create sequences for laboratory experiments. The first part of the study reported here, in contrast, is purely bioinformatic and focuses on the average amount of change a sequence has to go through in order to overlap with another sequence. The measures used are AA identity and similarity between original and altered sequence, HMM profile scores and secondary structure. By assessing differences between reading frames, taxonomic differences, evolutionary distances, the influence of the SGC on OLG creation and the optimality of the SGC, a theoretical foundation for different kinds of research on OLGs is established. While the evolution of

naturally occurring OLGs cannot be determined this way, the results can still be used as a first approximation in this mostly unexplored field. The results are published in [139].
Here protein domains are overlapped using the algorithm from [134] but covering all five alternative reading frames. While the overlap position is random, the shorter sequence is always fully embedded into the longer one. The sequences are taken from random protein domain families in the Pfam database. The property determined here is the average success rate of OLG design as opposed to the upper limit determined in [134], since it is more meaningful for naturally occurring OLGs, which are the focus of this study, as opposed to the different task of optimising synthetic genes. The results are nonetheless conservative as overlapping two protein domains is a “worst case scenario”. Overlapping one protein domain and a more flexible part of a gene, e.g. a disordered region or that which is used in protein folding but not other interactions, in contrast, has been proposed to be more frequently found in nature [140], [141]. Nevertheless, what is labelled as a successful overlap in this study does not claim definitively to maintain function of the original proteins but rather passes a threshold of similarity to known homologs.

4.1. Artefacts in the previous results

Before describing the methods used in this study, the weaknesses of the previous study [134] are discussed in order to understand the rationale of the alternative methods used in our study. While the algorithm developed in [134] works as intended, the specific results obtained in their evaluation were all artefacts determined by their choice of input protein domain sequences. Nevertheless, the general result that constructing OLGs is easier than expected still holds true.

4.1.1. Dataset-database dependencies

The dataset of sequences used for creating OLGs in [134] is a collection of sequences from the Pfam database. Not all of these sequences are part of the SWISS-PROT database as an exact copy as a BLAST search shows; see left panes of Fig. 4.1. Only 15% of the non-virus genes and 70% of the virus genes had a match sequence identity of over 80%. Consequently, OLGs constructed from this dataset have a much lower success rate in comparison to a curated set of sequences, which all have an exact copy in the SWISS-PRO database, see right panel of Fig. 4.1. Virus genes from the dataset of [134] are much better represented in the SWISS-PROT database compared to non-virus genes, explaining the higher success rate for OLG construction in virus genes reported in their study. In the curated dataset the difference between virus and non-virus genes vanishes. The extremely high success rate for OLG construction of over 95% in the curated dataset is a hint that the evaluation used in [134] is too relaxed, but could also mean that OLG construction is much less disruptive on sequence quality than expected.

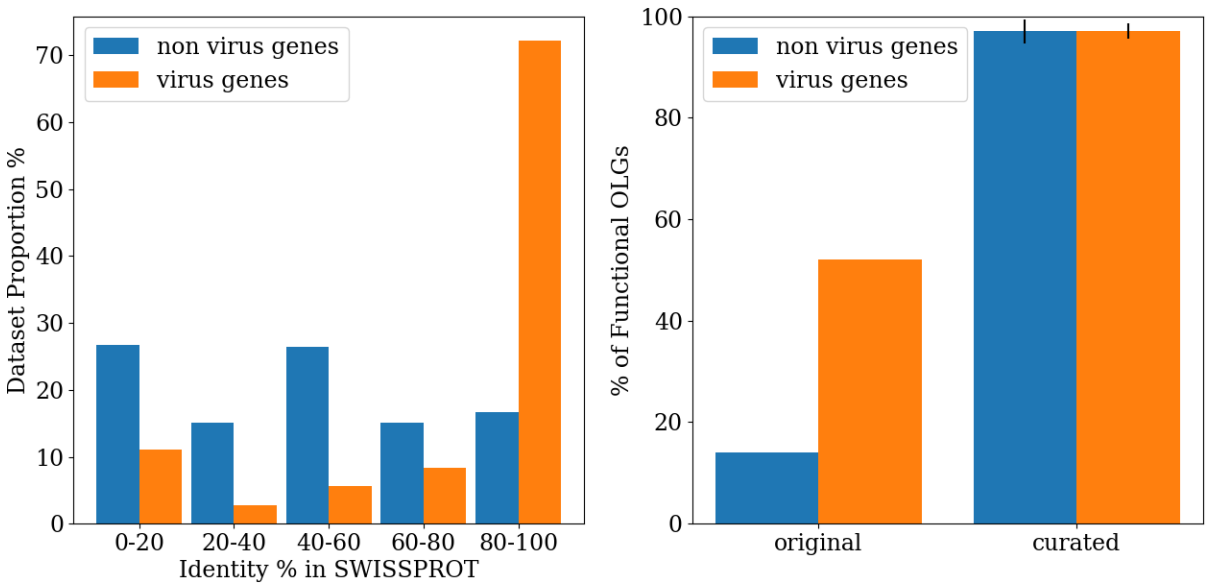

Figure 4.1: Dataset-database biases in [139]. *Left:* Frequency of match identities of sequences from the dataset used [134] in the SWISS-PROT database. Virus genes are much better represented than non-virus genes. *Right:* Success rate of OLG construction in the set of sequences used in [134] and the average of 10 curated sets of 100 sequences with an 100% identical match in the SWISS-PROT database. The sequences of the curated set have sequence lengths of 70-100 AAs just as in the set used in [134]. The difference between virus and non-virus genes vanishes in the curated set. The figure has been taken from [139].

4.1.2. Length dependencies in BLAST

The value determining a successful OLG in [134] is the e-value of the BLAST search, which is an expectation value attached to a sequence match in a database indicating how often a match of that quality is expected by chance in the database used. It is often used as it has an understandable meaning in contrast to the scoring calculated by the search algorithm and from which the e-value is calculated using the database size. It is seldom clear which database to use in such searches and mostly arbitrary collections of known and sometimes curated genes as in the SWISS-PROT database are used, so the e-value is convoluted by this arbitrary parameter. Consequently a sensible e-value cutoff is difficult to define. Mostly a seemingly very conservative value, e.g. 10^{-10} as in [134], is used. While the e-value is a good indicator whether a match in a database search is a chance event, as per its definition, it is not a good measure for the similarity of the two matching sequences as the e-value is strongly length dependent. Considering two matches with the same AA identity between the search sequence and the target sequence in the database, the longer sequence will have a much lower e-value. As such, very long sequences will pass a certain e-value cutoff more easily even if the AA identity of the match is much lower.

The OLG construction algorithm on the other hand is not expected to do worse on a longer sequence and will change a similar percentage of AAs in the sequence in most cases. While

that means that the total number of changes inflicted on the sequence is higher for longer genes, a BLAST search will calculate a much lower e-value for such sequences resulting in them passing the threshold more easily. Doing a BLAST evaluation with datasets of different length intervals reveals a strong length dependence of the e-values as expected. This explains the extreme success rates found in the curated dataset, as constructed sequences with a length of 70-100 AAs mostly have e-values scoring better than 10^{-10}.

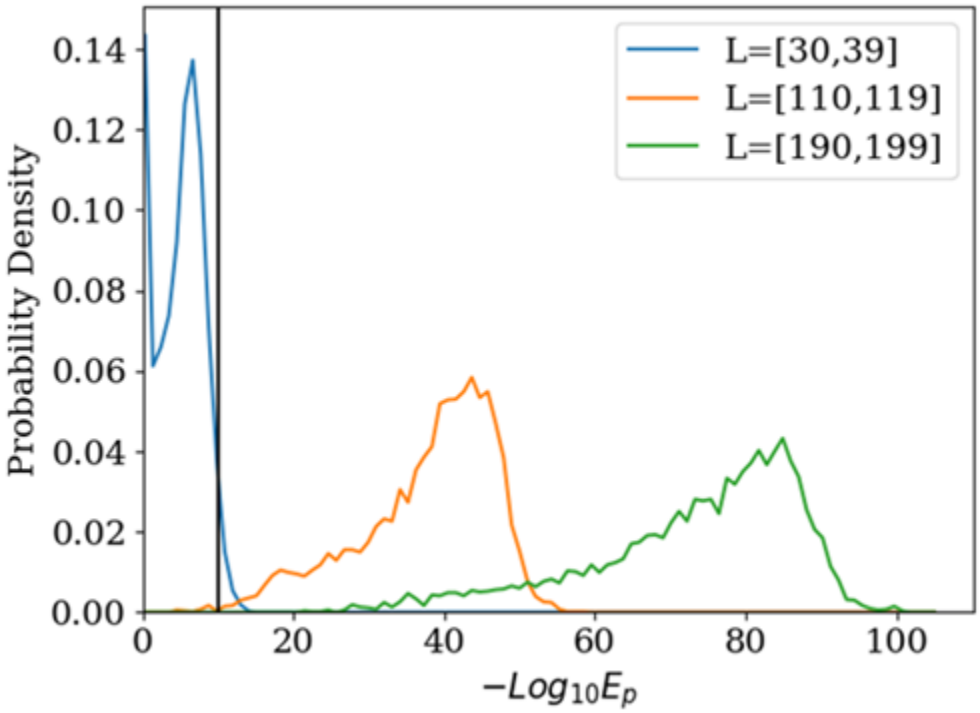

Figure 4.2: Distributions of e-values in constructed OLGs of different lengths. The e-value of 10^{-10} as used in [134] is indicated by the black line. Datasets of different length have very different e-value distributions. Figure taken from [139].

4.2. Methods, tests and parameter optimisations

A more reasonable compilation of sequences for OLG construction as well as a new measure for successful OLGs has been developed in this study, which yields a solution to the length dependence of the BLAST evaluation. The length dependence of the new evaluation is tested to verify the method and determine the optimal length for sequence selection. Furthermore, a parameter of the construction algorithm is able to be optimised - it is shown that the average success rate can be reasonably determined from a single random overlap position in each OLG pair.

4.2.1. Choosing sequences for OLG construction

The Pfam database used in the previous study to collect sequences for OLG construction [134] has two parts, namely the ‘seed’ database, and the ‘full’ database [142]. The former consists of

a curated set of protein domain sequences for each family used to create a HMM profile. A HMM in this context is a probabilistic model used to determine whether a protein sequence is part of a family of sequences related by common descent [143]. It incorporates the differing conservation of sites along a sequence, namely which parts are especially conserved or flexible. These profiles are used in the Pfam database to create the ‘full’ database by clustering sequences of the Uni-Prot database by similarity. While the ‘seed’ database is used to collect sequences for OLG construction in [134], it is used to create HMM profiles, using HMMER3 (v3.2.1) [144], in this study. The sequence of the ‘full’ database with the highest score against the HMM profile of its family is used as a starting point for OLG construction. This ensures that the chosen sequence is a typical sequence of the protein domain family. A random sequence from either the ‘seed’ or the ‘full’ database could be an outlier among its family, which is less likely to create functional sequences after OLG construction. Similar to how the sequences of the original dataset were not well represented in the SWISS-PROT database leading to artefacts in the results, starting with an outlier would create a negative bias for the OLGs constructed from it. A major advantage of HMM profiles is that they can be used to determine the ‘most typical’ sequence, i.e. the one with the highest score, which is not possible using BLAST.

4.2.2. Solving the length dependency with a relative threshold value

Fundamentally, using the same threshold value for OLGs of different lengths creates arbitrary results depending on the threshold value. A relative threshold value determined for each protein domain independently is a much more reasonable approach. This can be done by comparing the constructed OLG sequence with homologs from the same protein domain family, using the ‘full’ database. A constructed OLG is labelled successful if its score against the HMM profile of its respective protein domain family is higher than a chosen percentile of the sequences in the ‘full’ database of the same protein domain family. When scoring an OLG sequence against a HMM profile, only the part overlapping with the other gene is used, so the parts of the longer sequence that do not overlap will be cut out. This way only the changed parts of the sequences are evaluated, which is important to ensure no artefacts arise in OLG pairs with very different sequence lengths. Percentile values define a threshold score value individually for each protein domain family, which is assumed to have roughly the same length, therefore taking care of the length dependence. In order to sensibly compare scores of sequences with different lengths the score is divided by sequence length before each comparison. By using scores, this approach removes the arbitrary factor of database size from the results. Furthermore it takes into account that some protein domains have a wider spread of scores against their HMM profile as only a certain percentile has to be reached.

While most properties are studied for a wide range of threshold percentiles, two special percentile values are highlighted, namely the 50th percentile (median) and the lower 5th percentile. The former resembles a threshold for typical sequences in a protein family and an OLG passing this threshold is so similar to naturally occurring proteins that HMM profiles can no longer reasonably distinguish them from each other. The 5th percentile is used as the threshold for ‘biologically relevant’ sequences. Constructed OLG sequences scoring better than any of the sequences in the ‘full’ database are expected to be biologically relevant as they reach the

similarity of naturally occurring homologs. The 5th percentile is used in order to avoid outliers and which might be sequences that have been wrongfully sorted into a protein family.
The OLG construction algorithm attempts to balance the changes inflicted on both sequences when creating the OLG pair. In special cases it can happen that most of the change is inflicted on one sequence, but an OLG pair only makes sense if both sequences are functional. Consequently, both OLG sequences must be judged together for a sensible result. Following [134], the conservative solution of judging an OLG pair by the worse of the two sequences is used. The worse sequence is determined by the lower value of $\frac{S}{S_{threshold}}$, with S being the score of the OLG sequence and $S_{threshold}$ being the threshold score value for a given percentile. If not stated otherwise OLGs are always judged this way in this study.

4.2.3. Workflow and taxonomic filtering

The workflow for sequence selection for OLG construction and the following evaluation described above can be summarised as in Fig. 4.3. The OLG construction algorithm uses conservation weights indicating how much each position of each sequence varies in natural homologs in order to improve the quality of the constructed sequences. The weights of the sequence used for OLG construction are calculated from an alignment of the ‘seed’ sequences used in the HMM profile with the ‘best’ sequences of the respective protein family. MAFFT (v7.419) [145] was used for adding the ‘best’ sequence to the existing alignment of the ‘seed’ sequences of the respective protein family. In this study the influence of the conservation weights on the OLG construction success rate is studied in order to further optimise the algorithm.
In order to study taxonomic differences in OLG construction both the ‘seed’ and the ‘full’ Pfam database have to be split by taxonomic groups. Here only the four most basal groups are considered, namely archaea, bacteria, eukaryotes and viruses (although the biological relevance of classing all viruses together is debatable, we follow the initial study in this regard [134]). HMM profile creation, sequence selection and threshold calculation are performed just as before using the taxonomically filtered Pfam databases. HMMER3 needs an alignment of sequences in order to create a HMM profile, so the originally already aligned ‘seed’ sequences are realigned using MUSCLE (v3.8.31) [146] in order to improve the alignment. From the ~17000 Pfam families only those with at least 10 ‘seed’ sequences and 4 ‘full’ sequences are considered to ensure that thresholds, conservation weights and HMM profiles can be reasonably defined.

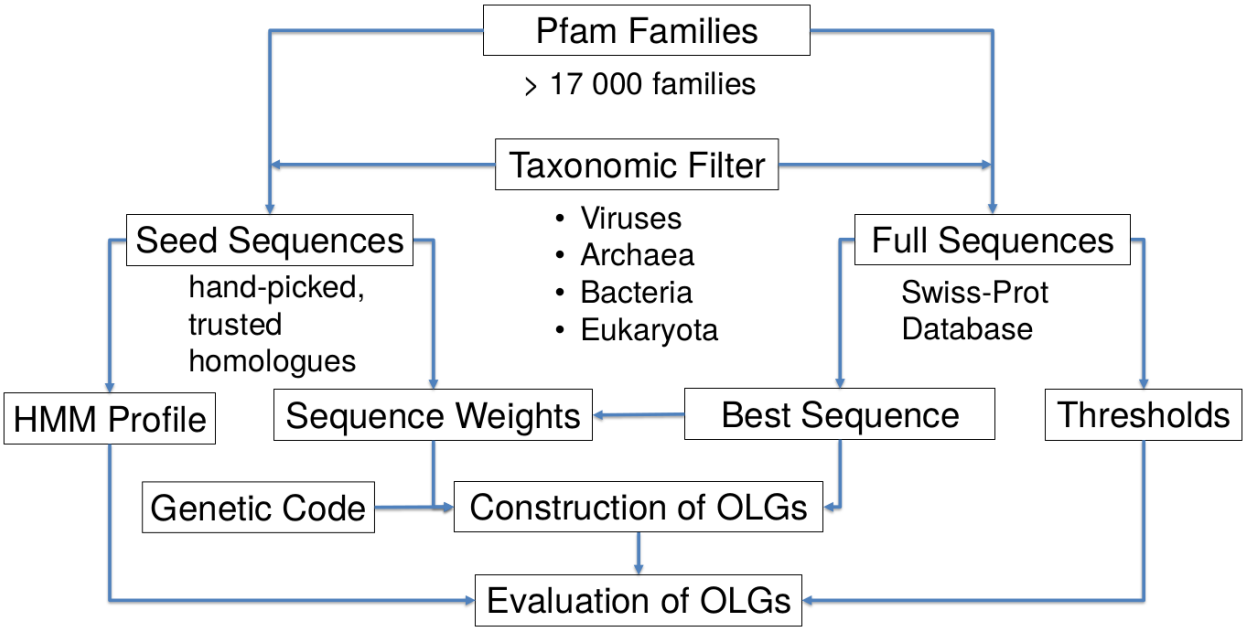

Figure 4.3: Summarised Workflow for OLG construction and evaluation. All data is taken from the Pfam database. HMM profiles used to define the thresholds are created from the 'seed' sequences. The thresholds are calculated from the scores of 'full' sequences in their respective HMM profile. Figure taken from [139].

4.2.4. Testing the length dependency of the relative HMM scores

The length dependence of the improved OLG evaluation can be tested at a threshold percentile. Here the 100th percentile is used, for which the threshold score is defined as the score of the original sequence, which is set up as the highest score S_{max} of the 'full' group of the Pfam database. This percentile has another special meaning as it determines the percentage of the score lost due to the overlap and is therefore also called the OLG quality Q. For an OLG sequence with score S the OLG quality is defined as $Q = 100 \cdot \frac{S}{S_{max}}$. An OLG pair is represented by the lower Q value of the two sequences. Comparing distributions of Q values for OLGs with different lengths, a domain of length independent results can be determined, see Fig. 4.4. Sequences with at least 70 AAs are sufficiently independent of sequence length. The dependence of shorter sequences is due to them not being recognised by their respective HMM profile resulting in a score of 0. While this happens for all sequence lengths, it is much more prevalent for short sequences. Shorter sequences have lower absolute scores but also larger fluctuations in their scores as single AA changes result in a larger percentage of the total sequences to change. The most likely explanation for shorter sequences often not being recognised by their respective HMM profile is that they fall below internal score thresholds of HMMER3 more easily and are not considered in the output. Changing internal parameters of HMMER3 did not increase the percentage of short sequences being recognised. Among sequences above 70 AAs less than 5% are not recognised by the HMM profile for the family, which is deemed acceptable for this study.

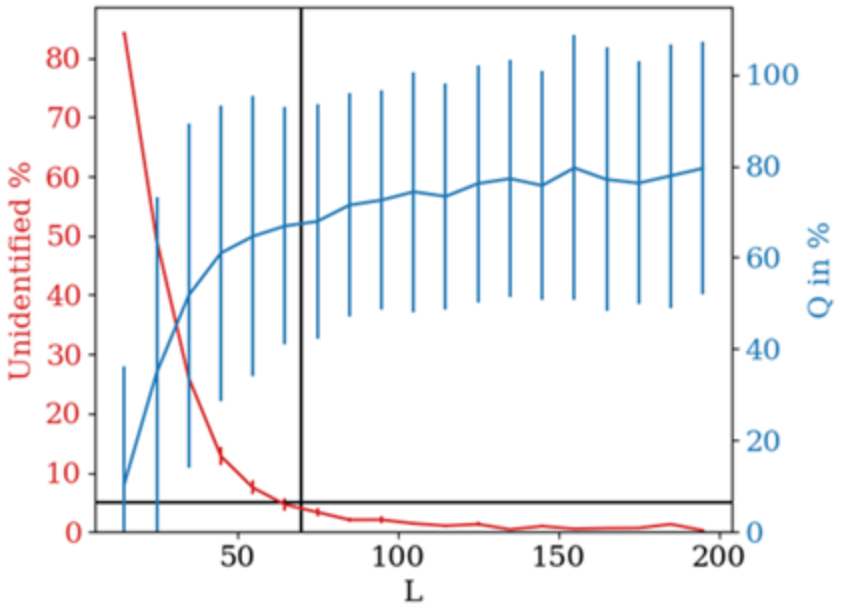

Figure 4.4: OLG quality Q distributions for different sequence lengths L represented by their mean value and standard deviation (*blue line*). Prevalence of OLG sequences not recognized by their respective HMM profile decreases significantly with sequence length (*red line*). At a sequence length of at least 70 AA (*vertical black line*) less than 5% (*horizontal black line*) of OLG sequences are not recognised by their HMM profile. The data was created from 20 datasets of 150 sequences for each length intervall. Figure taken from [139].

Calculating the average Q distributions but including the values of both sequences in an OLG pair, another piece of evidence that the improved evaluation is independent of length but a minimum length must be retained, is found. Distributions for larger sequence lengths start to converge to a distribution with an average of Q=76%, see Fig. 4.5. This distribution reveals the effect of OLG construction on the HMM score and is due to the different levels of flexibility in different protein domains but also how well two sequences fit together.

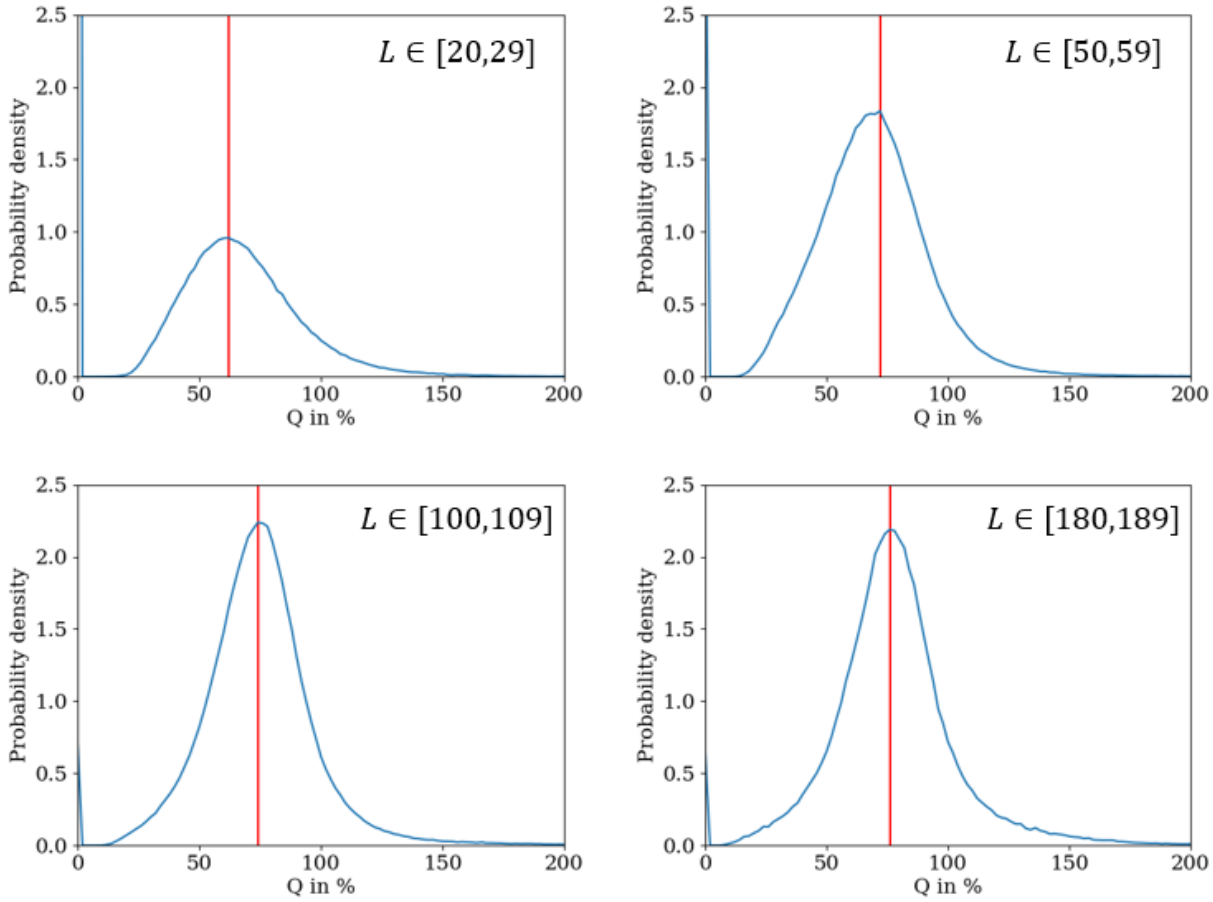

Figure 4.5: Averaged distributions of Q values of both sequences of an OLG pair for different sequence lengths. The distribution converges to a seemingly symmetric peak (red line) around 76%. The distributions of 20 datasets with 150 sequences each are averaged over for this data. Bottom right figure taken from [139].

4.2.5. Calculating the average success rate

Since only fully embedded overlaps are considered in this study, it has to be noted that the maximal number of different overlap positions is limited for each sequence pair. The AA length difference of two genes to be overlapped plus one is the maximal number of different positions available. The overlap positions will be chosen randomly from the available positions. In order to determine reliable values, averages will be created over multiple datasets, but also the number of sequences in each dataset as well as the maximal number of overlap positions is optimised to minimise fluctuations. The percentage of successful positions, the average success rate for each reading frame as well as the average overall success rate across reading frames is calculated after the success of each position of each OLG pair is determined.

Each sequence in a dataset is overlapped with itself and every other sequence in the dataset. In order to determine a sensible number of sequences in each dataset as well as a sensible number of random positions to overlap each sequence pair, their influence on the variability of the average success rate is determined, see Fig. 4.6. Above 150 sequences in each dataset,

there is no further reduction in the variance of success rate between datasets and larger sets are not sensible when considering computational resources, see left panel of Fig. 4.6. While a large relative reduction in variation can be achieved by increasing the number of overlap positions for each sequence pair, the absolute values of the variation are very small compared to the variation due to dataset size, see right panel of 4.6. Consequently, one random overlap position is enough to estimate the average success rate of OLG construction. This indicates that the influence of two sequences having a good 'fit' for overlap is much smaller than the variability of protein domains collected in the dataset in this evaluation.

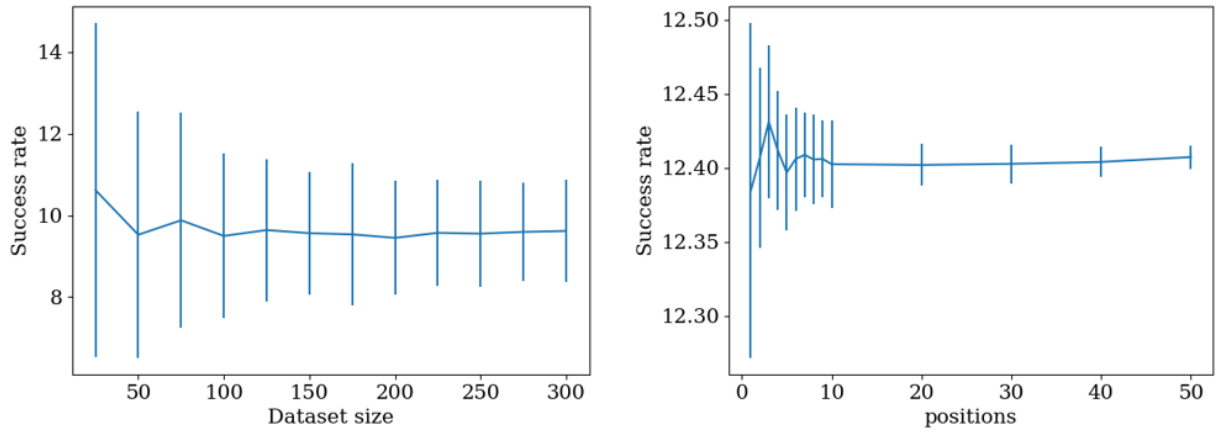

Figure 4.6: Success rate variations. *Left:* Variation of success rates as a function of dataset size. The bars represent the standard variation. The data is calculated from 100 datasets of each size and sequences are overlapped in one random position. *Right:* Variation of success rates as a function of the number of overlap positions for each sequence pair. The bars represent the standard variation. The data is calculated from 30 datasets with 150 sequences each size. A threshold percentile for successful overlap of 50% as well as a minimum sequence length of 70 AAs is used in both calculations. Figures taken from [139].

4.2.6. Optimising the influence of conservation weights

The algorithm for OLG construction developed in [134] optimises a total score for each overlap by searching the codon which contributed the most to this score for every position. Overlapping sequences X and Y and creates sequences X' and Y' with a total score $S(X', Y', X, Y)$. It is calculated as the sum over the local scores $S_i(X', Y', X, Y)$ for each position i, see eq. (23). The total score is maximised if each individual score is maximised, which is done by the algorithm by calculating the local score for each codon and calculating the score according to (24). E(a,b) is any AA exchange matrix scoring changes between AA a and b. In [134] the Blosum62 exchange matrix [30] is used, which is also used in this study. While a different exchange matrix could be used it is important to make sure that the program used for calculating sequence similarity in the evaluation, e.g. BLAST or HMMs, uses the same matrix if it needs one. If different matrices are used for construction and evaluation, lower success rates will be reported and the results will not be meaningful. The exchange matrix scores for both sequences of the overlap are weighted by conservation weights p_i and q_i indicating the importance of the current

position for the protein domain. This should increase the quality of the resulting OLGs. Codons resulting in stop codons for any of the two constructed sequences will not be considered.

$$S(X', Y', X, Y) = \sum_{i=0}^{L} S_i(X', Y', X, Y) \tag{23}$$

$$S_i(X', Y', X, Y) = p_i E(X_i, X'_i) + q_i E(Y_i, Y'_i) \tag{24}$$

While this approach is straightforward in the '-1' frame where one codon in the '+1' frame completely defines one codon in the '-1' frame, more effort is needed to use this approach for other reading frames. In these reading frames at least four consecutive nucleotides are needed to completely define a codon pair in the '+1' and an alternative reading frame. Similarly to splitting overlaps in the '-1' frame into a sequence of nucleotide triplets and optimising each one, the sequence is split into a sequence of quartets in other reading frames [134]. These quartets are no longer independent of each other and share their first and last nucleotide with their neighbouring quartets such that the last nucleotide of a quartet is the first of the subsequent one. In order to optimise this sequence, the algorithm actually optimises four different sequences, each ending on a different nucleotide. In every step 64 quartets each starting with the same nucleotide are attached to each of the four sequences and their score calculated. The four sequences with the highest scores ending on the four different nucleotides are saved for the next position. At the end of the overlap the single sequence with the highest score is the chosen OLG sequence. In reading frames '+3' and '-2' the sequence in the '+1' frame can be constructed from left to right, while in reading frames '+2' and '-3' the sequence in the '+1' frame has to constructed from right to left in order for the quartets to include nucleotide triplets in both reading frames.

The conservation weights in [134] are calculated from sequence alignments. Grouping AAs into six groups just as in [134], namely {LVIMC}, {FYW}, {G}, {ASTP}, {EDNQ}, and {KRH} according to the single letter symbols of the AAs, see table 1.2, the entropy s_i of each position in the alignment can be calculated. The conservation weights in [134] are then defines as $p_i = e^{-s_i}$ and can only take values in the range 1 to ⅙ by construction. Ultimately only the relative weight of the two positions matters as the absolute value of the score is not important and one weight can be factored out of the sum in (24). Consequently one exchange matrix score can at most be weighted 6 times more as the other. In order to control the strength of the relative weighting, a factor $k \geq 0$ is added into the weight calculation such that $p_i = e^{-ks_i}$, with $k = 0$ indicating no weights are being used.

After calculating the quality Q for different k values an optimal weight strength can be derived, see left panel of Fig. 4.7. While the weight strength only has a very small influence on the result, an optimum can be found at $k = 0.5$ for an evaluation with HMMs. Larger k values correspond to a stronger influence of sequence conservation. Interestingly, for extreme large k values the quality of the constructed OLGs goes to zero. In this case the weighting is so strong in the algorithm that the AA of the more conserved sequence will always be maintained no matter the cost in the other sequence. This showcases that changing both sequences in some positions is unavoidable in order to design sensible OLGs.

Doing the same analysis in BLAST but measuring the success rate of OLGs with shorter sequences in order to have a reasonable effect size, $k = 0$ is the optimal value, see left panel of Fig. 4.7. Despite conservation weights increasing biological relevance, introducing conservation weights always has a negative effect in the BLAST evaluation. While HMM profiles are statistical models which incorporate sequence conservation, the standard BLAST method only compares single sequences without including any information about conservation across the sequence. Weights in such an analysis improve one sequence at the cost of the other. Since the worse of the two sequences represents the OLG pair, the net positive effect can only be judged correctly if the conservation is considered. This is an example of construction and evaluation not taking into account the same properties, which consequently results in lower success rates.

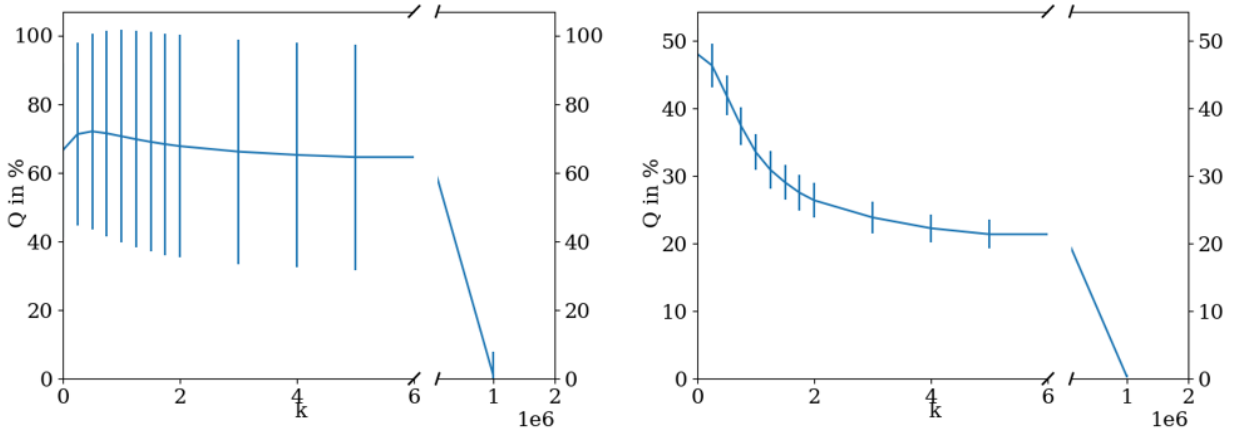

Figure 4.7: Average quality Q for different weight strengths k. The vertical lines indicate the standard deviation. *Left:* Using HMMs, $k = 0.5$ is an optimum. The data is calculated from 20 datasets with 150 sequences and a minimum sequence length of 70 AAs. *Right:* Using BLAST, introducing weights into OLG construction has a negative effect. The quality here is the percentage of OLGs above the e-value cutoff of 10^{-10}. The data is calculated from 5 datasets with 100 sequences. In order to have a reasonable effect size, sequences have a length of 40-60 AAs so that a change in sequence quality results in a change in success rate. Figures taken from [139].

4.3. Results

First the similarity of constructed OLG sequence and naturally occurring homologs is discussed in terms of AA similarity, AA identity, and secondary structure in order to gauge the significance of the following results. A wide range of biologically relevant properties, namely taxonomic differences, the evolutionary distance to OLGs, differences between reading frames, the influence of the SGC and its optimality in comparison with alternative genetic codes, is studied.

4.3.1. AA identity and similarity in OLGs

After overlapping arbitrarily chosen protein domains, both sequences can retain over 60% sequence identity and over 80% sequence similarity, see left panel of Fig. 4.8. In naturally occurring homologs, sequences with at least 34% AA identity have been found to share the

same protein structure [147]. While structure does not solely determine function it is still interesting that 96.5% of constructed sequences pass this threshold.

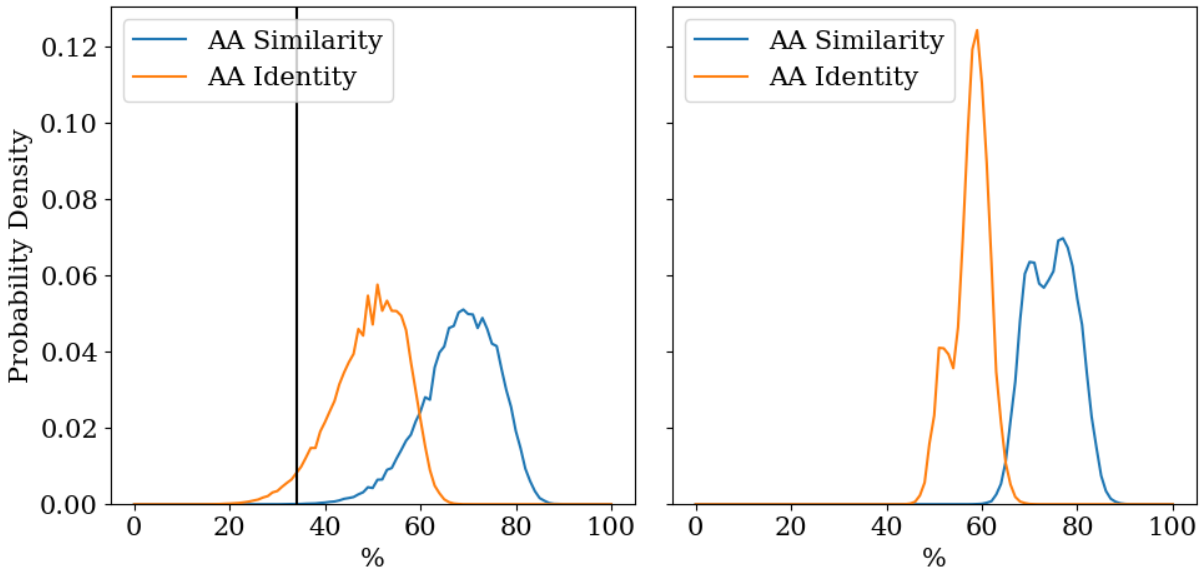

Figure 4.8: Distributions of AA identity and similarity. The probability densities are calculated from 505,000 OLG pairs. *Left:* The lower AA identity or similarity of the OLG pair is used. Genes above 34% sequence identity (black line) are assumed to have the same protein structure. *Right:* Instead of the lower AA identity and similarity, the average value of the two OLG sequences is used. Figure taken from [139].

Determining the average sequence identity and similarity of both OLG sequences, the expected impact of OLG construction on the original sequences can be determined. In most cases OLG sequences retain 60% AA identity and 75% AA similarity, see right panel of Fig. 4.8. Comparing the distributions of AA identity and similarity in Fig. 4.8, it can be concluded that in most cases one sequence is above and one below the average values of 60% for AA identity and 75% for AA similarity respectively. The case that both sequences of an OLG pair are above or below the average is very rare. The average values of both sequences in an OLG pair have even narrower peaks when split by reading frame, see Fig. 4.9. The peaks of the different reading frames add up to the double peak structure seen in the right panel of Fig. 4.8.
It can be estimated that in a typical constructed OLG pair 20% of positions preserve the AA of both sequences, while in 30% the AA of only one sequence is preserved and a similar AA can be fit in the other sequence. In the remaining 50% of positions only one original AA will be preserved but no similar AA can be fit in the other sequence. Cases in which both sequences retain a similar but not identical AA at the same position are neglected in this estimation since they are rare. Fig. 4.10 visualises this breakdown.
The narrowness of the average AA identity and similarity peaks for each reading frame individually indicate how little variability is due to sequence specific factors like the conservation of the two sequences, the overlap position and the two specific AA sequences. Instead, it is

clear that the average AA identity and similarity is mostly determined by the SGC, which is the constant factor across all constructed OLGs and also determines the reading frame differences.

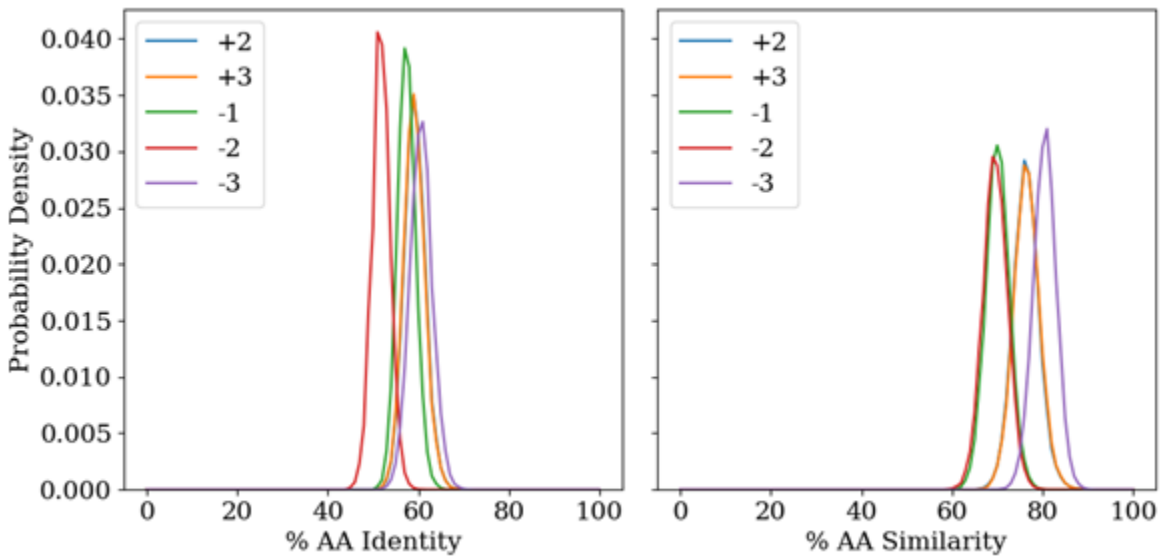

Figure 4.9: Distributions of AA identity (l*eft*) and similarity (*right*) by reading frame. The probability densities are calculated from 505,000 OLG pairs.The average value of the two OLG sequences is used. Figure taken from [139].

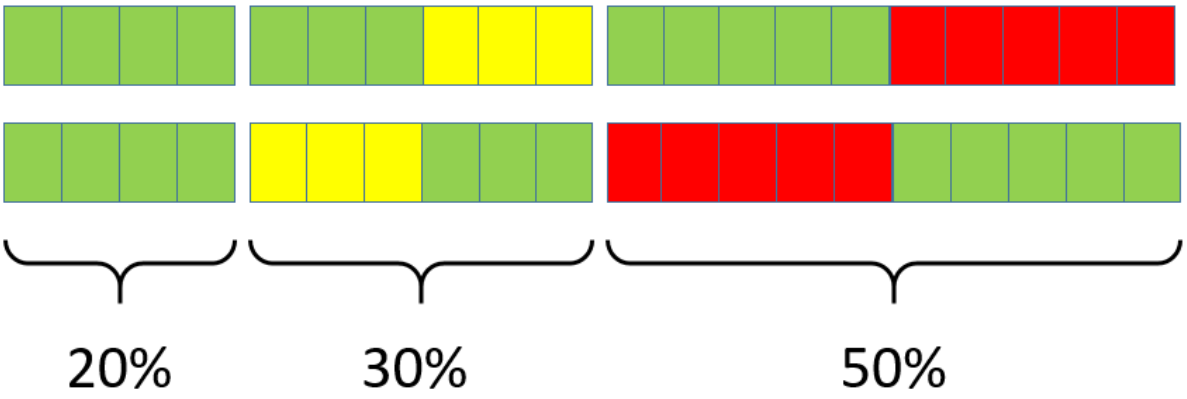

Figure 4.10: OLG construction cost breakdown. Each box represents 5% of overlapping positions. *Green* indicates that the original AA can be maintained, *yellow* indicates that a similar but not identical AA can be maintained and *red* indicates that neither a similar nor an identical AA can be maintained.

4.3.2. Secondary structure similarity of constructed OLGs

Predicting secondary structure using Porter 5 [148] (using the '--fast' flag), a similarity between the OLG and their original sequences can be determined. It predicts up to eight different secondary structure motifs from the dictionary of protein secondary structure (DPSS) [149], [150], [151], namely the 3_10-, alpha- and phi-helices, hydrogen bond turns, beta sheets, beta

bridges, bends, and coils. In order to judge the resulting similarity in secondary structure of OLGs, homologs taken from the ‘seed’ database of the respective protein domain family are compared to the sequence used for OLG construction in their secondary structure to determine their similarity. These similarities reflect the naturally occurring variations. A comparison of constructed OLG sequences and naturally occurring homologs reveals that both have very similar distributions, see Fig. 4.11. The dataset used for the secondary structure prediction only contained 50 sequences due to the computational resources required for secondary structure prediction, so larger fluctuations are expected. Nevertheless it can be concluded that the difference in secondary structure caused by OLG construction is comparable to the differences between naturally occurring homologs.

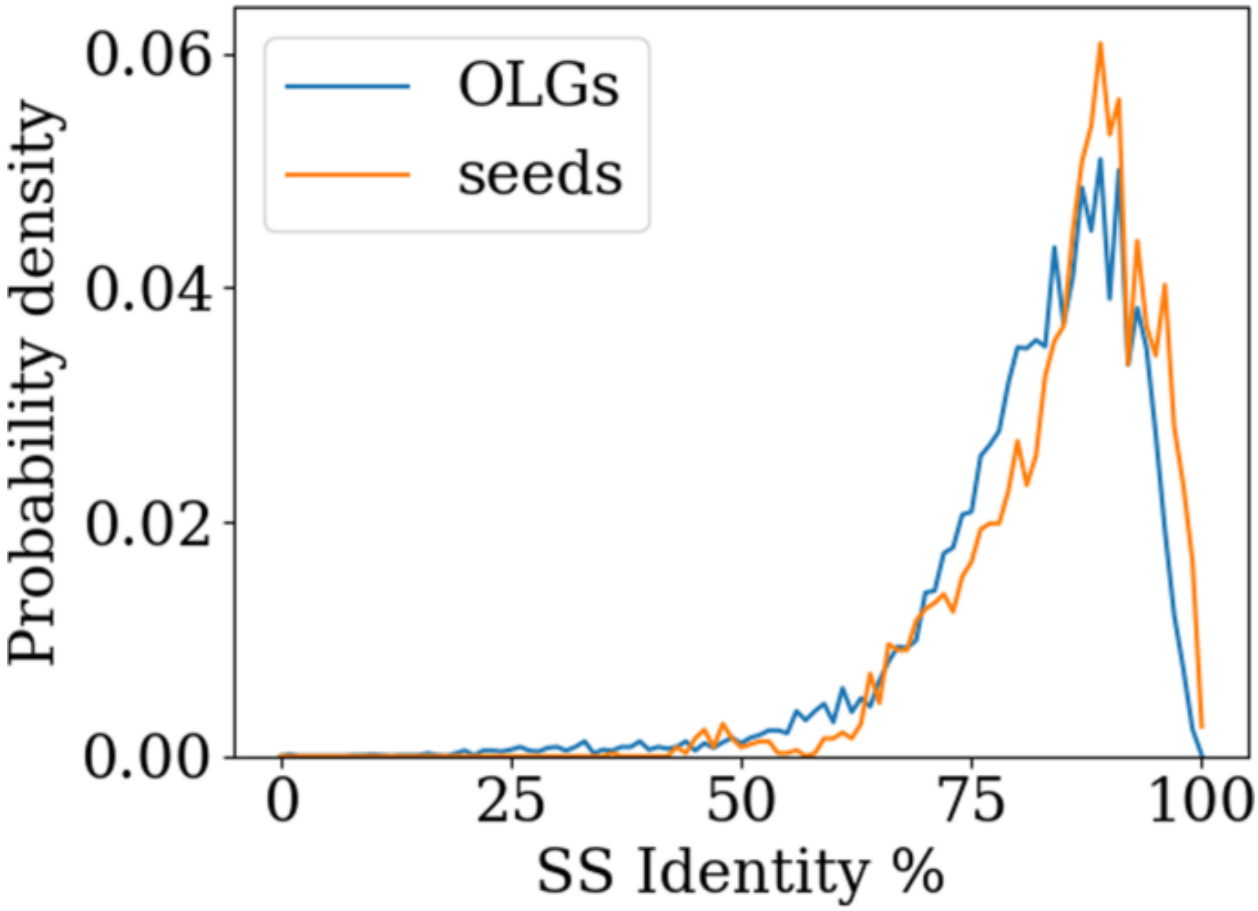

Figure 4.11: Distributions of secondary structure similarity between OLGs and naturally occurring homologs taken from the Pfam ‘seed’ database. The distributions are similar enough so that constructed OLGs cannot be distinguished from naturally occurring homologs by secondary structure. The data is calculated from 50 sequences with a minimum length of 70 AAs. Figure taken from [139].

4.3.3. Success rates for different OLG positions

In order to determine the influence of the overlap position, the percentage of positions in which a pair of sequences can be successfully overlapped in any reading frame is determined. This

percentage varies strongly across different OLG pairs, see Fig. 4.12. 50.3% of biologically relevant sequences, which score at least as high as the 5th percentile of naturally occurring homologs, can be overlapped successfully in every position, while 25% cannot be overlapped in any position. For typical sequences, which score at least as high as the 50th percentile of naturally occurring homologs, only 1.9% of pairs can be overlapped in all positions and 66.7% cannot be overlapped in any position. The range of successful positions can be anything from 0% to 100% depending on the sequences used for OLG construction for both threshold values. But the position still matters as values in between are very likely and it is not a distribution with narrow peaks at 0% and 100%.

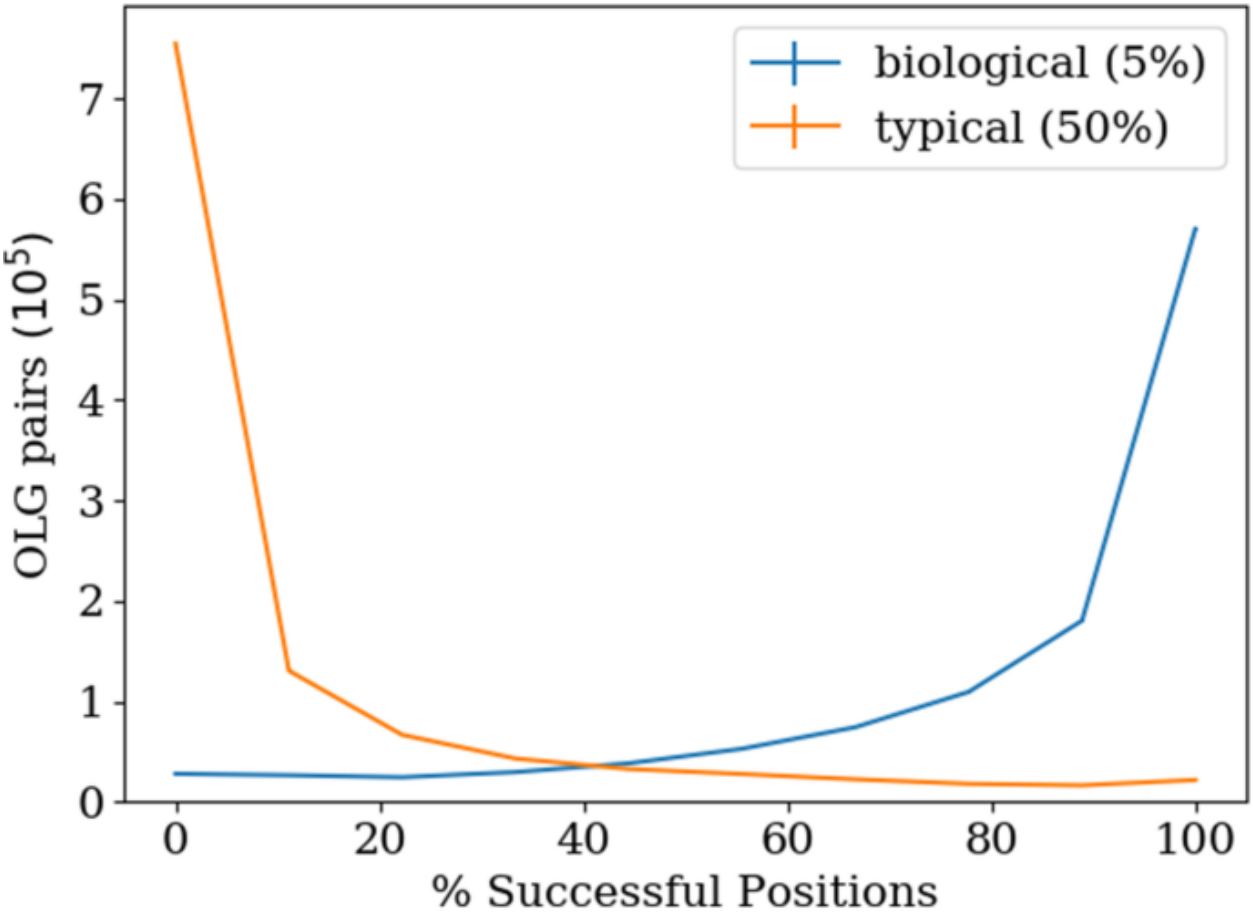

Figure 4.12: Percentage of successful overlap positions in a sequence pair. The data was created from 150 protein domains with a length of at least 70 AAs. Percentages are averaged over 30 sets of 50 random positions. The result strongly depends on the chosen threshold percentile. 'Biologically relevant' OLGs have at least a score at the 5th percentile of naturally occurring homologs, while 'typical' sequences have a score above the 50th percentile. Figure taken from [139].

4.3.4. Success rates for OLG construction in different reading frames

The SGC imposes combinatorial restrictions on the alternative reading frames [105], e.g. alanine in the '+1' frame will always also translate to alanine in the '-2' frame because alanine

always starts with ‘gc’ in the first two nucleotides translating to also ‘gc’ in the first two nucleotides of the ‘-2’ frame. As the SGC is used in the construction algorithm to translate each codon into an AA before calculating its exchange matrix score, these constraints influence OLG construction and should be observable in the success rates across relative reading frames. Plotting success rates as a function of threshold percentile for all reading frames affirms this hypothesis, see Fig.4.13. The success rates of different reading frames align exactly in reverse order of their combinatorial restrictions [105]: the ‘-3’ frame has the highest success rate and lowest number of restrictions, followed by the ‘+2’ and ‘+3’ frame which have the same number of restrictions, followed by the ‘-1’ frame and lastly the ‘-2’ frame with the most combinatorial restrictions and the lowest OLG construction success rate. In the ‘-3’ frame, which is the least restricted reading frame, 14.9% of pairs can be overlapped with the quality of a typical homolog, while in the ‘-2’ frame, the most restricted reading frame, only 3% can reach that level of quality. Interestingly the ‘+2’ and ‘+3’ frames have the exact same success rates in every dataset despite expecting fluctuation due to some sequences fitting better in one or the other reading frame. The average success rate across reading frames is 9.6% for constructed OLGs to score as typical sequences of the protein domain family. The chosen threshold percentile has a strong influence on the success rate as 94.5% of OLGs score at least as highly as the lowest score in the protein family, while only 0.02% score better than 95%. As a reference, the e-value medians of the constructed sequences passing different threshold percentiles are determined using BLAST resulting in an e-value range of 10^{-28} to 10^{-37}, with lower e-values corresponding to higher percentile thresholds. In order to determine the quality of the number of combinatorial restrictions taken from [105] as a predictor for OLG construction success rates this functional dependence is studied in more detail. Success rates are only approximately linearly dependent on the number of restrictions for the lowest possible threshold, which is the lowest score of the ‘full’ database of the respective protein family, see Fig. B.1 of appendix B. While the success rate cannot be predicted directly from the number of restrictions in each reading frame, the SGC nevertheless has a significant influence on the construction of OLGs.

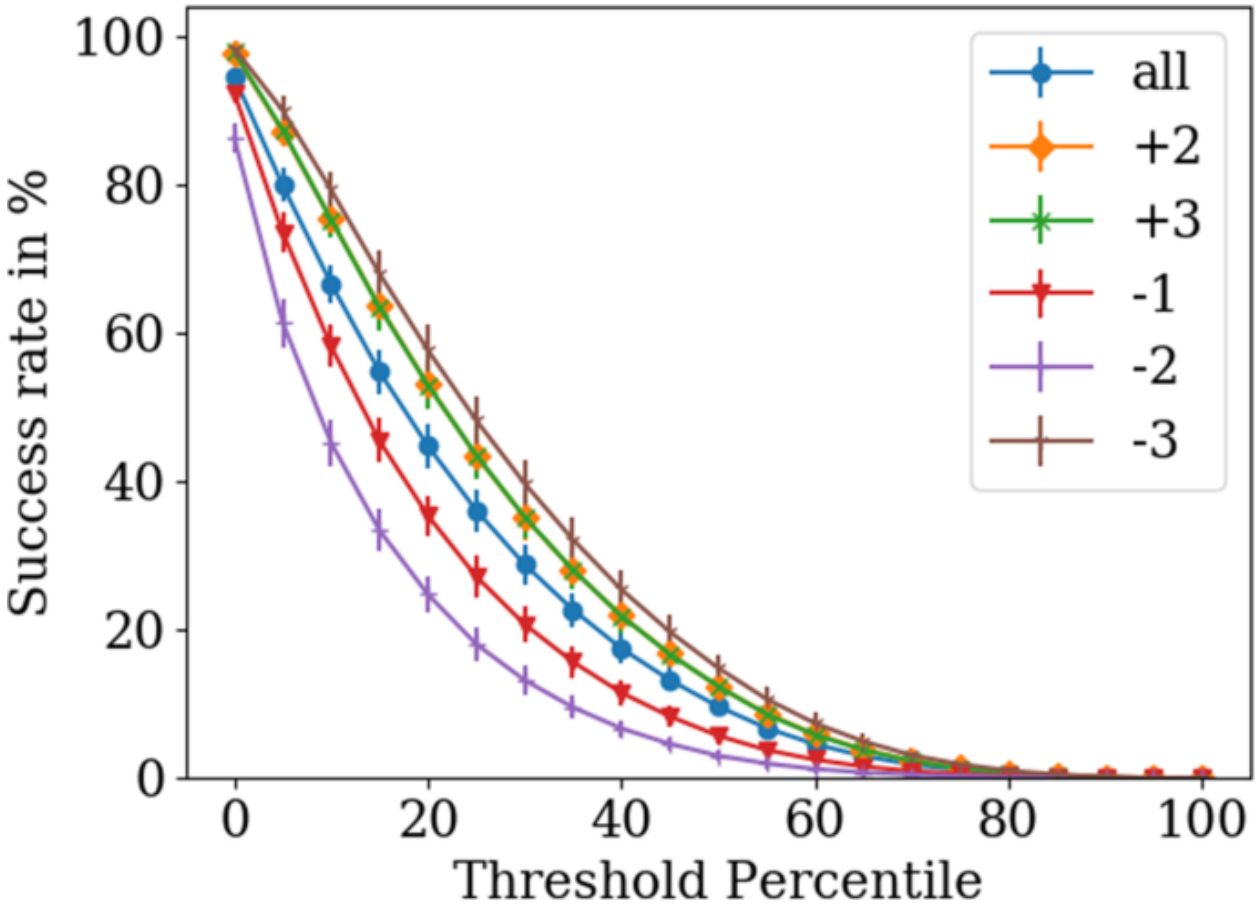

Figure 4.13: Percentage of successfully designed OLGs as a function of the threshold percentile in different reading frames. Each value is an average over 20 datasets with 150 sequences of at least 70 AAs each. The error bars indicate the standard deviation. The success rate of the reading frames are ordered by their combinatorial restrictions. Figure taken from [139].

4.3.5. Independence of different measures of OLG quality

Relative HMM scores, AA identity and similarity as well as secondary structure have been used to determine the quality of the constructed OLG sequences and all evaluations yield the result that constructed OLGs are not distinguishable from naturally occurring homologs. Using different measures only strengthens the meaningfulness of the results if they are independent of each other. While AA identity and AA similarity are obviously very similar properties, which also shows in their strong correlation $r = 0.82$ (Pearson's correlation), all other properties are surprisingly independent of each other as indicated by low correlation values $r < 0.2$ (Pearson's correlation). While sequences with a very high AA identity, such that only a few AA differ from the original sequence, will also have a very high secondary structure similarity, this correlation declines for lower AA identity values and the two parameters become reasonably distinct. All properties showed that the change inflicted on sequences in order to overlap them is in the same range of variation which is seen between natural homologs in Pfam families.

4.3.6.OLG construction in different taxonomic groups

Splitting the input sequences into the basic taxonomic groups, namely archaea, bacteria, eukaryotes and viruses, enables a fifth group to be studied, which are ancient protein domains that can be found in all taxonomic groups. A protein domain family is defined as ancient in this study if it has at least one sequence in every taxonomic group. The older a gene is the more widespread it is presumed to be. While horizontal gene transfer makes it very hard to determine the age of a gene, there is currently no better definition of gene age without detailed evolutionary study of individual gene families. Protein domains that can be found in all basic taxonomic groups are therefore assumed to be already present around the time of LUCA.
In this study viruses are unexpectedly found to be the least suited for creating OLGs, while bacteria and eukaryota are the most suited, see Fig. 4.14. The difference is very significant as OLGs, which have the quality of 'typical' sequences, are more than twice as likely in eukaryotes compared to viruses. The ordering of success rates of different taxonomic groups is stable across different threshold percentiles, see Fig. B.2 of the appendix B. The highest OLG construction success rates can be found for eukaryotes and ancient genes, also labelled 'Found in all'. While ancient genes had more time to explore sequence space and therefore to appear very flexible, the dataset of ancient genes only consists of 50 sequences as no more could be found in the Pfam database. The resulting values are therefore less reliable. Unexpectedly eukaryotes have the highest success rate despite being the youngest taxonomic group among the three cellular organism domains and therefore had less time to explore sequence space resulting in their genes appearing less 'flexible'.

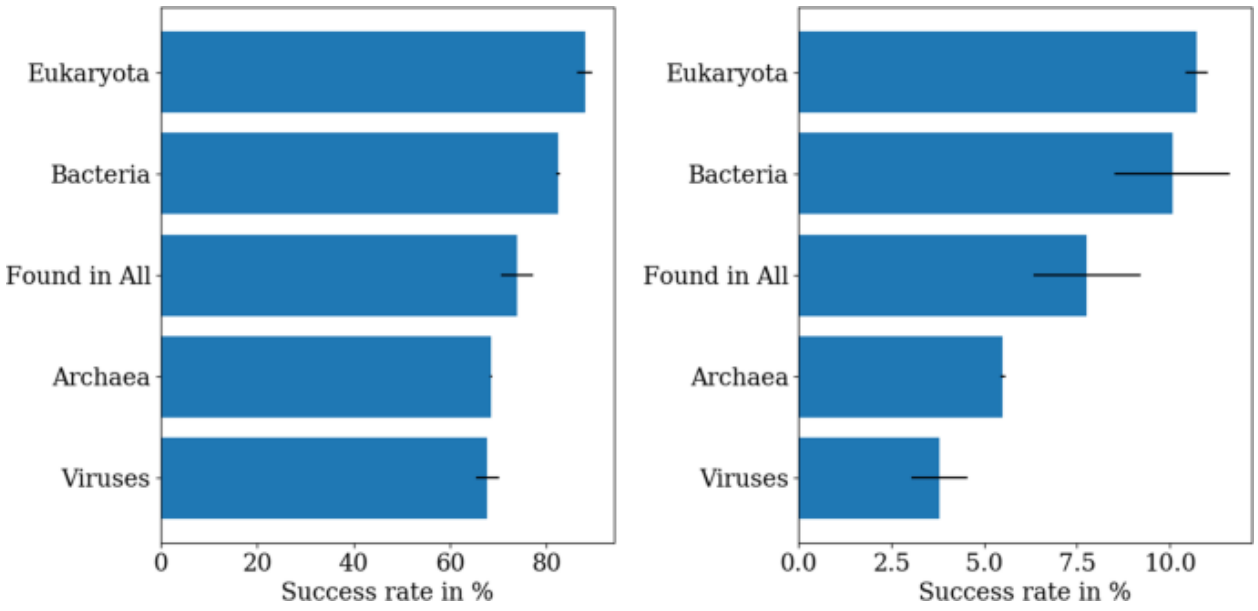

Figure 4.14: Success rates of sequences split into taxonomic groups. Average values are plotted with the standard deviation as error bar. The data was calculated from 20 data sets with 150 sequences of at least 70 AAs length. In both the 'biological relevant' threshold (*left*) and the 'typical sequence' threshold (*right*) the ordering of taxonomic groups is the same and viruses unexpectedly perform the worst. Figure taken from [139].

4.3.7. Evolutionary accessibility of the constructed OLGs

A rough approximation of natural accessibility by random mutation of the constructed genes can be made by determining the difference between the original and the constructed sequences in terms of nucleotide changes. It shows that constructed OLG sequences passing higher percentile threshold are not mutationally closer to their original sequence, see Fig. 4.15. Extreme outliers are removed with increasing threshold percentile, which is probably due to the number of sequences passing higher threshold being much smaller and therefore reducing the expected number of outliers. The distribution of nucleotide change percentages has a mean value of 25% and the range of 20-30% includes half of the designed OLG sequences. Changing 25% of a gene is not accessible by random mutations within a reasonable timescale, but there are outliers with nucleotide difference as low as 1.8% between the constructed and the original sequence, while still passing threshold scores at the 25th percentile. These outliers on the lower end of the distribution with less than 10% nucleotide change could plausibly be realised by chance mutations and are as frequent as 0.6% of sequences, while at least scoring as high as the lowest score in the ‘full’ group. In this dataset 955846 sequences pass this threshold and 5843 of these constructed OLGs have less than 10% nucleotide change to their original sequence.

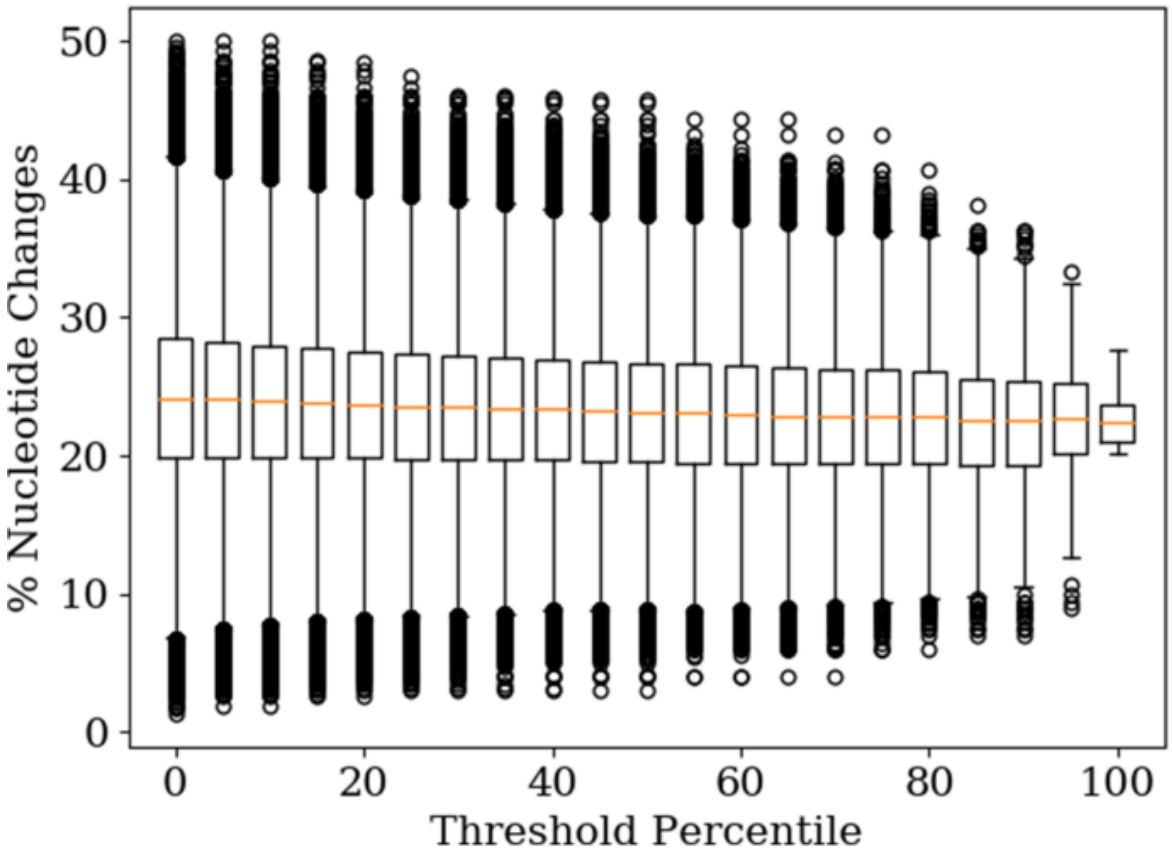

Figure 4.15: Percentage of nucleotide changes needed going from the original to the OLG sequence in different threshold percentiles. The boxplots are calculated from 1,010,00 OLG sequences with at least 70 AA length. Figure taken from [139].

4.3.8. Optimality of the SGC for OLG construction

Since the SGC is part of the OLG construction and its properties clearly impact the results, e.g. differences in reading frames, it is interesting to see the optimality of SGC with regards to OLG construction. Just as in the project covered in chapter 2, optimality is inferred in different code sets, namely the 'Random' code set, the 'Degeneracy' code set and the 'Blocks' code set. The conditional optimality of this property is also tested in a consecutive approach as described in chapter 2.1.2 by using the 'Blocks' code set but only collecting codes that score at least as high as the SGC in the mutational robustness. This code set is called the 'MR_blocks' code set.
The SGC is not optimal in the construction of OLG in any of the code sets, but the code sets are still very different in their results so the influence of different structures can be inferred on this property, see Fig. 4.16. Comparing the average success rates of the 'Random' and the 'Degeneracy' code set, see left panel of Fig. 4.16, it can be deduced that the degeneracy structure of the SGC has a positive influence on OLG construction. The degeneracy of the SGC resulting in its block structure on the other hand negatively impacts the success rate and codes from this code set that express at least the level of mutational robustness as the SGC perform even worse. Consequently, as the codes from the comparison code sets perform worse on OLG construction, the optimality of the SGC increases, see right panel of Fig. 4.16. The more restricted a set of alternative genetic codes is, the more optimal is the SGC. While the SGC is far from optimal, it is imaginable that introducing more restrictions due to other properties could result in the SGC being optimal among the set that takes into account restrictions due to other properties. Studying all reading frames on their own does not change the bigger picture and in no single reading frame does the SGC appear to be optimal, see Fig. B.3-B.6 in appendix B.

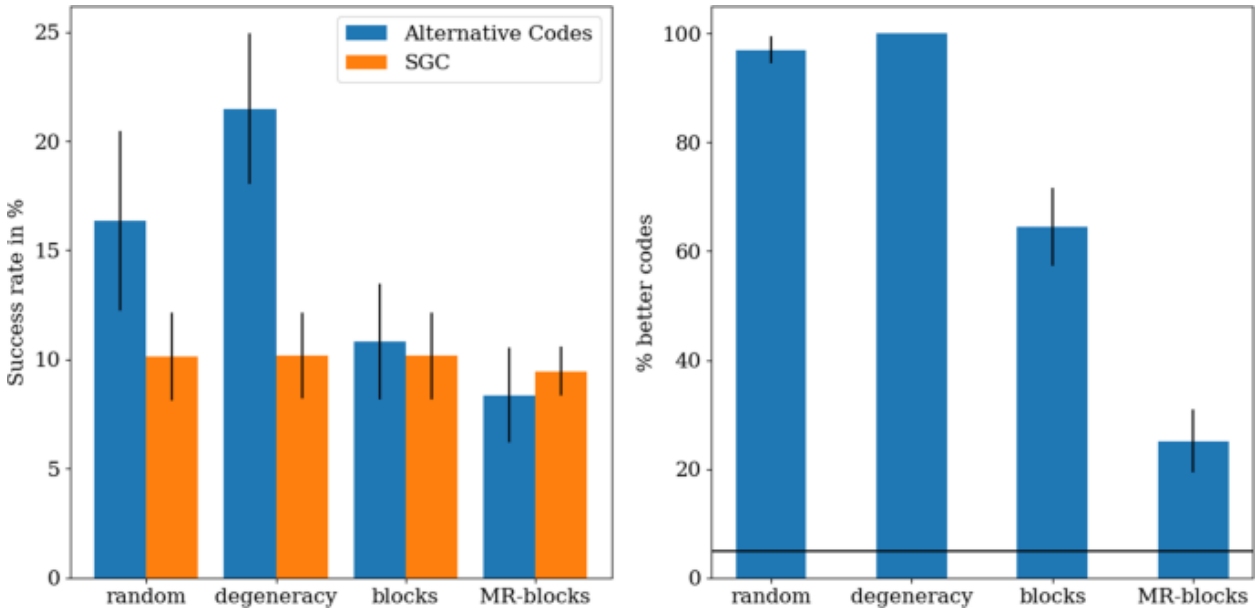

Figure 4.16: Optimality of the SGC in OLG design averaged over all reading frames. The mean and standard deviation (black bars) values are calculated from 10 datasets of 150 sequences with minimum length of 70 AAs and 20 datasets of 100 alternative codes, except for the 'MR-blocks' code set, where 10 sets of 500 codes are used. *Left:* Comparison of average success rates in different code sets. *Right:* The optimality of the SGC in different sets of artificial

codes. The 5% threshold for optimality is indicated by the black line. The SGC has increased optimality for more restricted code sets. Figure taken from [139].

4.4. Discussion

The various results of this study and their implications span a wide range of fundamental problems of biology and bioinformatic research especially on the topic of OLGs. First is how to judge the quality of a gene correctly and what to watch out for especially in artificial sequences. Next is the quality of artificial OLGs, which have been thought to be impossible for a long time. Also the evolvability of naturally occurring OLGs and the factors that make this possible despite the known difficulties of this process. OLGs have been found in all taxonomic groups but are still often neglected outside of viruses, an arbitrary choice which becomes less justified with every piece of evidence in this field. These topics and an outlook on the implications of constructed OLGs on synthetic biology is discussed in the following chapters.

4.4.1. Judging the quality of artificially created genes

Genetic sequences and their respective proteins have many different properties that can be taken into account in order to determine its protein family or infer function. For the OLGs constructed in this study the factors sequence length, conservation profiles of proteins, AA sequence similarity and identity, and sequence similarity as determined by BLAST,HMMs, and secondary structure are used. The results show that absolute e-value cutoffs, even if they are chosen to be excessively conservative, cause artefacts due to sequence length dependencies. In the case of constructed OLGs, such an evaluation completely determines the quality assessment of the sequences. Judging sequences relative to known homologs resolves this problem, which can best be done using HMMs instead of BLAST as an average has to be defined to which relative distances are calculated, which can be found using HMM profiles. Here both the HMM profiles as well as the homologs in each protein domain family are taken from the Pfam database, which is therefore an integral element of the quality of this evaluation. It is not clear how to judge the reliability of this database, but it is imaginable that artefacts can easily arise, e.g. if most sequences in a protein domain family originate from the same species or genus they will be very similar and the family will appear to be highly conserved and ‘rigid’, resulting in higher thresholds for constructed OLG sequences to pass. Also technical details, of which the state of the art is constantly being improved, like sequence selection for the ‘seed’ and ‘full’ database and alignment creation for the HMM profiles, determine the quality of this database and its suitability for creating relative thresholds.
Going beyond the properties used here, tertiary protein structure or intra-protein interactions could be used as has been done in a recent study of synthetic OLGs [138], as HMMs do not include the many important long range AA correlations between sites distant in the primary sequence which are close in the folded tertiary structure. Also hydrophobicity profiles, as used to compare frame-shifted sequences to their original in [40], or residue-residue co-evolution, which has been used in [152] to create artificial sequences, could be used. It is important to go to such lengths as the appropriate bar for functional sequences should be much higher for designed genes than for naturally occurring sequences. The latter have been maintained by natural selection, i.e. have survived a filter which tends to remove non-functional sequences,

and it is therefore sensible to assume they have a function yielding a fitness advantage for the organism and the question shifts to which function it is; while designed sequences must be shown to have any kind of function first. Also it is important to design and evaluate sequences using the same standards. BLAST for example does not use conservation weights in its scoring, which leads to a negative impact of the same weights being used in the construction process. An evaluation with HMMs on the other hand takes into account the conservation profile of a protein family and designing sequences using these weights has a positive effect, which was able to be optimised in this study by varying the influence strength of the conservation weights. The same is expected to happen when different AA exchange matrices are used in the construction and evaluation process. While AA similarity/identity and secondary structure are also determined in this study, most results are obtained using HMM scores.

4.4.2. Constructed OLGs are on the level of natural homologs

The similarity of artificial OLG sequences to naturally occurring homologs has been shown in three independent properties, namely the HMM score, the AA identity/similarity and secondary structure. 94.5% of the artificial sequences had a higher HMM score than the worst scoring homolog and 9.6% even scored as high as 50% of the homologs. The typical secondary structure deviations among homologs is at approximately the same level as the typical deviations caused by altering sequences to overlap with any other sequence. And lastly the AA identity (similarity) with an average value of 60% (75%) of constructed sequences to their respective original sequence, is far above the threshold of 34% AA identity, above which homologs have the same protein structure. Consequently, the necessary average change inflicted on sequences to create OLGs is in the typical range of variation between homologs of the same protein family. While this is an important result contradicting the common belief that formation of OLGs is impossible due to information content limitations of nucleotide sequences [115] and other factors, the analysis done here is only a first step to verify the accessibility of OLGs in sequence evolution. More sophisticated and biologically relevant but computationally more costly properties like tertiary protein structure or intra-protein interactions are an important next step in order to predict the functionality of the artificial sequences. While the latter has already been successfully applied to constructed OLG sequences [138], tertiary structure is more complex, as many more factors besides the AA sequence influence the result, e.g. codon usage [153] or inherently variable factors like the presence of chaperone proteins. Until bioinformatic prediction of sequence functionality becomes much more sophisticated and reliable, only experiments can yield certainty. As experiments are much more elaborate than bioinformatics study, it is very useful to create a gold standard for selecting sequences for experiment. The more properties are included the better the results but in order to reduce computational time sequence candidates should be filtered with simpler properties first. This study has shown that in particular the use of relative HMM scores is a useful property to pre filter sequences before analysing secondary structure or even more computationally intensive properties.

4.4.3. The case of naturally evolved OLGs

Here functional protein domains are directly overlapped with each other as a ‘worst case scenario’ for OLGs. Overlapping a functional domain with a less functionally important and therefore less conserved region yields more flexibility for creating an OLG. Taking into account that some OLGs only vary by 1.8% from their original sequence it becomes much more conceivable how existing OLGs evolved. The difficulty in evolving OLGs ties directly to the number of OLGs expected in a genome and thus makes *de novo* gene birth by overprinting an even more sensible hypothesis.

Shortly after the discovery of the first OLGs, the rate of evolution in OLGs was estimated [154] but no further research has been done on the inherent evolvability of OLGs. The results acquired here could be a starting point to reopen the field of OLG evolution. As in any genetic evolution, the SGC plays a central role and even more so for OLGs. It explains reading frame differences in OLG construction and creates the high AA identity/similarity between constructed and original sequence. While the SGC does not appear to be optimised in this study, the percentage of codes more suited for OLG construction steadily decreased with the introduction of additional evolutionarily sensible restrictions on the genetic codes used for comparison. The most restricted code set consists of codes with the same degeneracy in the third nucleotide position and all codes must have at least the mutational robustness of the SGC. The composition and arrangement of the SGC creates a strong optimality in the mutational robustness, which is the most straightforward property the genetic code should have. Especially the block structure of the SGC has been shown in this study to be highly detrimental for OLG construction and it is sensible to assume that the fitness advantage due to the mutational robustness outweighs the fitness advantage of OLG construction. Considering that the mutational robustness could be optimised much further, one explanation for its current extent in the SGC could be that an even further increase in this property deteriorates other properties such that the total fitness is no longer increased, indicating a turning point. Nevertheless it is astonishing that overlapping random Pfam protein domains can be achieved while inflicting so little change on the original sequences, despite the strong optimality for mutational robustness in the SGC.

4.4.4. Where to expect OLGs to exist

It has long been assumed that OLGs only exist in viruses, which has long been debunked, but their function is still strongly doubted or debated outside of viruses. The previously reported result that virus genes are more suited for constructing OLGs [134], which fit with common assumptions regarding the taxonomic distribution of OLGs, can be entirely explained by dataset-database biases and the improved evaluation even showed eukaryotic and bacterial genes are much more suited than virus genes to create OLGs. Trying to explain this intriguing result, the main differences between the taxonomic groups are the expected mutation rates and the average length of a protein. The sequence length has been successfully removed from the evaluation as a factor while the mutation rate is known to be much higher in viruses compared to other taxonomic groups. A higher mutation rate translates to more homologs being explored within a given timeframe, resulting in a protein domain family appearing more ‘flexible’. The main consequence of this is lower threshold values as the homologs differ more from the HMM

profile. Lower threshold values would result in higher success rates - but the opposite is observed as viruses, having the highest mutation rates, have the lowest success rates. Another factor influencing the ‘flexibility’ of sequences in protein domain families is the age of a protein domain. Older sequences had more time to explore sequence space resulting in the same effect as high mutation rates. This explanation can be partly confirmed by old proteins having high success rates, but there are only very few protein domains in the Pfam database that can be found in all taxonomic groups and therefore be labelled as old, so the results here are not very reliable. Also eukaryotes, which have the highest success rate, are assumed to be younger than prokaryotes again contradicting this explanation of apparent domain flexibility. While distinguishing real sequence flexibility from mutation saturation of explored sequence space is very difficult since sequence space is so large, it is very important to do so in order to make sure the results are not artefacts due to the biased nature of currently existing knowledge reflected in the existing protein databases.
The differences in success rates for OLG construction found in this study could also be due to biases in the exchange matrix used for OLG construction and evaluation in this study, namely the BLOSUM62. This matrix is constructed from AA exchange rates found in alignments of homologs. Depending on how these alignments are constructed and which proteins are chosen to contribute to the total exchange rates, biases towards certain kinds of proteins and therefore rates of AA exchanges could be incorporated. Repeating this study for different existing exchange matrices or even with matrices created from sequences of specific taxonomic groups would clarify whether these biases exist and contribute to the observed success rates. As stated before it is important to use the same exchange matrix in both the construction and evaluation in order to have sensible data.
A convincing explanation for differences in the ability to create OLGs between taxonomic groups is still missing. Approaching this problem from a more reliable point of view, it would be clarifying to see in which group more OLGs can be found. This would however require a reliable OLG detection tool, and methods for this are still in development. Factors that have a positive impact on successful OLG creation could aid in predicting OLGs in sequenced genomes. One such factor could be the flexibility of known sequences or parts of sequences. OLGs are, all else remaining equal, more likely to be encoded parallel to flexible sequence sections.

4.4.5. Outlook for OLGs in synthetic biology

Artificial OLGs are a very interesting topic for synthetic biology. Since mutations in overlapping sequences are more deleterious, OLGs are more resistant to mutation on a population level as mutated sequences are more likely to be selected against. While this may be an evolutionary risk in a competitive environment, it is a technical advantage in a controlled environment as the organism carrying an OLG construct is less likely to change and lose the function it has been originally designed for. Such a stabilisation has been achieved using OLGs in two ways. First, by overlapping a gene between an essential gene and its ribosome binding site so it is protected against ‘polar’ mutations such as frameshifts [155]. The second study uses the OLG construction algorithm of [134] and shows that OLGs indeed have a higher percentage of deleterious mutations compared to non-OLGs [138]. But the constructed OLGs did not fully recover the growth rate of the deletion mutant to that of the wild type, so the function of the

constructed sequences is partly impaired [138]. Further improving the construction and evaluation of OLGs should produce sequences with higher functionality.

Another application for OLGs in synthetic biology is biomolecular computing, which tries to mimic viruses by designing and inserting genetic programs into cells to control them for other purposes [156]. A limiting factor for more complex programs is the genome size of such programs, which can be reduced much more drastically using OLGs compared to existing approaches [157]. Furthermore, OLGs would stabilise the genetic material of such programs against random mutations so that programs shut down instead of going out of control as a fail-safe mechanism.

While only overlaps of two genes are studied here, overlaps of more than two sequences have been attempted in [134], which is very ambitious but does not seem impossible from their results and would push the genome stabilisation and compression even further.

5. OLG Construction for Experiments

Independently of the work in a study published in *Science* which we discovered later [138], an attempt at experimentally verifying the functionality of constructed OLGs was made in collaboration with the master student Alexandra Woller as a part of her master thesis supervised by Dr. Klaus Neuhaus. The general experimental setup is to overlap two reporter genes that can easily be tested for functionality, chosen from a set of antibiotic resistance genes, auxotrophic complementation genes and fluorescence genes. The latter group did not show promising candidates in the constructed OLGs so they have not been included in the experiments. While the function of the constructed OLGs could not be proven, the experiments are not conclusive yet. Here the focus is on the bioinformatic preliminary work for sequence selection, HMM profile construction, conservation weight determination, OLG construction, OLG evaluation and RBS insertion, which have all been done by me. Reporter gene family selection, manual alignment curation, final OLG selection and experimental laboratory work have been conducted by Alexandra Woller. While some parts conducted by her are described here, more details especially on the laboratory work can be found in her master thesis [158].

5.1. Bioinformatic methods

In the previous project, the Pfam database provided all prerequisites needed for OLG construction and evaluation. The protein sequences of the reporter genes were downloaded by Alexandra Woller from the UniProt database but these sequences vary substantially and no sensible alignment can be constructed right away, nor would all sequences be considered homologs. The preliminary work described in the following chapters is needed to prepare sensible input data for OLG construction.

In the project described in the previous chapter protein domains were used, which did not include start and stop codons needed for full genes. Consequently, the OLG construction algorithm has to be extended to include start and stop codons. The organism used for the experiments is *E.coli* MG1655 and some codons are very rare in its genome. In order for the constructed OLGs to be translated properly, codon usage statistics are also incorporated into the OLG construction algorithm.

Despite the condon weights, rare codons can accumulate hindering an optimal translation. Such OLG pairs and those that are below a minimum relative HMM score are removed from the list of candidates for the experiment. Since secondary structure prediction is computationally intensive, it is only predicted for the OLG pairs with the highest AA similarity. Sorted by secondary structure prediction, the OLGs are manually inspected as a last step.

In order to be able to use the OLG sequences for experiment, RBSs are added and sequences including cutting sites for enzymes used in the experiments discarded.

5.1.1. Preliminary work

For each protein group a curated alignment is created in order to determine the conservation of each position in the sequence for OLG design and to create reliable HMM profiles for OLG

evaluation. Sequences of all protein groups are scored against their respective HMM profile to determine the sequence best representing each group by the highest score.

5.1.1.1. Curated alignment creation

The protein groups must be curated before creating the alignments. Identical sequences are removed since the alignments are only meant to represent the functional space of each protein. In different species the same protein can be very different since the same function is realised in a different way or the protein has different interaction partners. In order to get only the sequences of one 'realisation' of each protein, the sequences are clustered with mmseqs (using the flags --min-seq-id 0.5 -c 1.0 --cov-mode 0) [159] and the biggest cluster with sequences of a minimum sequence length of 100 AA is used. In order to reduce the number of gaps in the alignments, which are often hard to handle by alignment programs, sequences of the most frequent sequence length of each cluster are selected plus those which deviate by up to 1% from this length. The remaining protein sequences contained only sequences from bacteria and mostly only of the same genus. These sequences are aligned using QUICKPROBS 2.06 [160] and manually checked for outliers by Alexandra Woller. If outliers are found they are removed from the alignment which is then realigned.

5.1.1.2. Conservation weight calculation

Sequence weights can prioritise the AA of one sequence over the other to increase the chance of obtaining functional sequences. Weights are calculated just as in the previous project, namely the weight p_i at position i of the sequence is calculated as $p_i = e^{-ks_i}$, where s_i is the entropy calculated at position i in the alignment and k is the weight strength determining the impact of the weights. Here a value of $k = 0.5$ is used as optimised in the previous study. The entropy is calculated by defining the 6 AA groups as in [134], namely {LVIMC}, {FYW}, {G}, {ASTP}, {EDNQ}, and {KRH}.

5.1.1.3. HMM profile construction and scoring

The HMM profiles are constructed with HMMER3 (v3.2.1) [144] from the aligned sequences using 'hmmbuild' without any flags. Sequences are scored against these profiles using 'hmmsearch' with the flags '-T 0 --max'. The resulting score is divided by the length of the input sequence in order to be able to compare sequences of different lengths.

5.1.2. OLG construction

The selected sequences are overlapped with each other except those of the same reporter gene type, e.g. one must be an antibiotic resistance gene and one an auxotrophy compensation gene. OLG pairs are created in every reading frame and at every possible position such that sequences are fully overlapping, which means that at most one nucleotide of the shorter sequence is not overlapping with the other sequence. Here the longer gene is labelled as the MG and is always placed in the '+1' frame. The shorter gene is consequently the OLG. Each gene must start with a start codon used in *E.coli*, namely ATN and NTG with N being any nucleotide. Stop codons are added with a variable length tail of up to five uncharged AAs in

order to find the best position for the stop codons. Besides the conservation weights also codon weights influence the OLG construction. The influence of codon weights is such that rare codons are similarly rare as in original *E.coli* genes.

5.1.2.1. Codon weight calculation

Codon usage statistics are determined from the *E.coli* O157:H7 str. EDL933 genome assembly ASM73296v1. The codon usage is the percentage of each codon used in the annotated genes. The codon weights w_i are the codon usage c_i to the power of the codon weight factor, which is 0.5, see eq. (25).

$$w_i = c_i^{0.5} = \sqrt{c_i} \tag{25}$$

5.1.2.2. Adapting the OLG construction algorithm

OLGs are constructed using the algorithm from [134] described in chapter 4.2.6 but expanded by using codon weights. The score of each position in the overlap originally calculated using eq. (24) will be calculated according to eq. (26) instead. The new equation is the score of the previous one but multiplied with the sum of the codon weights w_i and v_i, so codons with a higher usage percentage are more likely to be used in the OLGs.

$$S_i(X', Y', X, Y) = \left(p_i E(X_i, X'_i) + q_i E(Y_i, Y'_i)\right) \cdot \left(w_i + v_i\right) \tag{26}$$

In order to find a good position for the stop codon a tail of up to five uncharged amino acids is added at the end of both genes. Since these amino acids are outside of one of the two genes the codon and conservation weights are zero for this gene in (26). Of all possible tails, the one with the highest sum of scores is used. In order to be able to compare tails of different size and to disfavour long tails, which increase the overlap size, the AAs of the following gene which are not overlapped with the tail have the maximal possible score, which is realised by not changing the AA. For AAs outside of the overlapping region codons are chosen randomly, but according to codon usage percentages.

5.1.2. OLG evaluation

The relative HMM score, the sequence similarity and the secondary structure similarity are used as markers in order to select sequences for experiments. Furthermore OLG pairs in which any sequence has at least three consecutive rare codons of class I [161] in *E.coli* are discarded. Besides a small usage rate in genes, rare codons are also defined by a low abundance of tRNA decoding them. An excessive amount of rare codons most likely hinders translation of a gene and is therefore avoided. Next, the sequence similarity of the OLG sequences to their respective original sequence is determined and the OLG pairs are ordered by decreasing AA similarity. Up to 10 of the highest scoring sequences of each reporter gene pairing are chosen for the secondary structure evaluation. Ordered by their secondary structure identity, OLG pairs can be manually inspected and selected.

5.1.2.1. Relative HMM score

The relative HMM score is the quality $Q = 100 \cdot \frac{S}{S_{max}}$ just as defined in chapter 4.2.4 by the score of the OLG sequence S and the score of the original sequence S_{max}. Since all homologs have already been used in the HMM profile construction, their scores against the same profile are not used as a comparison for the constructed OLG sequence in order to avoid possible artefacts. Only OLG pairs in which both sequences have a relative score of at least 0.6 are considered for the next steps.

5.1.2.2. Amino acid similarity

Two AAs are defined as similar if their exchange matrix entry is larger than zero. Here the BLOSUM62 exchange matrix is used. The similarity used to order OLG pairs is the lower percentage of the two OLG sequences. The lower similarity of the two sequences of each OLG pair is used for the evaluation.

5.1.2.3. Secondary structure identity

Secondary structure of both original sequences and constructed OLG sequences is predicted using Porter 5 [148] with the "--fast" flag, which distinguishes between the eight different secondary structure motifs of the dictionary of protein secondary structure (DPSS) [149], [150], [151], namely 3_10-, alpha-, and phi- helices, hydrogen bonded turns, beta sheets, beta bridges, bends and coils. The percentage of AAs of the constructed genes which form the same secondary structure as their respective original defines the secondary structure identity. The lower of the identities of the two constructed OLG sequences of each pair to their original reference sequences is used as the pair's representative identity.

5.1.3. OLG refinement

OLGs which lie in the middle of the mother MG need a RBS in order to be translated. Using the RBS motives from [162], the highest order motive that can be added without changing the AA sequence of the MG is inserted into the sequence, while removing any premature start codons starting in any reading frame of the region three base pairs downstream of the RBS until the real start codon of the OLG without changing the MG.

5.2. Experimental setup and Methods

The experimental setup has been designed by Alexandra Woller and is briefly summarised here. The constructed OLG sequences of two reporter genes are introduced into *E.coli* separately using plasmids. The focus of this study is to determine whether the change inflicted on genes by OLG construction impairs their function, so testing each OLG sequence on their own is the simpler experiment and creates clearer results. If the individual constructed sequences are functional the OLG pair can easily be tested simultaneously subsequently. Since overlaps with fluorescence genes were not very successful, all plasmids contained one antibiotic resistance gene and one auxotrophy compensation gene. Only antibiotic resistance due to enzymatic antibiotic inactivation are used; more precisely, antibiotic inactivation is due to hydrolysis or

modification by the selected enzymes. In these cases, the source of antibiotic resistance can be traced to an enzyme encoded in a single gene and is therefore suited for OLG construction. The auxotrophic knockout mutants of *E.coli* have lost the ability to generate all essential organic compounds and need a special medium providing a compound in order to grow. Reintroducing the knocked out gene in intact form in a plasmid removes the auxotrophy and the bacteria can grow normally. *E.coli* is used in this study as its genome is well known [163] and many auxotrophic mutants are available [164]. Due to an unstable phenotype, vitamin auxotrophs are excluded and only AA auxotrophs are used. After plating the bacteria including the plasmids on selection plates containing the respective antibiotic or missing the substrate of the chosen auxotrophy, OLG functionality can be detected by measuring the growth of the bacteria.

5.3. Results

First the results of OLG construction, evaluation and refinement are discussed using the selected genes for experiment as a representative sample. Experimental results are briefly discussed afterwards.

5.3.1. OLG construction, evaluation and refinement

In the constructed sequences no longer tail length than three uncharged AAs was necessary, with tail lengths greater than zero being very rare. The increase in the OLG size due to the included tail is successfully disfavored by the adapted construction algorithm. Two sets of 10 OLGs sequences were selected for the growth experiment. Their evaluation values, percentage of rare codons, reading frame and RBS are shown in the tables 5.1 and 5.2. Rare codons have a prevalence of 2.98% in the annotated genes of *E.coli* but are usually 2-3 times more frequent in the OLGs. Increasing the codon weight strength, the usage of rare codons can be reduced in the OLGs but at the cost of OLG quality in other measures. With values barely above 60% in the best sequences, the quality Q is already much lower than in the previous project, which did not include codon weights and had average values of 80%, see Fig. 4.4. But this could also be due to more rigid HMM profiles in the current project since the alignment curation is very strict. The AA similarity on the other hand has very similar maximal values, which are around 80% in both projects, see Fig. 4.8. Also the secondary structure identity values match the previous project with the best values slightly above 90%, see Fig. 4.11. The construction process used here is more realistic and controlled in order to get sequences that can be used by an organism in comparison to the purely theoretical approach using the Pfam database, but the sequences still have a very high quality. Since genes with three or more rare codons in a row are removed the higher usage of rare codons is assumed acceptable. The OLGs with the highest quality are mostly in the ‘-3’ frame, which is expected as it is the most flexible. RBS sites could be included into most OLGs without changing the MG, but the motifs are often low quality, with S27 being the highest quality and S1 being the lowest. If no RBS motif could be included in an OLG sequence while removing all premature start codons, the OLG pair is discarded from the list of potential sequences for experiment.

Table 5.1: Bioinformatic quality of the first set of 10 constructed OLGs used in the growth experiment. The relative HMM score measured by the quality Q, the AA similarity, the secondary structure identity, percentage of rare codons and RBS motif are listed as OLG quality markers.

OLG	Quality Q	AA similarity	Secondary structure	Rare codons	Reading frame	RBS
#0	61.73%	80.70%	91.84%	8.88%	-3	S2
#1	64.04%	83.45%	90.68%	5.62%	-3	S23
#2	61.49%	83.44%	90.34%	7.14%	-3	S1
#3	65.64%	80.87%	90.32%	5.24%	-3	S13
#4	61.54%	83.44%	90.13%	7.59%	-3	S13
#5	62.15%	82.86%	89.80%	8.15%	-3	S13
#6	61.25%	77.13%	89.80%	3.57%	-3	S13
#7	63.89%	78.30%	89.61%	3.68%	-3	S23
#8	64.27%	82.75%	89.47%	6.21%	-3	S13
#9	61.09%	80.68%	88.82%	5.52%	+3	S13

Table 5.2: Bioinformatic quality of the second set of 10 constructed OLGs used in the growth experiment. The relative HMM score measured by the quality Q, the AA similarity, the secondary structure identity, percentage of rare codons and RBS motif are listed as OLG quality markers.

OLG	Quality Q	AA similarity	Secondary structure	Rare codons	Reading frame	RBS
#11	61.10%	82.75%	88.82%	10.71%	-3	S22
#12	62.42%	80.00%	88.57%	7.41%	-3	S13
#13	60.34%	82.13%	88.57%	4.29%	-3	S16
#14	60.37%	79.16%	88.34%	5.09%	-3	S13
#15	64.92%	80.72%	88.19%	6.55%	-3	S24
#16	61.24%	81.37%	88.16%	6.21%	-3	S13
#17	62.77%	80.87%	89.21%	5.97%	-3	S24
#18	60.43%	77.86%	89.12%	9.63%	-3	S13
#19	61.09%	80.68%	88.82%	5.52%	+3	S13

#20	63.88%	81.93%	88.06%	6.94%	-3	S21

5.3.2. OLG selection

Alexandra Woller manually compared the constructed OLG sequences with 10 random sequences from their respective alignment and their original sequence, see Fig. 5.1. The alignment was scanned for changes in hydrophobicity, charge and size in the AAs. Changes from a hydrophilic to a hydrophobic AA, uncharged to charged AAs and small to large AAs are especially problematic. Special attention is brought to the AAs glycine and proline. The former is the smallest protein and changes to glycine are always problematic. Inserting proline breaks helices and sheets by creating turns and is therefore also particularly harmful. Similar to the previous project at most 40-50% of AAs are changed in the constructed sequence, see Fig. 4.8. Problematic changes were found in 15-20% of all AA changes [158], resulting in 6-10% of positions in the constructed sequences being changed considerably. Since this is similar for all OLG sequences, this was not considered in the selection process.

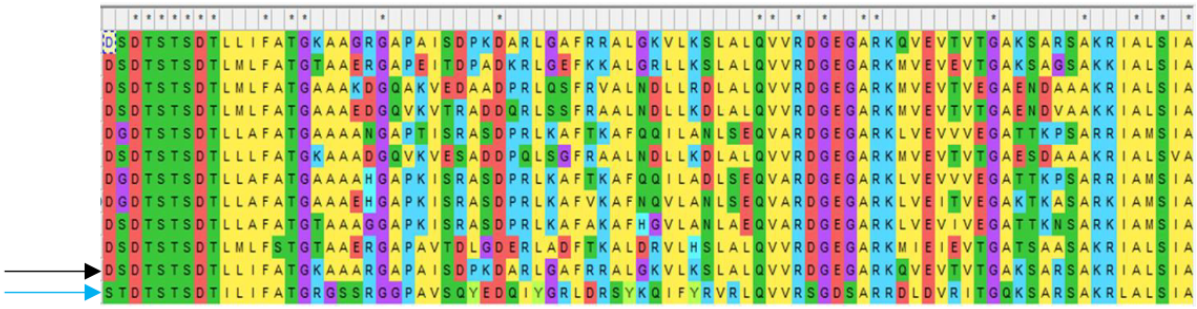

Figure 5.1: Part of the AA Sequence alignment of the N-acetylglutamate synthase including an OLG sequence (blue arrow). The black arrow indicates the sequences best representing the HMM profile of this alignment used for OLG construction. The colour scheme groups AA with similar properties into groups, e.g. AAs with a red background are negatively charged. Figure taken from [158].

The OLG sequences chosen for experiment are listed in tables 5.3 and 5.4 including the auxotrophic bacterial strain. The MG of chosen OLGs as well as their respective original sequences used for OLG construction are listed in appendix C. As a final step the selected sequences have been checked for restriction enzyme cut sites in order to select enzymes for experiments. For the two enzymes EcoRI and HindIII no cut sites are found, so they can be used for molecular cloning.

Table 5.3: First set of 10 OLG pairs selected for the growth experiments. Table taken from [158].

OLG	pBSK (mother) gene	pBKS (embedded) gene	Antibiotic resistance	Bacterial strain	Auxotrophy
#0	N-Acetylglutamatesynthase	Fosfomycin thioltransferase	Fosfomycin	JW2786	Arginine
#1	Fusaric acid resistance	Pyrroline-5-carboxylate reductase	Fusaric acid	JW0377	Proline
#2	Argininosuccinate lyase	Aminoglycoside acetyltransferase	Aminoglycosides	JW3932	Arginine
#3	Aminoglycoside nucleotidyltransferase	Pyrroline-5-carboxylate reductase	Streptomycin	JW0377	Proline
#4	Argininosuccinate lyase	Aminoglycoside acetyltransferase	Aminoglycosides	JW3932	Arginine
#5	N-Acetylglutamatesynthase	Fosfomycin thioltransferase	Fosfomycin	JW2786	Arginine
#6	Homoserine-O-succinyltransferase	Fosfomycin thioltransferase	Fosfomycin	JW3973	Methionine
#7	Aminoglycoside nucleotidyltransferase	Pyrroline-5-carboxylate reductase	Streptomycin	JW0377	Proline
#8	Homoserine-O-succinyltransferase	Aminoglycoside acetyltransferase	Aminoglycosides	JW3973	Methionine
#9	3-Isopropylmalate dehydrogenase	Aminoglycoside acetyltransferase	Aminoglycosides	JW5807	Leucine

Table 5.4: Second set of 10 OLG pairs selected for the growth experiments. Table taken from [158].

OLG	pBSK (mother) gene	pBKS (embedded) gene	Antibiotic resistance	Bacterial strain	Auxotrophy
#11	O-Succinylhomoserine lyase	Aminoglycoside acetyltransferase	Aminoglycosides	JW3910	Methionine
#12	Diaminopimelate decarboxylase	Fosfomycin thioltransferase	Fosfomycin	JW2806	Lysine
#13	Diaminopimelate decarboxylase	Fosfomycin thioltransferase	Fosfomycin	JW2806	Lysine
#14	Homoserine-O-succinyltransferase	Chloramphenicol acetyltransferase	Chloramphenicol	JW3973	Methionine
#15	Fusaricacid resistance	Diaminopimelate decarboxylase	Fusaric acid	JW2806	Lysine
#16	Serine hydroxymethyltransferase	Aminoglycoside acetyltransferase	Aminoglycosides	JW2535	Glycine
#17	Aminoglycoside nucleotidyltransferase	Pyrroline-5-carboxylate reductase	Streptomycin	JW0377	Prolin
#18	Threonin synthase	Fosfomycin thioltransferase	Fosfomycin	JW0003	Threonine
#19	3-Isopropylmalate dehydrogenase	Aminoglycoside acetyltransferase	Aminoglycosides	JW5807	Leucine
#20	Fusaricacid resistance	3-Isopropylmalate dehydrogenase	Fusaric acid	JW5807	Leucine

5.3.3. Experimental results

Since OLG sequences are tested individually, the experiment consists of 40 genes being tested for functionality. The results of the different experimental steps are summarised in table 5.5 and 5.6. The methodical details can be found in [158]. All sequences were successfully inserted into plasmids (restriction digest, ligation and transformation) and all but four plasmids could be successfully inserted into their respective bacterial strain (blue-white screening). Four strains did not show any growth in the blue-white screening. A colony PCR determined the correct insertion size in 30 of the 36 remaining strains but only 7 showed growth on the selection plates. A negative control of bacteria carrying an empty or no plasmid at all was done for all 30 strains with the correct plasmid size in the colony PCR on the selection plates. Growth was detected in 9 strains while none should be growing. This is especially alarming as 3 of these 9 strains were

auxotrophic and all autotrophs have been tested before, which led to the exclusion of vitamin auxotrophs, showing experimental inconsistencies. Visual investigation of the agar plates indicated a possible contamination with a different organism [158], but this has not been studied further. Unfortunately all 7 strains containing one of the designed sequences that showed growth on the selection plates also showed growth in the negative control, so no OLG could be proven to have maintained its function in the construction process. Due to time restraints for the master project, experiments had to be stopped at this point.

Table 5.5: Summary of the experimental results of the first 10 OLGs. Table taken from [158].

OLG #	Plasmid	Characteristic	Restriction digest	Ligation	Transformation	Blue-white screening	Colony PCR - correct	Growth on selection plate	Growth of negative control	Repeat
0	pBSK	Auxotrophy comp.	Yes	Yes	Yes	White	Yes	No	Yes	No
	pBKS	AB resistance	Yes	Yes	Yes	White	Yes	Yes	Yes	No
1	pBSK	AB resistance	Yes	Yes	Yes	White	No	X	X	Yes
	pBKS	Auxotrophy comp.	Yes	Yes	Yes	White	No	X	X	Yes
2	pBSK	Auxotrophy comp.	Yes	Yes	Yes	White	Yes	No	No	No
	pBKS	AB resistance	Yes	Yes	Yes	White	Yes	No	No	No
3	pBSK	AB resistance	Yes	Yes	Yes	White	Yes	No	No	No
	pBKS	Auxotrophy comp.	Yes	Yes	Yes	White	Yes	No	No	No
4	pBSK	Auxotrophy comp.	Yes	Yes	Yes	White	Yes	No	No	No
	pBKS	AB resistance	Yes	Yes	Yes	White	Yes	No	No	No
5	pBSK	Auxotrophy comp.	Yes	Yes	Yes	White	Yes	No	Yes	No
	pBKS	AB resistance	Yes	Yes	Yes	White	Yes	Yes	Yes	No
6	pBSK	Auxotrophy comp.	Yes	Yes	Yes	White	Yes	No	No	No
	pBKS	AB resistance	Yes	Yes	Yes	White	Yes	Yes	Yes	No
7	pBSK	AB resistance	Yes	Yes	Yes	White	Yes	No	No	No
	pBKS	Auxotrophy comp.	Yes	Yes	Yes	White	Yes	No	No	Yes
8	pBSK	Auxotrophy comp.	Yes	Yes	Yes	White	Yes	No	No	No
	pBKS	AB resistance	Yes	Yes	Yes	White	Yes	No	No	No
9	pBSK	Auxotrophy comp.	Yes	Yes	Yes	White	Yes	No	No	No
	pBKS	AB resistance	Yes	Yes	Yes	White	Yes	No	No	No

Table 5.6: Summary of the experimental results of the second 10 OLGs. Table taken from [158].

OLG #	Plasmid	Characteristic	Restriction digest	Ligation	Transformation	Blue-white screening	Colony PCR - correct	Growth on selection plate	Growth of negative control	Repeat
11	pBSK	Auxotrophy comp.	Yes	Yes	Yes	Blue/no growth	X	X	X	Yes
	pBKS	AB resistance	Yes	Yes	Yes	White	No	X	X	Yes
12	pBSK	Auxotrophy comp.	Yes	Yes	Yes	White	Yes	No	No	No
	pBKS	AB resistance	Yes	Yes	Yes	Blue/no growth	X	X	X	Yes
13	pBSK	Auxotrophy comp.	Yes	Yes	Yes	Blue/no growth	X	X	X	Yes
	pBKS	AB resistance	Yes	Yes	Yes	White	Yes	Yes	Yes	No
14	pBSK	Auxotrophy comp.	Yes	Yes	Yes	White	Yes	No	No	Yes
	pBKS	AB resistance	Yes	Yes	Yes	White	Yes	No	No	No
15	pBSK	AB resistance	Yes	Yes	Yes	White	Yes	Yes	Yes	No
	pBKS	Auxotrophy comp.	Yes	Yes	Yes	Blue/no growth	X	X	X	Yes
16	pBSK	Auxotrophy comp.	Yes	Yes	Yes	White	Yes	Yes	Yes	No
	pBKS	AB resistance	Yes	Yes	Yes	White	No	X	X	Yes
17	pBSK	AB resistance	Yes	Yes	Yes	White	Yes	No	No	No
	pBKS	Auxotrophy comp.	Yes	Yes	Yes	White	Yes	No	No	No
18	pBSK	Auxotrophy comp.	Yes	Yes	Yes	White	Yes	No	No	No
	pBKS	AB resistance	Yes	Yes	Yes	White	Yes	Yes	Yes	No
19	pBSK	Auxotrophy comp.	Yes	Yes	Yes	White	Yes	No	No	No
	pBKS	AB resistance	Yes	Yes	Yes	White	Yes	No	No	No
20	pBSK	AB resistance	Yes	Yes	Yes	White	No	X	X	Yes
	pBKS	Auxotrophy comp.	Yes	Yes	Yes	White	No	X	X	Yes

5.4. Discussion

The results are not conclusive yet and besides repeating the experiment, another control group would be insightful, namely to test the original unaltered sequences on selection plates, which should always result in growth of the bacteria [158]. Nevertheless, the current result, namely that the constructed OLGs are not functional, would not be unexpected. While OLG construction does change the original sequences to a much lower extent than expected, proteins are highly complex molecular machines and function is very rare in sequence space.

The alignment creation for HMM profiles and OLG construction weights is very conservative in this study. After the protein group curation most sequences are from *E.coli*. This way only the variation of the proteins that occurs in the organism used for the experiments is considered. If the known sequences do not reflect all possible variations, and the weights and HMMs are too strict. It might be better to also take the variation of other but still similar organisms to *E.coli* into account in order for conservation weights and HMMs to more accurately reflect the functional range of the used proteins.

The OLG construction process could be further adapted if the reason for the loss of functionality is known. Translation could be strongly impeded by suboptimal RBSs and a high frequency of rare codons. The former could be included in the construction process to enhance RBS quality at the expense of MG quality instead of including the best RBS that does not change the MG. The rate of rare codons could simply be reduced by increasing the codon weight strength but this also comes at the cost of lower quality overlaps. Functionality could also be lost due to change in essential parts of the gene like binding sites or intra protein interaction sites needed for its function or correct folding. This could be improved by designing the sequences more manually and defining areas of importance in every gene that must not be changed. A more automated approach to such a procedure was used in [138], which randomly optimised constructed sequences further using long range intra protein interactions until the OLG pair converges to an optimum. Including such an OLG refinement step, the functionality of their constructed sequences could be proven in experiments.

Despite possible improvements it is also very interesting how much effort is actually needed to create functional OLGs. While determining the minimal effort needed is only of limited interest in synthetic biology, as experiments are very expensive, it is very interesting from a theoretical point of view as it indicates how likely OLGs could evolve naturally. Bringing the experiments of this study to a clear conclusion could help further expand on this question.

6. Conclusions

On first glance OLGs bring a big fitness disadvantage to an organism as these constructs are much more susceptible to mutations and also reduce the level of optimisation that can be acquired in a gene due to mutual restriction of both genes. Nevertheless many OLGs have been found in nature, which raises the question why they have not been lost in the course of natural selection. One explanation is that they bring a fitness advantage to the organism exceeding their weaknesses. Their wide spread across the taxonomic tree indicates that they are probably an old component of life and possibly carry an essential function overall. The goal of this study was to better understand the theoretical foundation of OLGs in their role for early life, their possible functions and their sheer existence and accessibility through random mutations. Progress has been successfully made in all these topics but much is still to be done to arrive at clear answers to these fundamental questions.

6.1.Code structure and code optimality and OLGs

The origin of the SGC is still very little understood. Some structure of the code must have been determined by its components, e.g. the specific block structure of the SGC, which is as rare as 1 in 10^65 random codes and therefore too unlikely to be a chance event, is partly determined by the translation system, namely the wobble binding of tRNAs inside the ribosome. But there is still a lot of variability left and many different properties of the SGC have been found to be rare in comparison to alternative codes. One explanation is that the SGC was optimised by a selection period in its evolution. As the origin of the SGC is still a very speculative field, some properties can be explained without a selection process by choosing a very specific evolutionary hypothesis. We found that the optimality of the SGC itself is a robust feature as other properties remain optimal after removing the optimality of one property by adjusting the restrictions on the alternative code set according to a specific evolutionary hypothesis. A more sophisticated evolutionary hypothesis could potentially be crafted such that all optimalities are explained. This is a very difficult problem as the SGC has many very different properties and possible variants of the SGC cannot be ruled out completely as alternative genetic codes have been found in nature. A finding which could arguably make a master-hypothesis of code evolution more likely, explaining all properties of the SGC without selective optimization, is that some properties are highly dependent on each other (for instance the similarity of frameshifted codons and those related by point mutations are strongly correlated [41]) which is something not often taken into account when new properties are discovered. Also some optimalities might be artefacts due to incorrect property calculation. For example the SGC has recently been claimed to be resource conserving by limiting mutations which strongly change carbon and nitrogen content in the respective AAs [165], which has been revealed to be an artefact due wrongly calculated mutation effects [166] and highly dependent on their choice of alternative genetic codes [167].
Combining different properties into a proto fitness function showed that the optimality can be increased and conditional probabilities can be derived. Especially with regards to properties with an expected less pronounced fitness advantage for the organism, e.g. OLGs, conditional optimalities are more likely to reveal their optimality. The mutational robustness seems to be the most important property of the SGC judging by its extreme code optimality. While consecutive

testing of different properties very likely produces artefacts in the results and should therefore be avoided, it is probably acceptable to use it only for mutational robustness as no other property could be added to this property without creating a stronger overall optimality. A more appropriate fitness function could possibly help further the understanding of the diverse properties incorporated in the SGC. Such a fitness function can only be derived if the complexity and environment of organisms evolving their genetic codes is known, which is a question strongly tied to the evolution of the first replicators and life itself.
While it is tempting to dismiss the difficulties of evolving a genetic code via a selective process by instead devising an evolutionary hypothesis such that no optimization is needed in order to reasonably explain the structure of the SGC, this is not a justified approach. Proof exists in the alternative genetic codes known today that the SGC is changeable even if only in small ways and these codes exist in modern organisms which are highly complex machines compared to the first replicators, which had to be much simpler in order to evolve at all. Much bigger changes in a genetic code of a simpler replicator should be possible and could add up to form the SGC known today. Studying the many unknowns in the early stages of code origin could clarify whether code could be the result of a selection process. A more detailed hypothesis which takes not only the genetic code but also the DNA/RNA and the translation mechanism into account could clarify what properties actually yield a fitness advantage, which parts are biochemically restricted and how much change such a system can endure without breaking.

6.2.A possible fundamental function of OLGs

If the formation of the SGC included a selection process, it would not be surprising if the SGC is also optimal for OLGs in some sense. Here multiple properties of the SGC are studied that it could perhaps have been optimised for, including some properties concerning OLGs. While no clear optimality in previously studied properties for OLGs could be found, the alternative reading frames are surprisingly similar in their conservation, which is quite optimal in the SGC and could reflect an unknown OLG property as it is not clear yet which functions OLGs have in an organism. All alternative reading frames maintaining a similar conservation value is a strong hint that this value optimises a function or property regarding OLGs. Conservation and coding flexibility are two especially opposed properties in OLGs that are both essential for the existence of OLGs and tie to their two biggest perceived challenges. We showed that such a tradeoff can optimise the average fitness of sequences in a rugged sequence space. While this is only a toy model, many other studies indicate that OLGs might play an important role in *de novo* gene creation. Using alternative reading frames as a place to search for new genes would not only be an optimal use of the energetically costly genetic material, but would also fit the narrative of OLGs playing an essential role in life, as new genes facilitate adaptation of life to new and ever changing environments. This idea must be challenged further by using more complex and realistic sequence evolution models, e.g. generation based models. The simplified split between conservative mutations (acted on by selection in our toy model) and explorative mutations (not selected in the model), used here to represent conservation and coding flexibility, should in future analyses be removed as selection can act on all mutations. Determining the ages of OLGs found across life could give insight into the hypothesis of OLGs as places of *de novo* gene creation in general.

6.3. The existence of OLGs – not so unexpected

A fundamental question that has to be answered before OLGs as a place for gene birth can be considered is whether the coding flexibility along an existing gene is sufficient for creating functional sequences to begin with. Artificially constructing OLGs of random protein domains in this study showed that the necessary change induced on both sequences is much lower than the variation observed between naturally occurring homologs. This was confirmed using similarity to HMMs, secondary structure and AA similarity/identity, which are mostly independent measures. We have found that some protein domains have only to be changed so little that it is imaginable that the overlap could have been formed by mutation and natural selection over a relatively short time. On the other hand, this could also mean that the two protein domains once formed an OLG pair, but no further studies have been made on these pairs. These artificial OLG constructs can be used to determine factors favouring OLG evolution which in turn could be used for OLG detection, e.g. overlapping ORFs in a very conserved region of a gene are less likely to be OLGs compared to ORFs overlapping very flexible parts of a gene. Studying taxonomic differences of OLG construction further could help determine fundamental differences between different organisms, which have been missed so far. This should consequently answer whether eukaryotic genes are actually the most suitable and virus genes the least suitable for OLG construction as found in this study and not some kind of artefact due to database biases.
Our experimental tests of the sequences were not fully conclusive, but hint that HMM scores, AA similarity and secondary structure are not sufficient to find successful sequences despite the high similarity to naturally occurring homologs in these measures. Nevertheless, a similar study also testing the functionality of constructed OLGs could show their functionality by also optimising for long range intra protein interactions. Since the constructed sequences are so similar to naturally occurring homologs, sacrificing some of this similarity to optimise for other properties like intra protein interactions or tertiary structure using novel prediction servers [137] is possible. This suggests that OLGs are not fundamentally harder to find by natural mutations compared to non-overlapping genes. While finding genes in sequences space is not well understood yet due to the perceived astronomically low proportion of sequences space being functional, there has to be a way as genes exist. Consequently, OLGs as a place of *de novo* gene creation cannot be discarded on the argument of low probabilities for finding functional sequences.
As a next step, the evolution of OLG constructs can be studied experimentally, e.g. by steadily increasing antibiotic concentrations, to see whether nature can optimise the OLG sequences to increase function or will eventually even split the two genes. The population dynamics of artificial overlapping genes and their mutations could reveal just how stable these constructs are, which is crucial for synthetic biology.

In summary, in this thesis I have shown that the existence of OLGs is not so unexpected, that OLGs could have much more fundamental possible functions than gene expression regulation and that these functions could be selected for in the SGC as its optimality is quite robust so a selection process is reasonable. This shows that despite decades of careful thought about OLGs more remains to be discovered.

Appendix A

Table A.1: Percentages of better codes than the SGC for the frame shift abortion times in reading frames '+2', '+3' and the average of both for all code sets in the first three columns. Percentages of better codes than the SGC for the mutational robustness for all code sets in the last column. Table taken from the supplement of [106].

Code Set	Reading Frame			Mutational Robustness
	+2	+3	Both	
Random	30.14%	37.66%	32.37%	0.0056%
Random_fs	7.14%	16.83%	3.64%	0.0059%
Random_faa	23.01%	31.35%	25.41%	$4 \cdot 10^{-8}\%$
Random_fb	6.22%	15.71%	3.19%	$5 \cot 10^{-8}\%$
Degeneracy	0.16%	3.70%	0.05%	$2.19 \cdot 10^{-15}\%^*$
Random_Blocks	39.23%	45.52%	40.56%	0.0022%
Blocks	26.09%	28.57%	24.09%	0.00068%
Random_fb_n	5.95%	14.58%	2.79%	$1.4 \cot 10^{-7}\%$
Degeneracy_n	0.14%	3.64%	0.04%	$1.2 \cdot 10^{-13}\%^*$
Random_Blocks_n	11.05%	23.53%	6.43%	0.095%
Blocks_n	22.60%	24.80%	20.56%	0.0055%
213_Model	6.00%	41.71%	4.10%	9.62%

Table A.2: Percentages of better codes than the SGC for the frame shift abortion times in reading frames '-1', '-2' and '-3'. Table taken from the supplement of [106].

Code Set	Reading Frame		
	-1	-2	-3
Random	50.45%	69.83%	38.94%
Random_fs	47.25%	88.94%	20.47%
Random_faa	47.02%	69.14%	32.57%
Random_fb	48.71%	90.38%	19.28%
Degeneracy	63.41%	97.89%	7.85%
Random_Blocks	57.11%	69.91%	45.95%
Blocks	46.47%	57.29%	32.85%
Random_fb_n	49.53%	89.67%	20.46%
Degeneracy_n	63.89%	98.05%	8.74%
Random_Blocks_n	47.41%	73.89%	22.46%
Blocks_n	43.80%	56.39%	29.73%
213_Model	0.00%	99.96%	22.46%

Table A.3: Percentages of better codes than the SGC for the conservation of alternative reading frames in all reading frames for all code sets. Table taken from the supplement of [106].

Code Set	Reading Frame				
	+2	+3	-1	-2	-3
Random	9.46%	9.70%	1.31%	16.97%	22.11%
Random_fs	9.41%	9.65%	1.29%	16.88%	22.04%
Random_faa	12.35%	12.65%	1.41%	23.52%	31.13%
Random_fb	12.34%	12.69%	1.40%	23.51%	31.12%
Degeneracy	0.081%	0.086%	1.01%	1.48%	4.24%
Random_Blocks	16.08%	14.57%	2.71%	98.06%	27.70%
Blocks	19.43%	17.77%	5.20%	95.39%	32.19%
Random_fb_n	11.72%	13.61%	1.32%	23.15%	31.89%
Degeneracy_n	0.085%	0.16%	0.87%	1.46%	5.38%
Random_Blocks_n	18.54%	15.62%	3.83%	95.79%	29.11%
Blocks_n	26.10%	24.09%	8.32%	96.13%	40.69%
213_Model	0.01%	3.41%	10.91%	58.58%	0.01%

Table A.4: Percentages of better codes than the SGC for the average ORF length in all reading frames for all code sets. Table taken from the supplement of [106].

Code Set	Reading Frame				
	+2	+3	-1	-2	-3
Random	70.46%	62.84%	49.77%	30.02%	30.21%
Random_fs	92.47%	82.95%	52.75%	11.51%	79.05%
Random_faa	77.55%	69.21%	53.16%	30.70%	67.74%
Random_fb	93.42%	84.04%	51.29%	10.01%	80.22%
Degeneracy	99.81%	96.11%	36.56%	2.24%	91.70%
Random_Blocks	61.50%	54.98%	43.06%	29.93%	54.23%
Blocks	74.17%	71.71%	53.61%	42.53%	67.44%
Random_fb_n	93.70%	85.22%	50.48%	10.74%	79.06%
Degeneracy_n	99.81%	96.11%	36.56%	2.24%	91.70%
Random_Blocks_n	88.68%	76.32%	52.59%	26.42%	77.05%
Blocks_n	77.61%	75.44%	56.25%	43.33%	70.55%
213_Model	91.45%	59.88%	100.00%	0.01%	77.73%

Appendix B

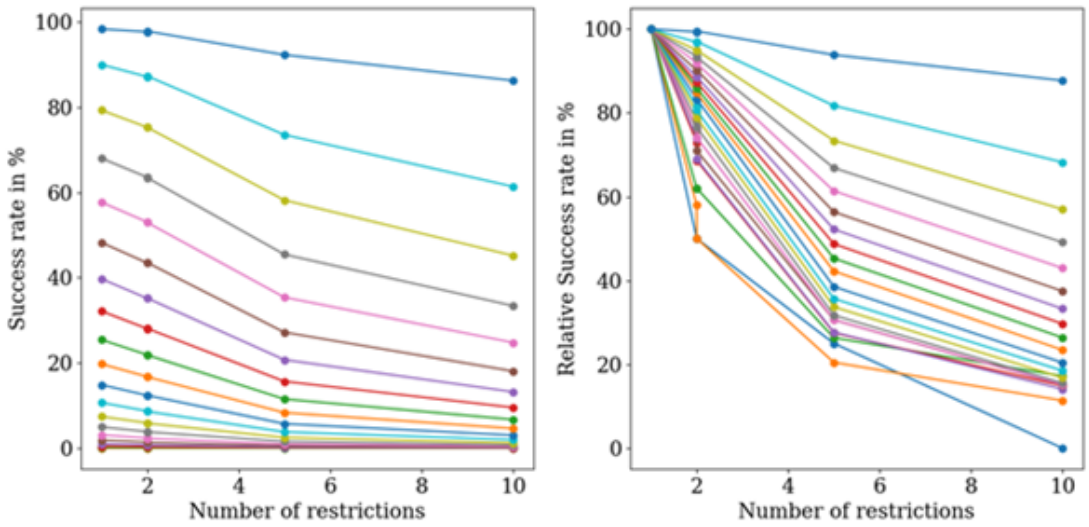

Figure B.1: Functional dependence between success rates of OLG construction and the number of combinatorial restrictions. The different lines represent different threshold percentiles, each subsequent line from top to bottom having a 5% percentile value starting at the threshold score value of the lowest scoring sequence in the protein family. Different percentile values can be compared by normalising each success rate value by the value in the '-3' frame which has the least number of restrictions. Figure taken from [139].

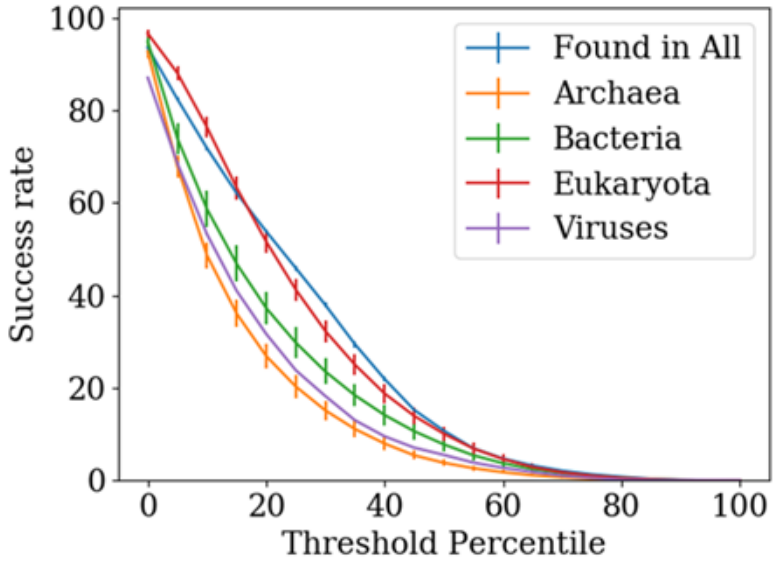

Figure B.2: Average success rates for different taxonomic groups. The average values and standard deviations, depicted by error bars, are calculated from 20 datasets with sequences of at least 70 AAs. The order of taxonomic groups is stable in a large range of threshold percentiles. Figure taken from [139].

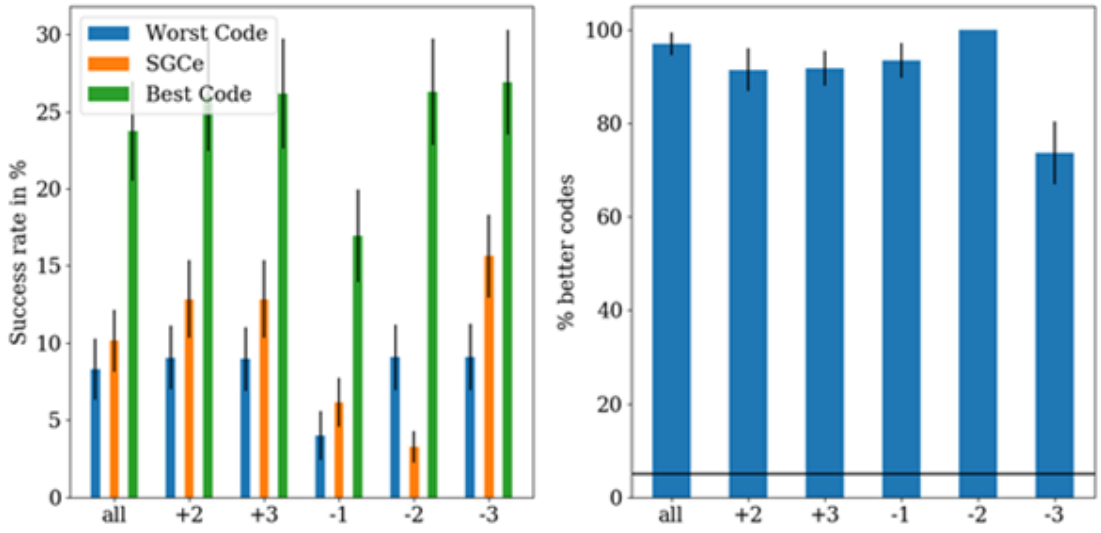

Figure B.3: Optimality of the SGC in OLG design split by reading frames in the 'Random' code set. The mean and standard deviation (black bars) values are calculated from 10 datasets of 150 sequences with minimum length of 70 AAs and 20 datasets of 100 alternative codes. *Left:* Comparison of average success rates in different reading frames. *Right:* The optimality of the SGC in different reading frames. The 5% threshold for optimality is indicated by the black line.Figure taken from [139].

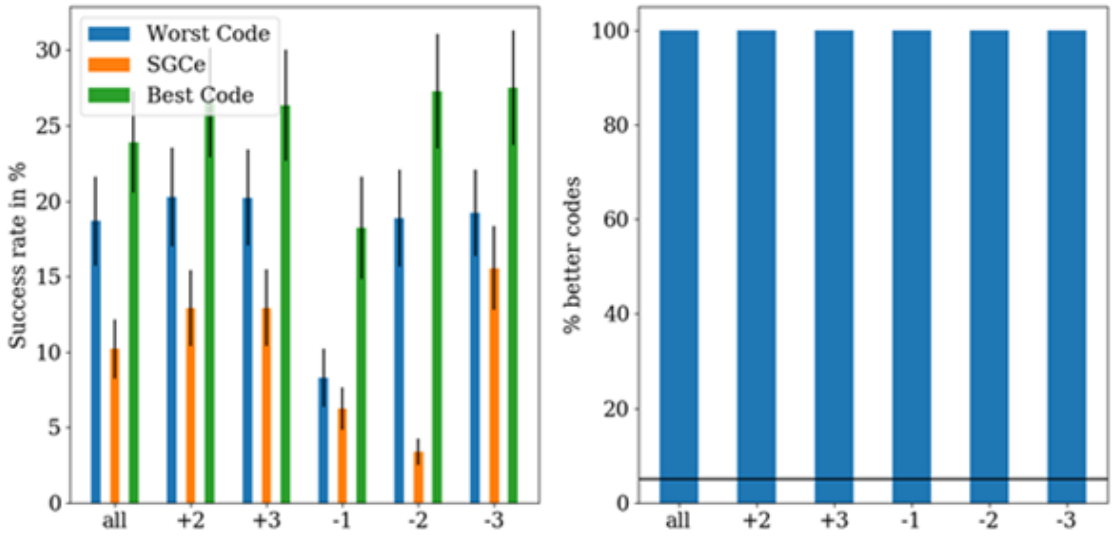

Figure B.4: Optimality of the SGC in OLG design split by reading frames in the 'Degeneracy' code set. The mean and standard deviation (black bars) values are calculated from 10 datasets of 150 sequences with minimum length of 70 AAs and 20 datasets of 100 alternative codes. *Left:* Comparison of average success rates in different reading frames. *Right:* The optimality of the SGC in different reading frames. The 5% threshold for optimality is indicated by the black line.Figure taken from [139].

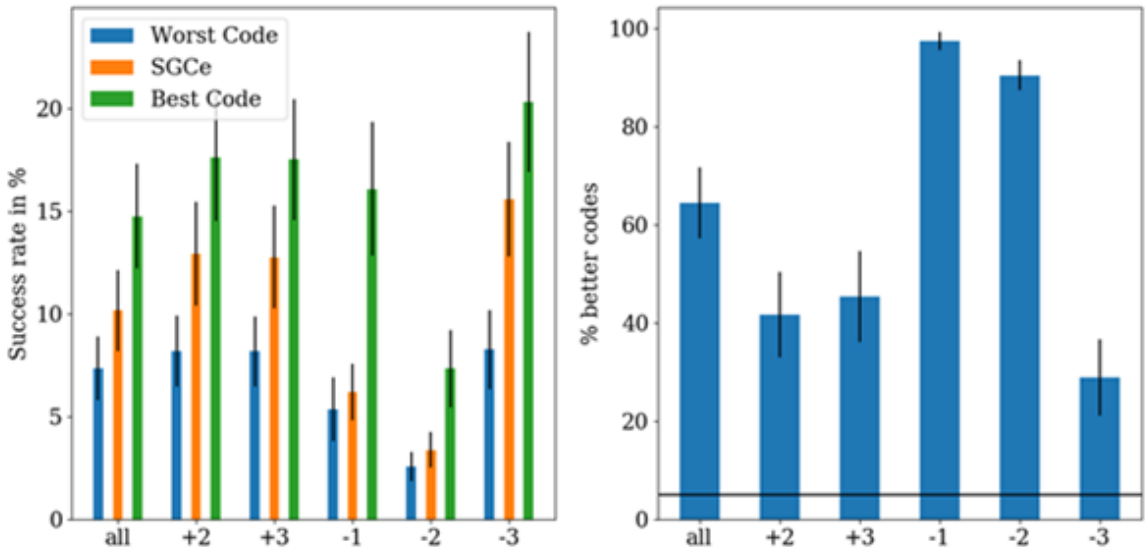

Figure B.5: Optimality of the SGC in OLG design split by reading frames in the 'Blocks' code set. The mean and standard deviation (black bars) values are calculated from 10 datasets of 150 sequences with minimum length of 70 AAs and 20 datasets of 100 alternative codes. *Left:* Comparison of average success rates in different reading frames. *Right:* The optimality of the SGC in different reading frames. The 5% threshold for optimality is indicated by the black line.Figure taken from [139].

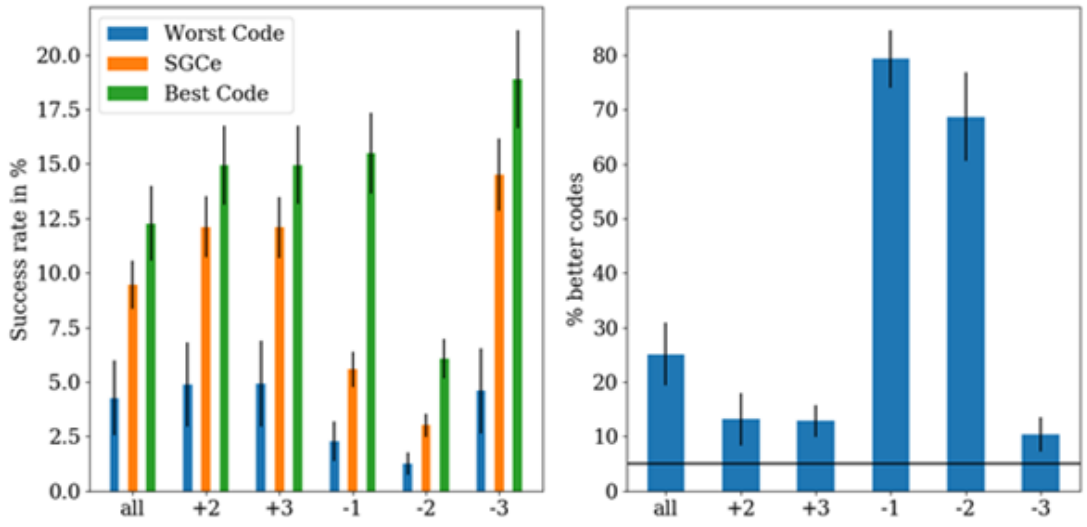

Figure B.6: Optimality of the SGC in OLG design split by reading frames in the 'MR-Blocks' code set. The mean and standard deviation (black bars) values are calculated from 10 datasets of 150 sequences with minimum length of 70 AAs and 10 datasets of 500 alternative codes. *Left:* Comparison of average success rates in different reading frames. *Right:* The optimality of the SGC in different reading frames. The 5% threshold for optimality is indicated by the black line.Figure taken from [139].

Appendix C

C.1. Original genes

List of the original nucleotide sequences of the genes, which have been overlapped and selected for experiments:

>N-acetylglutamatesynthase
MSATISPLAPKKYPKMPVIEGVRIATAEAGIKYKNRTDLLAMVFDPGTAVAGVFTRSKCPSAPVD
FCRQNLPDGKARVLVVNSGNANAFTGKKGKASTALTGEAAAKAAGCSQSEVFLASTGVIGEPL
DTTKFSHLLAGLVKDGKPDLWTEAAKAIMTTDTYPKVATATVKLGDADVTINGIAKGAGMIAPDM
ATMLSFIVTDAPIAAPALQDLLSRGTAKTFNAVTVDSDTSTSDTLLIFATGKAAARGAPAISDPKDA
RLGAFRRALGKVLKSLALQVVRDGEGARKQVEVTVTGAKSARSAKRIALSIANSPLVKTAVAGE
DANWGRVVMAVGKAGEPADRDRLSIWFGDNRLAHEGERDPSYSEEATSAYMKRDDIRIRADL
GIGRGKATVWTCDLTKEYVAINGDYRS

>Fosfomycin+thioltransferase
MSIKGLNHFLFSVSNLENSIAFYQNVFDAKLLVKGRSTAYFDLNGMWLALNQEKDIPRNEISHSY
THIAFSIEEQEFDKMYDKLNRLNVNILSGRPRDERDKKSIYFTDPDGHKFEFHTGTLQDRLDYYK
QEKTHMEFFD

>Fusaricacid+resistance
MPITFQALFAPSSLALKFAIKTLLGGGLALWLAMRWGLEQPSWALMTAFIVAQPLSGMVVQKGL
ARLAGTLVGTVMSVLFIGLFAQTPWLFLLTLALWLALCTAASTQLRSAWAYAFVLAGYTAAIIALPA
IDHPLQVFDQAVARCTEICLGIFCATASSALLWPMRVEQQLGGQARQAWQNGLQAARAMLGGE
DEARKGLLESLGRIVAIDSQREHAWFEGNRGRQRARAIRGLSQKLMVLLRISRSVRRQWRQLD
EREVEHLTPWLQEVRALLDQPDQPSLLLLRQRIWDAAHDEQISSAEHFCLARMALLLDYAMAAT
QALEDVEVGRAPKDVSQGLAAHRDWSLALLFGSRSALAFLVMSGFWLATAWPSAPGGLILTCV
VCSLFASRENGAQIGLSFLRGIFLAVPAAFLVGQIILPQWSSFAMLCLGMGVPLFLGALGMAHPR
TGATATSYCLHFIVLVSPLNAMQFGVATMLNSALAMLVGVSAAVMAFRLLVFRHPAWLGRRLRA
ATQNDLVRLTRRDLRGADSWFGGRMADRLMQLARHASELPEGERKRWDDGLHGLDIGDELVH
LRMCLAVAQAPLGPAEREYLQQVEAVLAKGPAAGRGQRLDAASEQFIAALRRLPASDPLRLAEG
AVLQLQKSWGKWCRWQEDTHGFA

>Pyrroline-5-carboxylate+reductase
MSNTRIAFIGAGNMAASLIGGLRAKGLQASHIRASDPGEETRQRVSAEHGIETFADNAQAIDGV
DVIVLAVKPQAMKAVCEAIRPSLKPHQLVVSIAAGITCASMTAWLGEQPIVRCMPNTPALLRQGV
SGLYATSEVTAEQRQQAEELLSAVGIALWLDEEQQLDAVTAVSGSGPAYFFLLIEAMTAAGVKLG
LPKEIAEQLTLQTALGAAHMAVSSDVDAAELRRRVTSPNGTTEAAIKSFQADGFEALVEKALGAA
AHRSAEMAEQLGK

>Argininosuccinate+lyase
MSTDKTNQSWGGRFSEPVDAFVARFTASVTFDQRLYRHDIMGSIAHATMLAKVGVLTDAERDSI
IDGLNTIQGEIEAGQFDWRVDLEDVHMNIEARLTDRIGVTGKKLHTGRSRNDQVATDIRLWLRDE

IDLILAEITRLQKGLLEQAEREAESIMPGFTHLQTAQPVTFGHHMLAWFEMLSRDYERLVDCRKR
TNRMPLGSAALAGTTYPIDREYTAQLLGFDAVGGNSLDNVSDRDFAIEFCSAASIAMMHLSRFS
EELVLWTSAQFQFIDLPDRFCTGSSIMPQKKNPDVPELVRGKTGRVFGALMGLLTLMKGQPLAY
NKDNQEDKEPLFDAADTLRDSLRAFADMIPAIKPKHAIMREAALRGFSTATDLADYLVRRGLPFR
DCHEIVGHAVKYGVDTGKDLAEMSLEELRQFSDQIEQDVFAVLTLEGSVNARDHIGGTAPAQVK
AAVVRGQALLASR

>AmGl+acetyltransferase
MDIRQMNKTHLEHWRGLRKQLWPGHPDDAHLADGEEILQADHLASFIAMADGVAIGFADASIR
HDYVNGCDSSPVVFLEGIFVLPSFRQRGVAKQLIAAVQRWGTNKGCREMASDTSPENTISQKV
HQALGFEETERVIFYRKRC

>AmGl+nucleotidyltransferase
MRTEKEMLDVIINIAKEDERIRAVIMNGSRVNPNVKKDCFQDYDIMYVVNDIQSFTSNHNWIHRF
GEIMIVQMPEEMSLVPPDEDGKFPYLMQFMDGNRIDLTLVPVELIKKFVGQDSLSKLLLDKDNCL
EEFPPASDKDYLIKKPTEKEFLDCCNEFWWCSTNVAKGLWREELSYAKGMLEGPVRDMFIVML
EWHIGMKTDFTVNTGKFGKHFEQYIEEDMWEQFKRTFSNAEYENIWESFFVMGDLFREVANEI
ANTYEYQYPQDEDDKVTNYLKHVKALPKDSTSIY

>Argininosuccinate+lyase
MSTDKTNQSWGGRFSEPVDAFVARFTASVTFDQRLYRHDIMGSIAHATMLAKVGVLTDAERDSI
IDGLNTIQGEIEAGQFDWRVDLEDVHMNIEARLTDRIGVTGKKLHTGRSRNDQVATDIRLWLRDE
IDLILAEITRLQKGLLEQAEREAESIMPGFTHLQTAQPVTFGHHMLAWFEMLSRDYERLVDCRKR
TNRMPLGSAALAGTTYPIDREYTAQLLGFDAVGGNSLDNVSDRDFAIEFCSAASIAMMHLSRFS
EELVLWTSAQFQFIDLPDRFCTGSSIMPQKKNPDVPELVRGKTGRVFGALMGLLTLMKGQPLAY
NKDNQEDKEPLFDAADTLRDSLRAFADMIPAIKPKHAIMREAALRGFSTATDLADYLVRRGLPFR
DCHEIVGHAVKYGVDTGKDLAEMSLEELRQFSDQIEQDVFAVLTLEGSVNARDHIGGTAPAQVK
AAVVRGQALLASR

>Isopropylmalate+dehydrogenase
MSKQILILPGDGIGPEIMAEAVKVLELANDKYSLGFELSHDVIGGAAIDKHGVPLADETLDRARAA
DAVLLGAVGGPKWDKIERDIRPERGLLKIRAQLGLFGNLRPAILYPQLADASSLKPEIVSGLDILIV
RELTGGIYFGAPRGTRELENGERQSYDTLPYSESEIRRIARVGFDMARVRGKKLCSVDKANVLA
SSQLWREVVEQVAKDYPDVELSHMYVDNAAMQLVRAPKQFDVIVTDNMFGDILSDEASMLTGS
IGMLPSASLDANNKGMYEPCHGSAPDIAGQGIANPLATILSVSMMLRYSFNLTEAADAIEQAVSR
VLDQGLRTGDIWSAGCTKVGTQEMGDAVVAALRNL

>O-succinylhomoserine+lyase
MTRKQATIAVRSGLNDDEQYGCVVPPIHLSSTYNFTGFNEPRAHDYSRRGNPTRDVVQRALAE
LEGGAGAVLTNTGMSAIHLVTTVFLKPGDLLVAPHDCYGGSYRLFDSLATRGCYRVRFVDQGDE
RALQAALEEKPKLVLVESPSNPLLRVVDIAKICRLAREAGAVSVVDNTFLSPALQNPLALGADLVL
HSCTKYLNGHSDVVAGVVIAKDPEMVTELAWWANNIGVTGGAFDSYLLLRGLRTLVPRMELAQ
RNAQAIVDYLQTQPLVKKLYHPSLPENQGHEIAARQQKGFGAMLSFELDGDEETLRRFLGGLSL

FTLAESLGGVESLISHAATMTHAGMSPQARAAAGISETLLRISTGIEDGEDLIADLENGFRAANK
G

>Diaminopimelate+decarboxylase
MDAFNYRDGELFAEGVSLTAIAERFGTPTYVYSRAHIEAQYNAYADALSGMPHLVCFAVKANSN
LGVLNVLARLGAGFDIVSRGELERVLAAGGSADKIVFSGVGKTRDDMRRALEVGVHCFNVEST
DELERLQVVAAEMGVRAPISLRVNPDVDAGTHPYISTGLKENKFGIAIADAEDVYIRAAQLPNLE
VVGVDCHIGSQLTTLDPFIDALDRLLALIDRLGDCGIYLRHIDLGGGLGVRYRDEEPPLAADYIKA
VRERIEGRDLALVFEPGRFIVANAGVLLTQVEYLKHTEHKDFAIVDAAMNDLIRPALYQAWMDVT
AVRPRDTEARAYDIVGPICETGDFLAKDRQLALAEGDLLAVHSAGAYGFVMSSNYNTRGRAAEV
LVDGDQAFEVRRRETVAELFAGESLLPE

>Chloramphenicol+acetyltransferase
MKFHVIDREDWNREQYFEHYLKLKCTFSMTVNVDITMLLEEVYQKGIKFYPVFIYLISRVVNNHK
KFRTCFNDEGVLGYWEEMIPSYTIFHKDDKSFSSIWTDYSSDFRTFYKNYEDDMRCYASVHGL
FTKENIPPNVFPISSIPWTSFTGFNLNINNDENFLLPIITCGKYFNEGNKVMLPVSLQVHHSVCDG
YDASQFIEDLQQLSNTCNEWLK

>Serine+hydroxymethyltransferase
MLKREMNIADYDAELWQAMEQEKVRQEEHIELIASENYTSPRVMQAQGSQLTNKYAEGYPGKR
YYGGCEYVDIVEQLAIDRAKELFGADYANVQPHSGSQANFAVYTALLQPGDTVLGMNLAQGGH
LTHGSPVNFSGKLYNIIPYGIDESGKIDYDDMAKQAKEHKPKMIIGGFSAYSGIVDWAKMREIADS
IGAYLFVDMAHVAGLIAAGVYPNPVPHAHVVTTTTHKTLAGPRGGLILAKGGDEELYKKLNSAVF
PSAQGGPLMHVIAAKAVALKEAMEPEFKVYQQQVAKNAKAMVEVFLNRGYKVVSGGTENHLFL
LDLVDKNLTGKEADAALGRANITVNKNSVPNDPKSPFVTSGIRIGSPAVTRRGFKEAEVKELAG
WMCDVLDNINDEAVIERVKGKVLDICARFPVYA

>Threonine+synthase
MTHQWRGIIEEYRDRLPVSDTTPVVTLREGGTPLVPAQVLSERTGCEVHLKVEGANPTGSFKD
RGMTMAISKAKEEGAKAVICASTGNTSASAAAYAVRAGMVSAVLVPQGKIALGKMGQALVHGAK
ILQVDGNFDDCLTLARSLSENYPVALVNSVNPVRIEGQKTAAFEIVDMLGDAPDIHVLPVGNAGN
ITAYWKGYKEYAADGIATRTPRMWGFQASGSAPIVRGEVVKDPSTIATAIRIGNPASWQYALAAR
DESGGAIDEVTDREILRAYRLLAAQEGVFVEPASAASVAGLLKAAEQGKVDPGQRIVCTVTGNG
LKDPDWAVAGAPQPVTVPVDAATAAERLGLA

C.2. Constructed OLG sequences

List of nucleotide sequences of the MGs (=longer sequence of the OLG pair) in the first set of 10 selected OLG sequences used in the growth experiments:

OLG #0
>N-acetylglutamatesynthase_Fosfomycin+thioltransferase_234_4_0_1124_CAT_S2_15

GAATTCATGAGCGCGACCATTAGTCCACTGGCTCCTAAAAAATATCCCAAGATGCCGGTAAT
CGAGGGGGTTCGCATCGCAACAGCAGAAGCGGGTATTAAGTATAAAAACCGGACGGATTTG
CTTGCCATGGTCTTTGATCCCGGCACCGCCGTGGCCGGTGTGTTTACACGGTCCAGTAAAT
AGCGGGAACGCAAACGCGTTCACTGGTAAAAAAGGTAAGGCATCAACCGCGCTGACCGGA
GAGGCCGCGGCCAAAGCCGCTGGCTGTTCTCAATCGGAAGTCTTCCTGGCTAGTACCGGT
GTCATTGGTGAACCGCTGGACACCACTAAGTTTTCGCACCTCCTCGCCGGGTTGGTTAAAG
ATGGTAAACCGGATCTCTGGACTGAAGCAGCGAAAGCTATTATGACGACAGACACATATCCT
AAAGTGGCAACGGCGACCGTCAAGCTGGGCGATGCTGATGTAACTATTAATGGTATAGCAAA
AGGTGCTGGCATGATTGCCCCTGATATGGCTACCATGTTGTCTTTTATCGTCACCGATGCGC
CAATTGCCGCACCGGCTCTACAAGATCTCTTATCCCGTGGAACCGCGAAAACCTTCAATGCA
GTGACGGTTTCTACAGATACCAGTACATCTGATACCATTCTAATTTTTGCGACTGGTCGCGGT
AGTAGTCGAGGCGGTCCTGCAGTGTCCCAGTATGAAGATCAAATTTATGGCCGTCTGGATC
GGTCATATAAACAGATTTTTTATCGCGTTCGTCTTCAGGTCGTCCGCTCAGGAGATTCAGCG
AGACGCGATTTAGACGTTCGTATAACTGGTCAAAAGTCTGCTCGTTCAGCGAAAAGGCTAGC
TCTGTCCATTGCGAACTCACCTCTGATCGCGGCAGCTGTTTCTGGCGATTCAGCGCAATGG
GGACGCCTTGTAATGGCAGTTGGGCGGTCTGGCGACCCTGCAGATAAAGATCGGCTTCAAA
TATGGTTTGGTAAAAACCGACTGAGTCACGAAGGTTCGAGAGATCCAGCATATAATGATTCA
GCCCTTTCAGCGACATTAAAAAGAGACGACATCCGTATCCGAGCGGATCTGGGTATCGGTC
GTGGCAAAGCGACCGTTTGGACATGCGACCTGACCAAAGAGTATGTGGCGATCAACGGTG
ATTATCGTAGCTGAAAGCTT

OLG #1
>Fusaricacid+resistance_Pyrroline-5-carboxylate+reductase_353_4_0_1877_CAT_S23_4
GAATTCATGCCGATCACCTTCCAAGCCTTGTTTGCCCCGAGTAGCTTGGCGCTGAAGTTTG
CTATCAAGACCTTACTGGGTGGCGGTCTGGCCCTGTGGTTAGCAATGCGATGGGGTCTGGA
ACAGCCATCGTGGGCATTAATGACGGCGTTCATTGTAGCACAACCTCTGTCGGGCATGGTC
GTTCAGAAAGGGTTAGCCCGGCTCGCCGGCACCTTAGTCGGAACCGTGATGAGCGTCCTT
TTCATTGGTCTGTTTGCGCAGACGCCTTGGCTATTTTTACTTACCCTGGCATTATGGCTGGCA
TTGTGCACCGCTGCGTCCACTCAGTTGCGGTCTGCTTGGGCCTATGCGTTCGTGCTTGCC
GGGTACACCGCCGCGATTATCGCCCTGCCGGCCATCGATCATCCCCTGCAGGTATTCGACC
AGGCCGTCGCTCGTTGCACGGAGATTTGCCTGGGCATTTTTTGTGCCACCGCTTCGTCCGC
GCTGTTGTGGCCAATGCGCGTGGAACAGCAGTTGGGTGGACAGGCGCGACAGGCGTGGC
AAAATGGTTTGCAAGCGGCCCGCGCCATGCTCGGCGGGGAAGATGAAGCCCGCAAAGGG
TTATTGGAATCGCTCGGCCGCATCGTAGCGATTGACTCCCAGCGAGAGCACGCGTGGTTTG
AGGGCAATCGCGGCCGTCAGCGCGCTAGAGCAATTCGTGGTCTGAGCCAAAAGCTGATGG
TGCTCCTGCGCATTTCCCGTAGCGTACGACGTCAGTGGCGTCAACTGGATGAGCGAGAAG
TTGAACACTTGACTCCGTGGCTCCAGGAAGTCCGAGCCCTTCTGGACCAACCCGACCAGC
CGAGCCTGCTTCTCCTGCGCCAGCGCATCTGGGACGCGGCCCACGATGAACAAATTTCCT
CAGCTGAGCATTTTTGCTTAGCTCGCATGGCCCTTCTGCTGGATTATGCGATGGCAGCTACC
CAGGCATTGGAGGATGTGGAAGTGGGCCGTGCACCAAAAGATGTGTCCCAGGGGTTGGCT
GCGCATCGCGACTGGTCTCTTAGCCTGCTGTTTGGTTCTCGCCCAGCTGTTGCGCTAATTC
TGCTGAGCGGTTTTTGGCTGCGGACAGCGTGGCCGTCAGCACCAGGCGGAATCCTTCTGA
CTGAAATGATTTGCTCGCTGTTTCTGTCGCGCCGTTCGGGCTCGCAACTCGGCGTCTCATT
TCTGCGCGGTCTATATCTGAGCTTACCGCCATGTGTGCTGCTCGGACAGCTGATTCTGCCG

CAATGGTCTGCGCTAGCTCTTCTGGCAATCCCAATGGGTATCCCGCTGCTGTTAGGGCTTCT
AGGATTAAAAAAAAATAGGACGGGCCTGACCCTGACATCGCTGTGCCTGCATCTAATTGTTC
TTCTTCATCCAGCCACAGCGCTACGCCTAGGGCTGACAACAGTTCTTCACTCTGCTGTCGC
TGTTCTGCTGTCAGTTCAGAGCGCGCGTACAGCCCTGAAACTCCTTGTTTTAAAAGAGCCG
GCGTGGCTGGGACGCACCTTACGAGCGGCTGTTCGCCAAGCCATGCTGAGACTGACGCG
CAGGTCACTCCGCGGCGCAGACTCATGGTTTGGCGGTGCGGTTGCAGATCAGGTCTTACA
GCTTCGCAAACACGCTTCAGAGCTTCCGGACGGACAGCGTAAACGATGGGATCAAGGCCT
TCACGGGCTTGATATTGGTCAAGAAATGATTCATATCCGTGTTCTGCTGGCACTCGCTGACG
CGCCTCTTGGCCCGGCTGAGAGGCCCTATCTTCAGCAGCTTGAAGCCCTTCTGCGCGAAG
GCCCGGCAGCAGGGCGCGGGCAACGTTACCAGCGCCAATCAGAGCAATTTCTGTCTGCGA
CATCACGCCTCCCAGCCTCCGACCCGCTGCGCCTGGCGGAAGGTGCAGTGTTGCAGTTG
CAGAAATCTTGGGGAAAATGGTGCCGCTGGCAAGAAGACACCCATGGCTTTGCATAAAAGC
TT

OLG #2
>Argininosuccinate+lyase_AmGl+acetyltransferase_239_4_0_1154_TAT_S1_3
GAATTCATGTCGACAGATAAGACGAACCAAAGCTGGGGTGGCCGCTTTAGCGAGCCGGTAG
ACGCTTTCGTCGCACGATTCACCGCATCTGTCACCTTTGACCAACGTCTGTACCGCCACGA
CATTATGGGAAGCATCGCTCACGCTACCATGCTGGCCAAAGTGGGCGTGTTAACTGATGCG
GAACGTGATAGTATCATCGATGGCTTGAATACCATTCAGGGGGAGATTGAAGCTGGGCAGTT
CGATTGGCGGGTGGATCTCGAAGATGTGCATATGAATATTGAGGCTAGACTGACCGATCGCA
TTGGGGTTACCGGCAAAAAGCTGCACACGGGGCGCAGCCGAAATGATCAAGTTGCCACTG
ATATTCGGCTGTGGCTGCGCGACGAAATCGACCTGATACTGGCGGAAATTACGCGTCTGCA
GAAAGGGCTGCTGGAACAAGCGGAGCGCGAAGCGGAAAGTATAATGCCGGGCTTCACCCA
TCTGCAGACGGCGCAGCCCGTGACCTTTGGTCATCACATGCTCGCGTGGTTCGAAATGCT
GTCTCGCGATTACGAACGTCTAGTTGATTGCCGGAAACGCACCAACCGTATGCCGCTAGGC
TCCGCGGCGCTGGCGGGCACGACTTATCCGATTGATCGTGAATATACAGCCCAGTTACTGG
GCTTCGATGCGGTGGGTGGCAACTCACTGGATAACGTTACAGATAGAGATTTTGGCATTTCT
TTTTGTAGTGCAGCAAGCGTTGCGTTTCTTCATATCCAACGCTTTGATGAACAGATTGTGATA
TGGTGTTCTGCGGACTTTCAGTTTCTGCAACTTCCTGACAGGTTTTGTTCAGGCCCCAGCG
TTGTACCGCAGAAATCAAATCCTGACGTACCCCGCGTTGTTCGTGGCTCGGCAGGACGAAT
ATTTGGCGCAGAAATGGGACTGGTGACGCTTCGCAAGGGTTCACCGCTTCGTTACGAACG
CTCGAATCAGGAAAACCAAGAGCCACTGTTTGACGCAGCGCCAACACTGAGGCCAGATGT
TCGCGCTGCAGCAGATCTTCTCCCGGCGCTAAAGCCGCGTCATCTGGATGTCCAGGCCAC
AGCTGTTCGCGGATTCCGCGCCAGTACTGAATATGCGTCTTATTTAATTCGCGAAGGTCTATC
ATTCCGTGACTGTCATGAGATTGTTGGGCATGCTGTGAAATATGGTGTGGACACGGGGAAG
GATCTTGCTGAAATGTCTCTGGAGGAGCTGCGTCAGTTCTCTGACCAAATTGAGCAGGATG
TGTTTGCAGTCTTAACATTAGAAGGTAGTGTTAATGCACGTGACCATATTGGGGGTACGGCT
CCGGCCCAAGTGAAAGCCGCCGTGGTCCGTGGACAAGCGCTGTTGGCTAGCCGTTGAAA
GCTT

OLG #3
>AmGl+nucleotidyltransferase_Pyrroline-5-carboxylate+reductase_8_4_0_842_AAT_S13_5

GAATTCATGCGCACTAAAAAAGAAATGCTGGACGTCCTAATTGATCTGCCATCTCAGCAGAA
CGATGTGCGCGCGCTCCTAATGAACGGTTCACGAGTGAATCCCAACCTGAAGCCTGATTGC
TTTCAAGACTACGATCTGTTGTACCTTGTGGGCTCGTTACGCGCCTTTTCAGCTCAGCACAA
TTGGTATCATCGCTTTGGGCCATTGATGCTGCTCCAAATGCCGTCTGAAATGTCAGTTGTTC
CGCCAGATCGCGAGGGACGCCTACCTTATATCCTGCAGCTGTTAGACGGTCAACGAGTAGA
TAAAACACTGCTGGACCTGAACCTGATACAGCAGTTTGTGCATCAAGATTCTCTTCTTCGTC
TAACCACAGATAAAGACCAATGCCTGAAAGAATTTCCTCCGCCTGCTGACGCTGATTACCTG
TTACGGAAGCCGACGCGTAAAGACCTGATAGATTGTTGCGAAGAATTTTGGTGGTGTTCGG
CAAACATCGCAAAATCGGTTTGGCGCCAAGCCATGCAGTACGCGAAAGGCATGTTACGCCC
GCCGCTACGCTCACAATTAATTGTTCTGGTTGAATGGCAGGTCGGACTGCGTCGCAAATTTC
TTTTAAACACTGGGGTTTTTGGCAAAGAATTTGAACAGTATATCCAGGAATCGCTTTGGCGTT
ATCTGAAAAGGACTCTAAGCCATGCTGAATACGAACGCGTTTGGCAGTCTTTTTTCCTGGTT
GGCTCGCTTTTTCGTGAGATTGCTGAAGACCTTGCGCGCGCATACCGCCAACAATATCCGC
AGCAAGATGACCAGCGCCTAAAAAACTATCTGAAACATCTGAAATCACTACCTAAAGACAGC
ACCAGTATTTATTAAAAGCTT

OLG #4
>Argininosuccinate+lyase_AmGl+acetyltransferase_222_4_0_1103_GAT_S13_5
GAATTCATGAGCACAGATAAAACGAACCAGAGCTGGGGTGGTCGTTTTTCAGAACCCGTCG
ACGCGTTCGTTGCGCGTTTTACAGCTAGTGTTACGTTTGATCAGCGTCTATATCGCCACGAC
ATTATGGGTAGCATTGCTCATGCCACAATGCTTGCTAAAGTGGGGGTACTGACGGATGCTGA
ACGGGATAGCATCATCGACGGTTTGAACACAATCCAGGGAGAAATCGAGGCCGGTCAATTT
GATTGGCGTGTCGATTTAGAGGACGTTCATATGAACATCGAGGCTCGGCTAACGGATCGCAT
TGGTGTTACTGGGAAAAAGCTACACACTGGCCGTTCGCGCAATGACCAAGTGGCCACTGAC
ATCCGCCTGTGGCTGAGGGACGAAATTGATTTGATTCTGGCGGAAATTACGCGACTACAGA
AAGGTCTGTTAGAGCAGGCTGAACGCGAGGCGGAAAGCATAATGCCGGGCTTTACCCATCT
GCAGACCGCGCAACCGGTCACTTTTGGCCATCATATGTTAGCGTGGTTTGAGATGTTATCCC
GTGACTATGAGCGTCTGGTAGACTGTCGTAAACGCACAAACCGGATGCCGCTGGGTTCGG
CTGCTTTGGCGGGGACTACCTACCCAATAGACCGTGAATATACTAGCCAGCTGCTGGGTTAC
AACGCTTACGGTAAAAAATCACTCGATCAGATTCCTGATAACCAATTTGCTGATGAATTTTGTT
CTGCAGCGTCGATTGCGGTGATGCATCTGACGCGATTTCGCGACAACCTTTATTTGTGGAC
CAGCGCTGAATTTCAGCTATTAGATCTTCCGCAACGCCTCTGTACCGGAAGCTCGGTAATGC
CAGAACGCCGTCAACCAGATATACCGGAGCTGATTCGCAACCGTTCAGGTAATCTCTTCGG
AGCGCTGCTTGGGCTGCTGACACTGCTAAAGGGTCAGCCATTGCAATATAACTCGCCAAAT
GATCAGGATAAAGAACCTCTTTTTGATCTAGCAGATGCGCTTCGTCCGGATGTCCAGGCCAC
AGCTGACGTCGTACCGGCCGCCAAACCGAAACATGCGATTTATTCATCTGCCGCACTTCGA
TCATTCTCCACGGCCACGGATCTCGCCGATTATCTGGTCCGACGTGGTTTACCCTTCCGGG
ACTGTCATGAGATAGTCGGTCATGCTGTCAAGTATGGCGTAGATACCGGTAAAGATCTGGCC
GAAATGTCCCTGGAAGAACTGCGTCAGTTTTCGGACCAGATTGAACAGGATGTCTTTGCTG
TACTGACCCTGGAGGGGAGCGTGAACGCCCGTGATCACATCGGAGGGACAGCCCCCGCG
CAAGTAAAAGCGGCGGTGGTCCGCGGTCAGGCGTTACTCGCGTCTAGATAAAAGCTT

OLG #5
>N-acetylglutamatesynthase_Fosfomycin+thioltransferase_237_4_0_1133_CAG_S13_6

GAATTCATGTCCGCAACTATAAGTCCGCTGGCCCCGAAGAAGTATCCTAAAATGCCGGTTAT
CGAGGGTGTTCGTATTGCCACCGCCGAAGCGGGTATCAAATACAAAAACCGCACCGACTTA
TTAGCGATGGTTTTTGATCCAGGTACAGCTGTAGCTGGTGTTTTTACCCGCAGCAAATGTCC
GTCTGCGCCCGTCGACTTTTGTCGGCAAAACCTGCCGGACGGCAAAGCGCGCGTTCTTGT
CGTAAACAGCGGTAACGCGAACGCATTTACAGGTAAAAAGGGAAAGGCTTCGACGGCGTTA
ACAGGAGAAGCAGCCGCGAAGGCGGCGGGCTGTTCCCAGAGCGAAGTATTTTTGGCCTCG
ACAGGTGTGATTGGTGAACCTTTAGACACCACCAAATTTAGCCATTTATTAGCCGGCCTGGT
TAAAGACGGTAAACCGGATCTGTGGACAGAGGCGGCAAAAGCGATAATGACCACGGACACA
TACCCGAAAGTCGCTACCGCAACCGTTAAATTGGGTGATGCGGACGTCACGATCAATGGGA
TCGCGAAAGGTGCCGGGATGATTGCCCCAGATATGGCCACCATGCTGTCGTTCATCGTGAC
GGATGCGCCCATCGCGGCGCCGGCCCTGCAGGATCTGCTCTCCCGCGGTACCGCCAAAA
CGTTCAATGCTGTGACTGTCGACTCTGATACTAGTACATCAGATACGCTTCAAATATTTGCATA
TGGGTCCGCTGCTGCTCGTGGTAGTCCAGCCGTTCCTGAACCGCGCCAGTATGAAATTCAA
ATTTTTCGCCGTCTGGTTGGGAAAAATATACGCTCTCTTTGTCTTCAGATTCTCGCGGACGG
CCAGGGAGCACGGAAACGTTTAGACGTGACAGTTTCTGGTGCATCCGATCAAAGGTCTGCT
CGTCGACTGAAAATGCAATATGCGAATAGCCCGCTGATAAAGACTGCCGTGGCAGGTCGCG
ATGCGAATTGGGGGCGAGCCATAATCCGTTTAGGTCAAAGTGGGCAGCCGGCCGACCGCG
AACGATTAAGCGTGTGGTTTGGAGATTCTCGTATAGCGCAATCTGGTTCTCGAGATCCGAAA
TACTCAGAAGAAAATGATTCAGCCCTTTTAAAGACAGACGACATCCGCATTCGGGCCGACCT
GGGCATCGGCAGGGGGAAAGCGACGGTGTGGACTTGCGACCTCACGAAAGAGTATGTCG
CGATAAACGGCGATTATCGGAGTTAAAAGCTT

OLG #6

>Homoserine+O-succinyltransferase_Fosfomycin+thioltransferase_116_4_0_770_CAT_S13_5

GAATTCATGCCTACTGCGTTTCCTCCCGACTCCGTGGGTCTGGTAACGCCACAACTGGCCC
ATTTCAGCGAACCGTTAGCCTTAGCCTGTGGACGCAGCCTGCCCGCATATGATCTCATCTAC
GAAACGTACGGCCAGTTAAACGCAAGCGCGTCGAACGCAGTCCTTATTTGTCATGCACTGT
CCGGCCATCATCATGCTGCGGGGTATCATTCGGTGGATGATCGTAAGCCGGGATGGTGGGA
TTCCTGTATTGGTCCGGGTAAACCTATTGATACCAACAAATTCTTCGTGGTATCCCTGAACAA
CTTGGGTGGCTGTAATGGTTCCACCGGGCCTAGCTCGCTGAACCCAGAGACAGGAAAACC
ATATGGCGCTGATTTTCCTTATAATACTGCAGACGATCTTGTACATAGCCAGTATGAAATTGCA
GATCGTGTCGGTCTGGATCAGTGGTGTGCAGTGATTGGCGGTCGCGTTGGCGGCGTGGAC
GCCCTGCAATGGCGTTTACGTTATCCTGATTCAGTTCGTCATTGCCTTTGTCTAGCTTCTGCT
CCGAAACTGAGTGCGCAAAATGTGTATATGAATGAGATAGCTCGTTACGCGGTACTGACTGA
TCCTGATTTTCACGGAGCCACATTCCAGGAAGATCAAATTGTGCCGAAGCGCGGCCTTTTAT
TAGCAGAGATGCTTGGAAAACTGACATATAAAAGCGAAGACTCGATGGGAGATTACTTTGGG
AAAACAAAAAATGATGAACGCCTGAAATATTCATTCCACTCCGTGGAGTTTCAAGTGGAATCT
TACCTGCGGTATCAGGGTGAGGAATTTTCAGGCCGTTTTGATGCAAATACGTACCTTCTGAT
GACCAAAGCGCTCGACTATTTCGATCCGGCAGCGAATTTTGACGATGATTTGGCAAAAACCT
TTGCAAATGCAAGTGCTAAATTTTGTGTTATGAGCTTCACCACCGACTGGCGCTTTTCGCCG
GCGCGGAGTCGTGAACTCGTTGACGCCCTGATGGCGGCCCGTAAGGACGTGTGCTATCTG
GAGATTGATGCCCCTCAAGGCCACGATGCCTTTCTGATTCCGATCCCTCGCTACCTGCAGG
CCTTCTCACATTACATGAATCGTATTACCCTGTGAAAGCTT

OLG #7
>AmGl+nucleotidyltransferase_Pyrroline-5-carboxylate+reductase_7_4_0_839_CAT_S23_4
GAATTCATGTCTACAGAAAAAGAAATGCTTGACCTAATTGTTCAGCTAGCTCAGCAGGACGA
TGAGCTGCGCGCCCCAGTGCTGAACGGATCAGAGATTCAAAGCCATCTGCGGAAAAACTG
TTTTCAGGACTACGATCTGTTGTATCTGGTGAACGATATCGACGGCGTAACTCAGCAGCATC
AATGGATTCATCGATTTGGAGATGTGCTGCTCCTACAGCTGCCTGAAGAGTTAAGTCTTCTG
CCACCTGACGAGGAAGGCCAATTTCCATACCTGCTGCAGTTTCTGCCTGGATCAGAAGTAG
ATAATACGCTGGTCCCGATCCGCTTACTGCAGAAATTTGTTCCAGCTGATTCTCTTCATCAAG
CCATACTGCAACGCCAACACTGCTTAGAAGAGTTTCCGCCTGCTGGCGATCGCGACTATTTA
ATTCAGAAACCGCGTAAAGACCGCTTTCTGGATTGCGCAGAAGAATTTTGGTGGTGCTCAG
GCAACGTAGCACAGGGTCTTTGGCGAGCCAAGCTGTCATACGCGCGCACGTTACTCCAGC
CGCCAGTGAAACAATTATTTGTTCTGGTTTTAGAATGGCACGTAGGGCTTCGCACAGACTTT
TCAGTGCCTGCGGGGAAATTTGGAAAACAATTTGATCAATACCTTGAATCCGATCTGTGGAA
TCAGCAAAAACGTCAATTCCGTGATGCGCGCTATAACGCTGTTTGGTCTTCTTTCCTGGTTC
TGAGCGACGTATTTCGCTCGCTTGCAAACCGCGTGCGCGCAACCCACCAATATAGCTACCC
GCAAGATTCCCAAGACCAACAAACGCAATACGTGTCCCACGCATCAGCCCTCCCGAAAGAC
AGTACTTCCATTTATTAAAAGCTT

OLG #8
>Homoserine+O-succinyltransferase_AmGl+acetyltransferase_100_4_0_737_CAT_S13_5
GAATTCATGCCGACTGCCTTTCCTCCCGACAGCGTGGGCCTGGTGACCCCGCAGTTGGCG
CATTTTTCTGAGCCGCTGGCGCTGGCATGTGGTCGTAGCCTGCCAGCATATGATTTGATATA
TGAGACGTACGGCCAGCTGAATGCCAGCGCCTCGAATGCCGTTCTGATTTGCCATGCACTC
TCCGGTCACCACCATGCCGCTGGCTACCATAGTGTAGATGACCGTAAACCGGGGTGGTGG
GACAGTTGCATTGGACCGGGCAAACCGATTGATACTAACAAGTTCTTTATTGTGAGCCTGAA
TAACATCGGCGGTTGTAATGGATCAACCGGTCCGTCTGCTCAAAACCCAGAGACTGGTGAA
CCTTTTGGCTCAGATTTTCCTCTGGTGACGTATCAGAACTGGATTCACGGCAACCTTCGTTT
GGCCCCCAACGTTGGACTGCAGCAATGGTCTGCTGTGCTAGGCCCCTCTGTCGGTGGGAT
GGAAGCACTGCAATGGACTGTAACATATCCAGATCGCTTGCGTCACTGCCTTGCACGTAGT
CGTGCGCCAGAGCTGCAGGCGCAAAACCTAGCGCTAAACCATCTGCAACGGCAACAAATT
CTGACAGATGATCAGTTTGAAGGAGGTTCTTTCCGTCAGCAAGGTGTGCTTCCGAAACGTG
GCCTGGCCACAGCTCGCGTCGCAGGCCACGCCACCTATCTAAGTGACGACGATTTAGGCG
AGAAATTTCCATCAGGACTCAAATCGGAAAAACTGAATTATGATTTCCACTCAGTGGAGTTCC
AGGTAGAGTCTTATTTACGCTATCAGGGTGAAGAGTTTTCAGGTCGCTTTGACGCTAACACG
TACTTGCTGATGACAAAAGCCCTGGACTATTTTGACCCGGCCGCGAACTTTGATGATGATCT
CGCGAAAACGTTTGCGAACGCGTCCGCGAAATTTTGTGTGATGTCGTTCACGACCGACTGG
CGTTTTTCTCCGGCACGCAGCCGTGAATTGGTTGATGCCCTGATGGCAACGATGCCTTTCT
CATTCCTATTCCTCGTTATTTACAAGCCTTCAGTCACTATATGAATAGAATCACTCTGTAAAAG
CTT

OLG #9
>Isopropylmalate+dehydrogenase_AmGl+acetyltransferase_160_1_0_479_ATG_S13_6
GAATTCATGTCAAAACAGATCCTCATCCTGCCAGGAGACGGGATCGGGCCGGAAATTATGG
CTGAAGCCGTAAAAGTATTGGAACTTGCCAACGACAAATATTCCCTGGGCTTTGAACTTAGT

CATGATGTGATCGGCGGAGCGGCGATAGACAAACATGGCGTGCCACTGGCGGATGAAACT
TTAGATAGAGCGCGAGCCGCTGATGCGGTGCTGCTGGGCGCGGTGGGGGGTCCAAAATG
GGACAAAATCGAGCGTGATATCCGCCCTGAACGCGGTTTGCTGAAAATTCGGGCCCAGCTG
GGACTGTTTGGGAATCTGCGACCGGCCATTCTGTATCCGCAGCTGGCCGACGCCTCAAGC
TTAAAACCGGAGATTGTCTCGGGCCTGGATATTCTGATTGTGCGAGAATTAACCGGCGGAAT
TTATTTTGGCGCCCCGCGTGGCACACGAGAATTGGAGAACGGGGAACGCCAGAGTTACGA
ATGGATGCCGTACAGCGAAACCGAACTGAGGCGGATCATTGGCGTGGGCTTCGATCTCGC
GCGTGGCCGGGGCACCCAGATGATCAGCATCTCGCAGATGGAGATTCTGTCCTCCAGTCA
GATCTGGCGGCAAGTTTTATCGCAGTTAGCGAAGGATTGGCCGTCGGTTTCGCTGACGCAT
CTGTACGTCACGAATATGTCAATGGATGTGATTCGAGCCCCGAAATCTTTCGACGTGGTGTT
TACGTCAAACCTGTTTGGCGATATCGTGGCGTCGCGCGCCAGCTTATTGCTGGGGTCGATC
GGTATGGTACCAACCGCGGCTGTCGACGCCAACAATCAGGGACTGTACCAGAAAACACATG
GAGCCGCGCCGTCCATCGCGGGCTGGGGTTTAGCGAATCCGATCGCGTCATTTTTTACCGT
AAGCGTTGTGCTGCGTTATTCATTTAACCTCACTGAAGCTGCGGATGCCATCGAGCAGGCC
GTGAGCCGTGTTCTGGATCAGGGTCTGCGTACCGGCGATATCTGGAGCGCGGGGTGCACG
AAAGTCGGAACCCAGGAAATGGGCGATGCGGTGGTTGCGGCCTTGCGCAATCTGTGAAAG
CTT

Nucleotide sequences of the MGs (=longer sequence of the OLG pair) in the second set of 10 selected OLG sequences used in the growth experiments:

OLG #11
>O-succinylhomoserine+lyase_AmGI+acetyltransferase_46_4_0_575_TAT_S22_7
GAATTCATGACTCGTAAACAGGCTACCATCGCGGTTAGATCAGGTTTAAACGATGACGAGCA
GTATGGCTGTGTTGTCCCGCCTATTCACCTGAGTAGCACCTACAATTTTACAGGTTTTAACGA
ACCTAGAGCGCATAATTATAGCAGGCGCGGCGATAAAACACGAGACGTTCTGTCTCGTCAAA
TCGCAGAGCTTGATGGAGGCGCTGGCTCAGTGTTGACTCAGGCTGGGTATCAGGCGCTAC
ATCTTTACACCCTTTATTTTCTGACCAGCGGCGAACTGCTGCTAGCACCTCACGACTGCTAC
GGCGGCGCCTATAGGCTGGCAGAACAAATAGCCCTTCGAGGAATATATCAGGTGCGCTATCT
AGACCGTTCAGATAATCGTGCCTTACAGGCGCGTCTGCAAAACCAACCGCAACTGATTCTG
CTAGAATCACCAGCGAACCCAGTTCTTCGGCTTGCAGATATTCTTCGCCTGTGCCGATATGC
GCGTCATCTGGGTGCCCTGGCCATAATTGATAACGCATTCCTTTCCAATGCTCTGCAAAATC
CGCTTGCATTTGGCGCAGATCTATTACTACACTCCTGCACGAAATATCTGAACGGCCACTCT
GACGTGGTCGCTGGTGTTGTTATTGCAAAAGACCCCGAAATGGTTACCGAACTGGCCTGGT
GGGCAAACAATATCGGAGTAACGGGCGGAGCGTTTGATTCTTACCTGTTACTGCGAGGTTT
GCGCACCCTGGTACCACGGATGGAATTAGCACAGCGTAACGCACAAGCCATCGTTGATTAC
CTGCAGACACAGCCGCTGGTAAAGAAACTCTATCATCCGAGCTTGCCGGAAAACCAGGGC
CACGAAATCGCGGCCCGTCAACAGAAAGGTTTTGGAGCCATGCTGTCTTTCGAATTAGATG
GTGATGAGGAAACGCTTCGGCGCTTTCTGGGTGGTTTAAGCCTGTTCACTCTGGCGGAAAG
TCTGGGGGGTGTCGAATCCCTTATTTCACATGCCGCCACCATGACACATGCAGGCATGTCA
CCCCAGGCCCGCGCAGCGGCCGGGATCTCGGAGACTCTGCTTCGCATCAGTACTGGAATT
GAAGATGGAGAGGACCTCATTGCAGATCTGGAAAATGGATTTCGCGCAGCCAACAAAGGTT
AAAAGCTT

OLG #12

>Diaminopimelate+decarboxylase_Fosfomycin+thioltransferase_97_4_0_713_CAG_S13_5

GAATTCATGGACGCGTTTAACTACCGCGATGGTGAATTGTTTGCTGAGGGTGTTTCGCTTAC
TGCAATCGCTGAACGCTTTGGGACCCCGACGTATGTGTACTCTCGCGCACATATTGAAGCC
CAGTACAACGCCTACGCAGACGCATTAAGCGGAATGCCGCATCTCGTGTGTTTTGCAGTCA
AAGCAAATTCAAATCTGGGGGTGTTGAACGTGCTGGCCCGTTTGGGGGCTGGCTTTGATAT
CGTGTCCCGTGGGGAGCTGGAACGCGTGCTGGCGGCTAGGGGTTCAGCAGACCGTCAAA
TATTTTCAGGTGTTGGCGAGACTCGCGATGATATGAGACGCGGTCTTGAAGTTGGCCTGTAT
GCATTTCAAATTTATGGCCGTCCGGATCTGGAAAATATACAGATTGTTTGTCGCGAGATGGG
CGTTCGCGCCCCGATATCACTGAAACTGAATCCTGATTTAGACGCTGGTACACATCCGTATA
TTCAGACTGGTCTAAAAGAAAACAAATATGGGTATAGCTATGCTGAAGCTCGTGACGTGGAA
TATCGCGCTGCTGATTTACCGCAAGCCAAAGTCCTTGGAGTTGATTGTAGGCTGTCGAGCG
ACCTTTCAACAGCAGATCCGTTTCTAGATGCGATTGATAAAATGCTAGCGCTGATTCAACGTT
TGGGAGACTGTGGAGTATATGTGAGACACCTTGAATTGTCAGGAGGACTCGGTGTACGCTA
CAGGGATGAAGAACCGCCGCTAGCTGCAGATTACATCAAAGCGGTGCGTGAACGTATTGAG
GGCCGTGACTTAGCGTTAGTGTTCGAGCCTGGGCGTTTTATTGTGGCTAATGCGGGTGTATT
GTTGACCCAAGTGGAATATCTCAAGCATACAGAACATAAAGATTTCGCAATTGTGGATGCGG
CAATGAATGATCTGATTCGTCCGGCTTTATATCAGGCATGGATGGACGTTACCGCAGTTCGG
CCGCGCGATACCGAGGCTCGTGCCTATGATATCGTGGGCCCTATTTGCGAAACGGGCGATT
TTCTCGCGAAAGATCGTCAGTTGGCACTGGCCGAAGGTGACTTACTGGCAGTGCATTCTGC
AGGAGCGTATGGATTTGTAATGAGTAGTAACTATAACACGCGCGGTCGTGCTGCGGAAGTC
CTGGTAGACGGGGATCAGGCGTTTGAAGTGCGTCGGCGCGAAACGGTTGCCGAACTATTT
GCGGGTGAAAGTCTGCTGCCAGAATGAAAGCTT

OLG #13

>Diaminopimelate+decarboxylase_Fosfomycin+thioltransferase_47_4_0_563_AAT_S16_7

GAATTCATGGATGCGTTTAATTACAGGGATGGTGAGCTATTTGCGGAAGGCGTGTCTCTCAC
CGCAATTGCCGAACGTTTCGGTACGCCAACATATGTTTACTCCCGCGCCCATATTGAAGCAC
AATACTCTAGCTTTGCAGATTCGCTTCAAAAAATTCCATATGTGATTTGTTTTGCTTATAAAAGT
CCAAGCGATCTTGGAGTGTACCAGTATGTAGCTCGAATTGGTGCGGGTCTGGATCTGTATAG
TAAAGGCTCTTTGGATCGCGTTCTGGCCGCGGGCGGCCCTGCAGAACGTTTAGTTTTTTCT
GGTTTAGGCGAGACCAAAGACGATCAAACTCGCGCTCTTGAATTGGGAATGCATTGTGTGA
ATATTGATGCGACAGATCGTTACGAGGAACTTCAGATTCTTGCTGCAGAGCAAGCCATACGC
CGTCCAGTTCAAAATAGGCTGAATCCTGACCTTTCAGCAGGAACTCACCCGTATATATCGAC
TGGTAAAAAAGAATCGAAATTTCCAATTGCGATAGCTGATGCAGAAAATGTTTATATCCGCGC
AGCGCAATTACCAAACCTCGAAGTTGTCGGGGTGGATTGCCACATAGGCAGCCAGCTGACC
ACCCTGGATCCGTTCATAGATGCGCTCGACCGGCTGTTGGCCCTAATCGACAGACTGGGCG
ATTGTGGAATTTATCTCCGCCACATTGACCTGGGAGGTGGCTTAGGAGTCCGCTATCGCGAT
GAAGAGCCGCCGTTAGCTGCGGATTACATCAAAGCTGTCCGCGAGCGCATTGAGGGCCGT
GATCTGGCGCTGGTATTTGAGCCGGGGAGATTCATTGTTGCGAATGCGGGGGTGCTGTTAA
CGCAGGTGGAGTACCTGAAACATACCGAACATAAGGATTTTGCCATTGTCGACGCCGCTATG
AACGATTTGATTCGGCCGGCGCTCTATCAAGCGTGGATGGATGTTACCGCTGTGCGTCCGC
GGGATACCGAAGCTAGAGCCTACGACATCGTCGGGCCAATCTGCGAAACCGGAGATTTCCT
GGCCAAGGACCGACAGCTGGCACTGGCCGAAGGTGATTTATTAGCGGTTCACTCGGCAGG

CGCTTATGGCTTCGTCATGTCCAGTAATTATAATACCCGGGGTCGCGCAGCCGAAGTCCTTG
TAGACGGCGACCAGGCATTTGAAGTTAGACGGCGTGAGACAGTTGCTGAACTGTTCGCTG
GCGAGAGCTTACTGCCGGAGTGAAAGCTT

OLG #14
>Homoserine+O-succinyltransferase_Chloramphenicol+acetyltransferase_85_4_0_905_CAG_S
13_5
GAATTCATGCCTACAGCTTTTCCTCCGGATTCTGTAGGTTTGGTCACACCGCAGCTGGCTCA
TTTTTCGGAACCTCTGGCGCTGGCATGTGGTCGCTCTCTGCCGGCATACGATTTGATTTATG
AGACTTACGGTCAACTGAACGCATCTGCATCAAATGCGGTACTTATCTGTCATGCCCTGTCT
GGTCACCACCACGCCGCGGGCTACCATTCTGTAGACGACCGTAAACCGGGTTGGTGGTCT
AGCTGCCTGGGCCCAGGCGAAGCCATTGATTGCAATCGTTTTTTTGTTGTTGCAGTGAATCA
ATTAGGCGGCTGCAATGGTAGCCTTGGCAAATCGAGTGATGAACCTGAAACTGGACGTCCA
TTTGGAGCTGATTTCCCTGTGAAAACCATTTCCCAGTGGTTACAATCGGAAGCAGAAATTGC
TGATCGTTTGGGATATTCAGATTGGCGCCTGTATATTGGGTCCACGGTAGGCGGCTTACAGG
CATTACGTTGGGCGGTAAGTTATCCTGATAAAATACGCCATGCACTGGCGCGTAGCAGCGCA
GATCGTCTTCGTAGTCAGAATATAGCGTTCGAAAATCTTGCGCGTAATCTGTCCATAACTGAT
CCAGACTTTCATCGGGGCGATGAAAAAGAGTATGGCTTGGTACCATCTCGCGCCAATATCCT
AGCACGCCTTCTTGGTTATAGCACGTACGAAAGCGACGATGATTATTCAGAATCCTTTGGAT
CAGGTATAAAAACACAGGATAAAAATTACGATTTTTTTGATATACAGTTTCAAGCAGAATCGTA
TCTTCGATATCAAGGCGAAGAGAAAAGCGGCAGGTTTGACGCAAATACCTATCTGTATATGA
CTCGCGGTTTAGATCATTTCGATCCAGCACGTGAAATTTCAGACGACCTCGCCAAAACCTTC
GCGAACGCGTCGGCGAAATTCTGCGTGATGAGTTTCACCACCGATTGGCGGTTTTCTCCAG
CCCGCTCTCGCGAGCTCGTAGACGCGCTGATGGCCGCGCGAAAGGACGTTTGCTATCTGG
AGATTGATGCACCCCAAGGGCATGACGCGTTTCTGATTCCTATTCCGAGATATCTGCAGGCA
TTTTCGCATTACATGAACCGTATTACGCTGTGAAAGCTT

OLG #15
>Fusaricacid+resistance_Diaminopimelate+decarboxylase_156_4_0_1715_CAG_S24_10
GAATTCATGCCTATCACTTTTCAGGCATTATTCGCACCGAGTTCTTTGGCGTTGAAATTCGCG
ATTAAAACTCTGCTGGGCGGGGGGCTGGCCCTGTGGCTGGCAATGCGATGGGGCCTGGAA
CAACCCAGTTGGGCCCTGATGACAGCCTTTATTGTAGCGCAGCCGCTGAGCGGTATGGTGG
TCCAAAAAGGCCTGGCGCGGCTGGCTGGTACGCTGGTAGGAACGGTCATGAGCGTGTTAT
TTATCGGTCTGTTTGCCCAGACGCCGTGGCTGTTTCTTCTGACACTCGCCCTTTGGCTGGC
CCTCTGCACCGCAGCCTCGACCCAGCTGCGCAGCGCCTGGGCCTACGCGTTCGTTTTAGC
GGGATATACCGCGGCGATCATTGCACTTCCCGCGATTGATCACCCTCTACAGGTGTTTGACC
AGGCGGTCGCTAGATGCACCGAAATCTGCCTGGGAATTATATGTTCCACTGCGGCAGCAGC
GATTCTGTGGCCAGTACGCGTTGAACAGTCTCTCGGCGGCGAAGCTCGAAGGGCATGGCA
GAATCAACTGCAAGCTGCGCGGCGCGTCCTCGGCGGTTATAATTCTGCGAGAAAAGGAATC
CTTGAGAGCCTGGGCCGTGTAATTGCAGTAGATCGCCAGCGCGAACACGCATGGTTCGAG
GGCGGGCGAGGAAATCGCCGGTCTCGCACACTGGGCCAATTAAGTCGTAAGCTGATGCTT
CTGTATCGCGTGGCCGAATCGCTGAAACGTCAATGGCGGCAGTATACAGAGCTGGACGTAG
AACATCTGACACCGTGGCTTCAAGAAGTGCAAAGTCTTTATGATCAGTATGACGAAGATATTC
TGCTTCTGTTAAAAGAACGCCTGCATGACGCAGCACATGACGACCAGGTTCAAACAGCAGA

GCAATTTTGCGTCCTTCGATACGCTCTTTTACTGCCTTATAGTATGGCAGCGACAGCGGCGG
TTCCTGATCTCGATATCGGACGCCCAGACCGCCGCCTAAGTCAAGGTGTCGCAGCACATCG
CCAGTGGAGCCTAGCCTATCTATTTGGCTCAGCAAGCGGTCTAGCGCATCTAAAAATGGATC
AATTTTGGTTAGCTGCGAGCTGGCCGAGCAGTCCAGGCCAATTAATTCTAACTGCGGTAATT
TGCTCGCTTTTTGGTAAACGTCAGAACGGTCTGCAATTGGGACTCCAATTTTTGCGCGGCG
TATTCCTGTCGATACCAGCGGGTGTGTACCTAGGTCAATATCTGGTTCCACAGTGGAGCTCA
CTGGCGCTTTTATGCCTTGGTCTGGGGCTACCACTTGTAATCGGCGCAGTTGGTCTGTCGC
ATCCACGTTCAGGCAGTACAGCCCAATCTCAATGCCTTCACGCATATCTTCTCGTGTCTCCC
CTAAACCTGAAAAAATTTGGCGTTGCGCTGATCCTGAACTCAGCAGTAGCCGTTCTAATTGG
CCTTTCTGCAGCAGTTCAAAGCCTGCGCCTAATCGTGCTAAAACATCCAGCATGGCTAGGTT
CGCGTTTGCGCGCAGCGCAAAACAAACAATATGTGGCATTGACGCGACGCGATCTGCGTG
GCGCTGATAGCTGGTTTCGAGGTGCGCGCGCTGATAAATATATGCAGGTGTGCCGACACGC
TCAGCAACTGCCGTCAGGCGAACGCCTTCGCTGGTCAGATGGCCTTCACGGTATTGATATT
GGTCCAGAACTAGTACACCTCCGTATGTGCCTCGCAGTGGCTCAGGCGCCGCTGGGCCCG
GCTGAACGCGAGTACCTACAGCAGGTCGAAGCAGTGCTTGCGAAAGGCCCGGCGGCCGG
CCGTGGCCAGCGCCTGGACGCTGCGAGTGAACAATTCATCGCGGCACTGCGTAGGCTGCC
TGCAAGCGATCCGTTACGCCTCGCCGAGGGCGCTGTGCTTCAGCTTCAGAAATCATGGGG
TAAATGGTGCCGTTGGCAAGAAGACACCCATGGCTTCGCGTGAAAGCTT

OLG #16

>Serine+hydroxymethyltransferase_AmGl+acetyltransferase_68_4_0_641_CAC_S13_5
GAATTCATGCTTAAACGCGAAATGAATATCGCCGATTATGACGCTGAATTATGGCAGGCAATG
GAACAGGAGAAAGTTCGCCAAGAAGAACACATTGAACTGATCGCGTCAGAAAACTACACCT
CGCCACGTGTTATGCAAGCTCAAGGTAGCCAGCTCACTAATAAATATGCAGAAGGATATCCG
GGTAAATCTATATTTGGAGGTTGCCAATACGTTGACGATATAGAACAACTCGCTCTGTCTCGT
CAAAACCAACTGTTTGGTGCAGATTTTGCGAACGTGTATCCTCATTCTGGGTCTCAGGCGC
CATTTCGCGTGTACCTTTGTTTGTTGACCAGCGGCGATACGCTGCTAGGTATGAATCTGCAA
CAGGGCGGTCACGAAACGCAGGGATCACCAGTAAACCTTTCAGGAAAATTACTGGGCTTGA
TTCCGTACGGTTTAGATAATCGTGGCGAACTGAACTATCAGCAAATGGCACAGCAAGCCCGT
CAGCATAAGCCAAAAATGCTGCTAGGTGGTTTTTCTGCAGAATCTGGTCTCCTTGACTGGCG
TGAGCGTCGTCAGGATGCCCAGGCCATAGGCGCTTACGTATTCCTCGCCAATGCTCATATTG
CGTCTTTGATTGCTGCCGGATATCCACCAAACCCTGTGCCGCACGCACATGTCGTGACGAC
TACTACCCATAAAACACTGGCGGGGCCGCGTGGGGGCCTCATCCTAGCGAAGGGGGGAGA
CGAAGAACTGTATAAGAAATTAAATAGCGCGGTTTTCCCTAGTGCGCAGGGTGGCCCGCTG
ATGCACGTTATAGCGGCTAAAGCCGTGGCGCTGAAAGAAGCAATGGAGCCTGAATTTAAGG
TGTATCAACAGCAAGTGGCCAAAAATGCTAAAGCGATGGTGGAGGTGTTCCTGAATCGTGG
TTACAAAGTAGTGTCGGGAGGCACTGAGAACCATCTTTTTCTGTTAGACCTGGTTGACAAAA
ACTTAACCGGTAAAGAGGCCGATGCGGCATTAGGACGCGCTAATATCACTGTAAATAAGAAC
TCCGTGCCAAATGATCCCAAAAGTCCCTTTGTTACTAGCGGTATCAGGATAGGCTCGCCAGC
AGTTACCAGGCGCGGGTTCAAAGAAGCTGAGGTCAAAGAGTTGGCTGGCTGGATGTGCGA
CGTTCTGGATAATATTAACGATGAAGCCGTAATCGAACGCGTCAAAGGCAAAGTTTTAGACAT
TTGTGCGCGTTTTCCGGTTTACGCTTAAAAGCTT

OLG #17

>AmGl+nucleotidyltransferase_Pyrroline-5-carboxylate+reductase_5_4_0_835_AAT_S24_10
GAATTCCTATGCGTACAGAAAAAGAGATCCTAGACGTTCTGCTAAATCTGCTGAACGATGAC
GAGCGGCTCCGAGCGCTGATTCAATCAGGCTCTCGAATCCATCCGCCTGAAAAGAAAGAAT
GCTTTCAGGATTACGACCTGATGGTGATGTTAAACGACGTCGAATCTCTGACGCATCAACAT
CGCTGGTTACACCGATATGGGCAGCTCCAAGTGCTGTCTGTACCAGAAGAGATGTCGCTAA
TTCCGCCGGAAGACCCAGGCGAGTTCCCGCTGCTGTTACAGTTTCAATCAGGAGAAAAAAT
TGACCTGGCCCTGATCCCGATACAGCTGTTACAGCGTCTAATTGGTCAGGATCATCTAACCA
AACTGCTGCTGGATAAAGACTCAGCAATTCCTCAGTTTCCTGACGCTGCTGATAAAGAATAT
CTGATTCAGCGTACAACCCAGAAACACTTTCTCGAATGCTGCTCGGAGTATTGGTGGTGCA
GCGCACAACTGGCTGCTGGCCTAACCAGAGAAGAACTGAGCTACACGAAAGGCCTGCTGC
AAGGCCCAGTACGAGATCTTCTGGTTTTAATGCTGGAATGGCATATTGGCATGCAGACTTCA
TTTTCTGTGGAGACAGGCAAATTTGGTAAACATCTACGCCAGTATCTGCCTGAAGATTTGTG
GCAAATGTTTCAACGCCGTTTTCAGAACGCACCCTACGACGAGATTTGGCGCCAGGATCTG
ATGCTCGGAGATGTGATGCGCGAAGTCCTTTCGCGCGTAGCCCGCCAATATAGCTACCAGC
AACCGAACCAGCGCCAAGAAAAGCTAACCGGGTATCTGAAATACGTAAAAGCCCTCCCAAA
GGACTCGACCTCTATTTATTGAAAGCTT

OLG #18
>Threonine+synthase_Fosfomycin+thioltransferase_13_4_0_461_CAT_S13_5
GAATTCATGACCCATCAGTGGCGTGGTATTATTGAAGAATACCGAGATCGAATACCTGTAAGT
GATACGACTCCTGTTTATACCATTCGAGACGGTCGGACACCGTACCTGTATGCAGCTCAAAT
TTATGACCGTCTGGGTTGCGAAATTCATCTGAAAGTTTCTGGCGCGAATCCGACGGGCGGC
CTGAAAGAACGTTCACTGACAGTCGCGATAAGCGATCGTAAAGAAGAGGGAATCCGCGCTG
TTCTATGCCAAAGCACAGGTGGGTATAGCTCCTCTGCAGCTCGTTACGCGGTACGTCGCGG
TCTTGTTTCAGCGCTAACCATACCCCGTTCAGATCAAAGTATTGGGTCGATCGGCCAGGCAC
TAATACACGGCGCGAAAATACTTCAGATTGATGGGCAATTTGATGATTGCCTGACGCTGGCG
AAAACAGAAAGTGATTCATACCCTGTAGCGACATTAAACTCCGTTAACCCGGTCCGTATTGAA
GGCCAGAAGACCGCCGCCTTTGAGATTGTAGACATGCTCGGCGATGCCCCGGACATTCATG
TGCTGCCAGTTGGTAACGCCGGCAACATAACCGCCTATTGGAAGGGTTATAAAGAGTACGC
CGCAGACGGCATCGCGACGCGTACGCCGCGTATGTGGGGTTTCCAGGCATCCGGTAGCGC
GCCAATTGTCCGCGGCGAAGTGGTGAAAGATCCGAGTACTATCGCGACCGCTATTAGGATC
GGCAACCCCCGCATCATGGCAATACGCGCTGGCTGCGAGAGACGAGAGTGGGGGTGCTATT
GATGAAGTCACCGATCGTGAGATTCTTCGTGCCTACCGCCTGCTGGCGGCCCAGGAGGGA
GTTTTTGTTGAGCCAGCTAGTGCGGCGAGCGTGGCAGGTTTACTGAAAGCGGCAGAACAG
GGAAAGGTGGACCCGGGTCAGCGTATCGTTTGTACCGTGACGGGCAATGGGCTGAAAGAC
CCGGACTGGGCGGTTGCGGGCGCGCCTCAGCCCGTTACGGTTCCAGTAGATGCAGCTACA
GCAGCGGAACGTCTGGGTCTGGCCTAAAAGCTT

OLG #19
>Isopropylmalate+dehydrogenase_AmGl+acetyltransferase_160_1_0_479_ATG_S13_6
GAATTCATGTCAAAACAGATCCTCATCCTGCCAGGAGACGGGATCGGGCCGGAAATTATGG
CTGAAGCCGTAAAAGTATTGGAACTTGCCAACGACAAATATTCCCTGGGCTTTGAACTTAGT
CATGATGTGATCGGCGGAGCGGCGATAGACAAACATGGCGTGCCACTGGCGGATGAAACT
TTAGATAGAGCGCGAGCCGCTGATGCGGTGCTGCTGGGCGCGGTGGGGGGTCCAAAATG

GGACAAAATCGAGCGTGATATCCGCCCTGAACGCGGTTTGCTGAAAATTCGGGCCCAGCTG
GGACTGTTTGGGAATCTGCGACCGGCCATTCTGTATCCGCAGCTGGCCGACGCCTCCAGC
TTAAAACCGGAGATTGTCTCGGGCCTGGATATTCTGATTGTGCGAGAATTAACCGGCGGAAT
TTATTTTGGCGCCCCGCGTGGCACACGAGAATTGGAGAACGGGGAACGCCAGAGTTACGA
ATGGATGCCGTACAGCGAAACCGAACTGAGGCGGATCATTGGCGTGGGCTTCGATCTCGC
GCGTGGCCGGGGCACCCAGATGATCAGCATCTCGCAGATGGAGATTCTGTCCTCCAGTCA
GATCTGGCGGCAAGTTTTATCGCAGTTAGCGAAGGATTGGCCGTCGGTTTCGCTGACGCAT
CTGTACGTCACGAATATGTCAATGGATGTGATTCGAGCCCCGAAATCTTTCGACGTGGTGTT
TACGTCAAACCTGTTTGGCGATATCGTGGCGTCGCGCGCCAGCTTATTGCTGGGGTCGATC
GGTATGGTACCAACCGCGGCTGTCGACGCCAACAATCAGGGACTGTACCAGAAAACACATG
GAGCCGCGCCGTCCATCGCGGGCTGGGGTTTAGCGAATCCGATCGCGTCATTTTTTACCGT
AAGCGTTGTGCTGCGTTATTCATTTAACCTCACTGAAGCTGCGGATGCCATCGAGCAGGCC
GTGAGCCGTGTTCTGGATCAGGGTCTGCGTACCGGCGATATCTGGAGCGCGGGGTGCACG
AAAGTCGGAACCCAGGAAATGGGCGATGCGGTGGTTGCGGCCTTGCGCAATCTGTGAAAG
CTT

OLG #20
>Fusaricacid+resistance_Isopropylmalate+dehydrogenase_13_4_0_1121_CAT_S21_3
GAATTCATGCCCATCACCTTCCAGGCCCTTATAGCGCCGTCATCAGCAGCCCTGAAATTTGC
GCTAAAATCGCTTCTGGGAGGCGGTCTGTGCCTGTGGCTGTGCATCCGCTGGACCATAGAT
CGCCCGTCCTGGCGCCTGATGACAGCATTTCTGATAGCGCAGCCTCTATCAGGTCTGCTGC
TTCAGAAAGGTTTAGCGCGTATCGCAGGTACATTGATAGGGACAGTAATGTCGCTACTGGAT
TTGGGATTGTTTGCCCAGACACCGTGGCTGCTGATCCGTGGCATGGCTCTGTGGCTCCTTT
ATTGTTCAGCTGCAGCGACGCAGCTGGAATCAGCGTGGCGCTACCTGTTTGTACTGAGCG
GTTATCTTGCAGCAGTGATCCAAATACCGAGTCTGTCACATCCACTTCAAATTTTTTCGGAGC
GCGTAGCAGCTGCAACGCAGCGTTGTCTGGGTATTTTTTGCGCAACTGCACATCAGGGTAT
TCTTTGGCCACTTCGCGTAGAATCTCGCGCCACAGGTCAGACGAGGCAAGCATGGCAGAA
CGGTCTGCAGAGCACACGCGCTTACCTCGGAGGCGAGCAAGATCAAAACCGAGGCGTGCT
AGACGCCTTAGGTCGGATTCTGAGTATGGAAGCGTATCGTAAGCACGCCTGGTTCCGTGGT
CAACGCGGCGAACGCCGCGCGGAGGCCCTACGTAAACTGAGCCAGTCAATTCGCGTATTAT
TAAAATATCAACGCCAGATACGACGTCAGTGGCGACAGCTGACGCATCGCGAAGTTGAGGA
TATAACACCGTGGGTCGAAGATTACCAAACAGCCCTAGATCAGCCTGAATCTCCAGCAGTCC
TCGTTCTGGGCGAACGTCTTTTTCAAGCTGCTGCCAACGAGGACCTCCAATCGCGCCAAGA
ATTTTGCCTGGCGCGGCGCGCGCTTTATCTAGAGTTTCGTCTGCCAGCGGCACGCCGTGTT
TATCAAGTGCAGCTCGGCCGAGCACCTCGTGACTTAAGTCAAGGCCTAGCTGCACATCGTG
ATTGGTCAGTTGCAGTACTTTTTGGCTCTCGCTCAGCACTTGCGGTCCTAATCCTGAGCCC
GGTATGGTTAGCAACTGCTTGGCCATCAGCTCCTGGCGGTCTGATACTGACGTGCGTTGTC
TGTAGTCTGTTTGCGTCACGGGAGAACGGTGCGCAGATCGGTCTGAGCTTCCTTCGCGGC
ATCTTTCTGGCCGTTCCGGCTGCGTTTCTGGTTGGTCAGATCATCTTACCTCAGTGGTCATC
CTTCGCCATGCTGTGTCTCGGCATGGGGGTGCCTCTGTTCCTCGGCGCCCTAGGTATGGC
GCATCCACGTACAGGCGCCACAGCCACCTCATACTGCCTTCACTTTATTGTCCTGGTATCGC
CGCTAAATGCGATGCAATTCGGCGTAGCGACGATGCTTAACTCTGCATTGGCTATGCTCGTG
GGGGTGAGCGCGGCCGTTATGGCCTTTCGTCTGTTGGTTTTCAGGCATCCGGCCTGGTTG
GGACGGCGTCTGCGCGCTGCCACCCAGAACGATCTCGTAAGATTAACCCGTCGTGATCTTC

GGGGAGCGGACAGCTGGTTCGGGGGTCGCATGGCAGATCGCCTGATGCAATTGGCGCGG
CATGCGTCGGAACTTCCGGAGGGCGAACGTAAAAGATGGGATGATGGACTGCATGGGCTG
GATATCGGCGATGAATTAGTACATTTACGTATGTGCCTGGCTGTTGCACAGGCTCCGTTGGG
TCCGGCTGAACGCGAATATCTTCAACAAGTAGAAGCCGTCTTAGCGAAAGGTCCGGCAGCG
GGGCGCGGCCAGCGCCTGGATGCCGCGTCTGAGCAGTTTATTGCGGCCCTCCGCCGCCT
GCCGGCCTCGGATCCACTGCGTCTAGCGGAAGGTGCCGTGTTACAGTTGCAAAAAAGCTG
GGGGAAATGGTGCCGTTGGCAGGAGGACACCCACGGTTTTGCTTAAAAGCTT

Bibliography

1. Darwin, C. (1859). The Origin of Species; And, the Descent of Man. Modern library.
2. Bateson, W., & Mendel, G. (2013). Mendel's principles of heredity. Courier Corporation.
3. Gerstein, M. B., Bruce, C., Rozowsky, J. S., Zheng, D., Du, J., Korbel, J. O., ... & Snyder, M. (2007). What is a gene, post-ENCODE? History and updated definition. Genome research, 17(6), 669-681.
4. Palade, G. E. (1955). A small particulate component of the cytoplasm. The Journal of biophysical and biochemical cytology, 1(1), 59.
5. Sabatini, D. D., Tashiro, Y., & Palade, G. E. (1966). On the attachment of ribosomes to microsomal membranes. Journal of molecular biology, 19(2), 503-IN9.
6. MB, H., PC, Z., & ML, S. (1957). Intermediate reactions in protein biosynthesis. Biochimica et biophysica acta, 24(1), 215-216.
7. Hoagland, M. B., Stephenson, M. L., Scott, J. F., Hecht, L. I., & Zamecnik, P. C. (1958). A soluble ribonucleic acid intermediate in protein synthesis. Journal of Biological Chemistry, 231(1), 241-257.
8. Nirenberg, M. W., Jones, O. W., Leder, P., Clark, B. F. C., Sly, W. S., & Pestka, S. (1963, January). On the coding of genetic information. In Cold Spring Harbor Symposia on Quantitative Biology (Vol. 28, pp. 549-557). Cold Spring Harbor Laboratory Press.
9. Barrell, B. G., Air, G. M., & Hutchison, C. A. (1976). Overlapping genes in bacteriophage φX174. Nature, 264(5581), 34-41.
10. Crick, F. H. (1958, January). On protein synthesis. In Symp Soc Exp Biol (Vol. 12, No. 138-63, p. 8).
11. Morange, M. (2009). The Central Dogma of molecular biology. Resonance, 14(3), 236-247.
12. Bussard, A. E. (2005). A scientific revolution? The prion anomaly may challenge the central dogma of molecular biology. EMBO reports, 6(8), 691-694.
13. Knight, R. D., Freeland, S. J., & Landweber, L. F. (2001). Rewiring the keyboard: evolvability of the genetic code. Nature Reviews Genetics, 2(1), 49-58.
14. Ilardo, M., Meringer, M., Freeland, S., Rasulev, B., & Cleaves II, H. J. (2015). Extraordinarily adaptive properties of the genetically encoded amino acids. Scientific reports, 5, 9414.
15. Philip, G. K., & Freeland, S. J. (2011). Did evolution select a nonrandom “alphabet” of amino acids?. Astrobiology, 11(3), 235-240.
16. Pape, T., Wintermeyer, W., & Rodnina, M. (1999). Induced fit in initial selection and proofreading of aminoacyl-tRNA on the ribosome. The EMBO journal, 18(13), 3800-3807.
17. Koonin, E. V., & Novozhilov, A. S. (2009). Origin and evolution of the genetic code: the universal enigma. IUBMB life, 61(2), 99-111.
18. Agris, P. F., Vendeix, F. A., & Graham, W. D. (2007). tRNA’s wobble decoding of the genome: 40 years of modification. Journal of molecular biology, 366(1), 1-13.
19. Berg, J. M. (2001). Biochemistry. Chemical & Engineering News, 79(13), 130-130.
20. Goodenbour, J. M., & Pan, T. (2006). Diversity of tRNA genes in eukaryotes. Nucleic acids research, 34(21), 6137-6146.

21. Barbieri, M. (2015). Code biology: a new science of life. Springer.
22. Srinivasan, S., Torres, A. G., & Ribas de Pouplana, L. (2021). Inosine in biology and disease. Genes, 12(4), 600.
23. Parker, J. A. C. K. (1989). Errors and alternatives in reading the universal genetic code. Microbiological reviews, 53(3), 273.
24. Woese, C. R. (1965). On the evolution of the genetic code. Proceedings of the National Academy of Sciences of the United States of America, 54(6), 1546.
25. Woese, C. R., Dugre, D. H., Dugre, S. A., Kondo, M., & Saxinger, W. C. (1966, January). On the fundamental nature and evolution of the genetic code. In Cold Spring Harbor symposia on quantitative biology (Vol. 31, pp. 723-736). Cold Spring Harbor Laboratory Press.
26. Buhrman, H., van der Gulik, P. T., Klau, G. W., Schaffner, C., Speijer, D., & Stougie, L. (2013). A realistic model under which the genetic code is optimal. Journal of molecular evolution, 77(4), 170-184.
27. Haig, D., & Hurst, L. D. (1991). A quantitative measure of error minimization in the genetic code. Journal of molecular evolution, 33(5), 412-417.
28. Mathew, D. C., & Luthey-Schulten, Z. (2008). On the physical basis of the amino acid polar requirement. Journal of molecular evolution, 66(5), 519-528.
29. Woese, C. R., Dugre, D. H., Saxinger, W. C., & Dugre, S. A. (1966). The molecular basis for the genetic code. Proceedings of the National Academy of Sciences of the United States of America, 55(4), 966.
30. Henikoff, S., & Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. Proceedings of the National Academy of Sciences, 89(22), 10915-10919.
31. Freeland, S. J., Knight, R. D., Landweber, L. F., & Hurst, L. D. (2000). Early fixation of an optimal genetic code. Molecular biology and evolution, 17(4), 511-518.
32. Massey, S. E. (2008). A neutral origin for error minimization in the genetic code. Journal of molecular evolution, 67(5), 510.
33. Butler, T., Goldenfeld, N., Mathew, D., & Luthey-Schulten, Z. (2009). Extreme genetic code optimality from a molecular dynamics calculation of amino acid polar requirement. Physical Review E, 79(6), 060901.
34. Freeland, S. J., & Hurst, L. D. (1998). The genetic code is one in a million. Journal of molecular evolution, 47(3), 238-248.
35. Goldman, N. (1993). Further results on error minimization in the genetic code. Journal of molecular evolution, 37(6), 662-664.
36. 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. Biology direct, 2(1), 24.
37. Lynch, M., & Marinov, G. K. (2015). The bioenergetic costs of a gene. Proceedings of the National Academy of Sciences, 112(51), 15690-15695.
38. Itzkovitz, S., & Alon, U. (2007). The genetic code is nearly optimal for allowing additional information within protein-coding sequences. Genome research, 17(4), 405-412.
39. Mir, K., & Schober, S. (2014, August). Investigation of genetic code optimality for overlapping protein coding sequences. In 2014 8th International Symposium on Turbo Codes and Iterative Information Processing (ISTC) (pp. 152-156). IEEE.

40. Bartonek, L., Braun, D., & Zagrovic, B. (2020). Frameshifting preserves key physicochemical properties of proteins. Proceedings of the National Academy of Sciences, 117(11), 5907-5912.
41. Xu, H., & Zhang, J. (2021). On the origin of frameshift-robustness of the standard genetic code. Molecular biology and evolution, 38(10), 4301-4309.
42. Ade, P. A., Aghanim, N., Arnaud, M., Ashdown, M., Aumont, J., Baccigalupi, C., ... & Matarrese, S. (2016). Planck 2015 results-xiii. cosmological parameters. Astronomy & Astrophysics, 594, A13.
43. Axe, D. D. (2004). Estimating the prevalence of protein sequences adopting functional enzyme folds. Journal of molecular biology, 341(5), 1295-1315.
44. Tian, P., & Best, R. B. (2017). How many protein sequences fold to a given structure? A coevolutionary analysis. Biophysical journal, 113(8), 1719-1730.
45. Dalrymple, G. B. (2001). The age of the Earth in the twentieth century: a problem (mostly) solved. Geological Society, London, Special Publications, 190(1), 205-221.
46. Aghanim, N., Akrami, Y., Ashdown, M., Aumont, J., Baccigalupi, C., Ballardini, M., ... & Roudier, G. (2020). Planck 2018 results-VI. Cosmological parameters. Astronomy & Astrophysics, 641, A6.
47. Tripathi, S., & Deem, M. W. (2018). The standard genetic code facilitates exploration of the space of functional nucleotide sequences. Journal of molecular evolution, 86(6), 325-339.
48. Maeshiro, T., & Kimura, M. (1998). The role of robustness and changeability on the origin and evolution of genetic codes. Proceedings of the National Academy of Sciences, 95(9), 5088-5093.
49. Blalock, J. E. (1990). Complementarity of peptides specified by ‘sense’ and ‘antisense’ strands of DNA. Trends in biotechnology, 8, 140-144.
50. Zull, J. E., & Smith, S. K. (1990). Is genetic code redundancy related to retention of structural information in both DNA strands?. Trends in biochemical sciences, 15(7), 257-261.
51. Konecny, J., Eckert, M., Schöniger, M., & Hofacker, G. L. (1993). Neutral adaptation of the genetic code to double-strand coding. Journal of molecular evolution, 36(5), 407-416.
52. Crick, F. H., Brenner, S., Klug, A., & Pieczenik, G. (1976). A speculation on the origin of protein synthesis. Origins of life, 7(4), 389-397.
53. Kun, Á., & Radványi, Á. (2018). The evolution of the genetic code: Impasses and challenges. Biosystems, 164, 217-225.
54. Hinegardner, R. T., & Engelberg, J. (1963). Rationale for a universal genetic code. Science, 142(3595), 1083-1085.
55. Crick, F. H. (1968). The origin of the genetic code. Journal of molecular biology, 38(3), 367-379.
56. Woese, C. R., Hinegardner, R. T., & Engelberg, J. (1964). Universality in the genetic code. Science, 144(3621), 1030-1031.
57. Griffith, F. (1928). The significance of pneumococcal types. Epidemiology & Infection, 27(2), 113-159.

58. Aggarwal, N., Bandhu, A. V., & Sengupta, S. (2016). Finite population analysis of the effect of horizontal gene transfer on the origin of an universal and optimal genetic code. Physical biology, 13(3), 036007.
59. Knight, R. D., Freeland, S. J., & Landweber, L. F. (1999). Selection, history and chemistry: the three faces of the genetic code. Trends in biochemical sciences, 24(6), 241-247.
60. Barbieri, M. (2019). Evolution of the genetic code: The ambiguity-reduction theory. Biosystems, 185, 104024.
61. Woese, C. R., Olsen, G. J., Ibba, M., & Söll, D. (2000). Aminoacyl-tRNA synthetases, the genetic code, and the evolutionary process. Microbiology and Molecular Biology Reviews, 64(1), 202-236.
62. Seligmann, H. (2013). Pocketknife tRNA hypothesis: anticodons in mammal mitochondrial tRNA side-arm loops translate proteins?. Biosystems, 113(3), 165-176.
63. Yarus, M., Widmann, J. J., & Knight, R. (2009). RNA–amino acid binding: a stereochemical era for the genetic code. Journal of molecular evolution, 69(5), 406-429.
64. Higgs, P. G., & Pudritz, R. E. (2009). A thermodynamic basis for prebiotic amino acid synthesis and the nature of the first genetic code. Astrobiology, 9(5), 483-490.
65. Yarus, M. (2017). The genetic code and RNA-amino acid affinities. Life, 7(2), 13.
66. Miller, S. L. (1953). A production of amino acids under possible primitive earth conditions. Science, 117(3046), 528-529.
67. Kvenvolden, K., Lawless, J., Pering, K., Peterson, E., Flores, J., Ponnamperuma, C., ... & Moore, C. (1970). Evidence for extraterrestrial amino-acids and hydrocarbons in the Murchison meteorite. Nature, 228(5275), 923-926.
68. Lawless, J. G., Kvenvolden, K. A., Peterson, E., Ponnamperuma, C., & Moore, C. (1971). Amino acids indigenous to the Murray meteorite. Science, 173(3997), 626-627.
69. Wong, J. T. F. (1975). A co-evolution theory of the genetic code. Proceedings of the National Academy of Sciences of the United States of America, 72(5), 1909.
70. Santos, J., & Monteagudo, Á. (2011). Simulated evolution applied to study the genetic code optimality using a model of codon reassignments. BMC bioinformatics, 12(1), 1-8.
71. Buhrman, H., van der Gulik, P. T., Kelk, S. M., Koolen, W. M., & Stougie, L. (2011). Some mathematical refinements concerning error minimization in the genetic code. IEEE/ACM transactions on computational biology and bioinformatics, 8(5), 1358-1372.
72. Massey, S. E. (2006). A sequential "2-1-3" model of genetic code evolution that explains codon constraints. Journal of molecular evolution, 62(6), 809.
73. Mir, K., Neuhaus, K., Scherer, S., Bossert, M., & Schober, S. (2012). Predicting statistical properties of open reading frames in bacterial genomes. PLoS One, 7(9), e45103.
74. Pavesi, A., Magiorkinis, G., & Karlin, D. G. (2013). Viral proteins originated de novo by overprinting can be identified by codon usage: application to the “gene nursery” of Deltaretroviruses. PLoS Comput Biol, 9(8), e1003162.
75. Sieber, P., Platzer, M., & Schuster, S. (2018). The definition of open reading frame revisited. Trends in Genetics, 34(3), 167-170.
76. Fukuda, Y., Nakayama, Y., & Tomita, M. (2003). On dynamics of overlapping genes in bacterial genomes. Gene, 323, 181-187.

77. Merino, E., Balbás, P., Puente, J. L., & Bolívar, F. (1994). Antisense overlapping open reading frames in genes from bacteria to humans. Nucleic Acids Research, 22(10), 1903-1908.
78. Kreitmeier, M., Ardern, Z., Abele, M., Ludwig, C., Scherer, S., & Neuhaus, K. (2021). Shadow ORFs illuminated: long overlapping genes in Pseudomonas aeruginosa are translated and under purifying selection. Available at SSRN 3866842.
79. Fiddes, J. C., & Godson, G. N. (1978). Nucleotide sequence of the J gene and surrounding untranslated regions of phage G4 DNA: Comparison with phage φX174. Cell, 15(3), 1045-1053.
80. Cassan, E., Arigon-Chifolleau, A. M., Mesnard, J. M., Gross, A., & Gascuel, O. (2016). Concomitant emergence of the antisense protein gene of HIV-1 and of the pandemic. Proceedings of the National Academy of Sciences, 113(41), 11537-11542.
81. Dinan, A. M., Lukhovitskaya, N. I., Olendraite, I., & Firth, A. E. (2020). A case for a negative-strand coding sequence in a group of positive-sense RNA viruses. Virus evolution, 6(1), veaa007.
82. Nelson, C. W., Ardern, Z., Goldberg, T. L., Meng, C., Kuo, C. H., Ludwig, C., ... & Wei, X. (2020). A previously uncharacterized gene in SARS-CoV-2 illuminates the functional dynamics and evolutionary origins of the COVID-19 pandemic. bioRxiv.
83. Schlub, T. E., & Holmes, E. C. (2020). Properties and abundance of overlapping genes in viruses. Virus evolution, 6(1), veaa009.
84. Normark, S., Bergström, S., Edlund, T., Grundström, T., Jaurin, B., Lindberg, F. P., & Olsson, O. (1983). Overlapping genes. Annual review of genetics, 17(1), 499-525.
85. Scherbakov, D. V., & Garber, M. B. (2000). Overlapping genes in bacterial and phage genomes. Molecular Biology, 34(4), 485-495.
86. Fukuda, Y., Tomita, M., & Washio, T. (1999). Comparative study of overlapping genes in the genomes of Mycoplasma genitalium and Mycoplasma pneumoniae. Nucleic acids research, 27(8), 1847-1853.
87. Tunca, S., Barreiro, C., Coque, J. J. R., & Martín, J. F. (2009). Two overlapping antiparallel genes encoding the iron regulator DmdR1 and the Adm proteins control sidephore and antibiotic biosynthesis in Streptomyces coelicolor A3 (2). The FEBS journal, 276(17), 4814-4827.
88. Kim, W., Silby, M. W., Purvine, S. O., Nicoll, J. S., Hixson, K. K., Monroe, M., ... & Levy, S. B. (2009). Proteomic detection of non-annotated protein-coding genes in Pseudomonas fluorescens Pf0-1. PloS one, 4(12), e8455.
89. Hücker, S. M., Vanderhaeghen, S., Abellan-Schneyder, I., Wecko, R., Simon, S., Scherer, S., & Neuhaus, K. (2018). A novel short L-arginine responsive protein-coding gene (laoB) antiparallel overlapping to a CadC-like transcriptional regulator in Escherichia coli O157: H7 Sakai originated by overprinting. BMC evolutionary biology, 18(1), 1-14.
90. Hücker, S. M., Vanderhaeghen, S., Abellan-Schneyder, I., Scherer, S., & Neuhaus, K. (2018). The novel anaerobiosis-responsive overlapping gene ano is overlapping antisense to the annotated gene ECs2385 of Escherichia coli O157: H7 Sakai. Frontiers in microbiology, 9, 931.

91. Watanabe, S., Iimori, M., Chan, D. V., Hara, E., Kitao, H., & Maehara, Y. (2018). MDC1 methylation mediated by lysine methyltransferases EHMT1 and EHMT2 regulates active ATM accumulation flanking DNA damage sites. Scientific reports, 8(1), 1-10.
92. Spencer, C. A., Gietz, R. D., & Hodgetts, R. B. (1986). Overlapping transcription units in the dopa decarboxylase region of Drosophila. Nature, 322(6076), 279-281.
93. Iwabe, N., & Miyata, T. (2001). Overlapping genes in parasitic protist Giardia lamblia. Gene, 280(1-2), 163-167.
94. Williams, T., & Fried, M. (1986). A mouse locus at which transcription from both DNA strands produces mRNAs complementary at their 3′ ends. Nature, 322(6076), 275-279.
95. Makalowska, I., Lin, C. F., & Makalowski, W. (2005). Overlapping genes in vertebrate genomes. Computational biology and chemistry, 29(1), 1-12.
96. Pham, Y., Li, L., Kim, A., Erdogan, O., Weinreb, V., Butterfoss, G. L., ... & Carter Jr, C. W. (2007). A minimal TrpRS catalytic domain supports sense/antisense ancestry of class I and II aminoacyl-tRNA synthetases. Molecular cell, 25(6), 851-862.
97. Rodin, S. N., & Ohno, S. (1995). Two types of aminoacyl-tRNA synthetases could be originally encoded by complementary strands of the same nucleic acid. Origins of Life and Evolution of the Biosphere, 25(6), 565-589.
98. Martinez-Rodriguez, L., Erdogan, O., Jimenez-Rodriguez, M., Gonzalez-Rivera, K., Williams, T., Li, L., ... & Kuhlman, B. (2015). Functional class I and II amino acid-activating enzymes can be coded by opposite strands of the same gene. Journal of Biological Chemistry, 290(32), 19710-19725.
99. National Center for Biotechnology Information. (2019, January 7). NCBI Prokaryotic Genome Annotation Standards. Retrieved from: https://www.ncbi.nlm.nih.gov/genome/annotation_prok/standards/
100. Price, M. N., Wetmore, K. M., Deutschbauer, A. M., & Arkin, A. P. (2016). A comparison of the costs and benefits of bacterial gene expression. PloS one, 11(10), e0164314.
101. Sakharkar, K. R., Sakharkar, M. K., Verma, C., & Chow, V. T. (2005). Comparative study of overlapping genes in bacteria, with special reference to Rickettsia prowazekii and Rickettsia conorii. International journal of systematic and evolutionary microbiology, 55(3), 1205-1209.
102. Belshaw, R., Pybus, O. G., & Rambaut, A. (2007). The evolution of genome compression and genomic novelty in RNA viruses. Genome research, 17(10), 1496-1504.
103. Brandes, N., & Linial, M. (2016). Gene overlapping and size constraints in the viral world. Biology direct, 11(1), 1-15.
104. Warren, A. S., Archuleta, J., Feng, W. C., & Setubal, J. C. (2010). Missing genes in the annotation of prokaryotic genomes. BMC bioinformatics, 11(1), 1-12.
105. Lèbre, S., & Gascuel, O. (2017). The combinatorics of overlapping genes. Journal of theoretical biology, 415, 90-101.
106. Wichmann, S., & Ardern, Z. (2019). Optimality in the standard genetic code is robust with respect to comparison code sets. Biosystems, 185, 104023.
107. Sáenz-Lahoya, S., Bitarte, N., García, B., Burgui, S., Vergara-Irigaray, M., Valle, J., ... & Lasa, I. (2019). Noncontiguous operon is a genetic organization for coordinating bacterial gene expression. Proceedings of the National Academy of Sciences, 116(5), 1733-1738.

108. Thomason, M. K., & Storz, G. (2010). Bacterial antisense RNAs: how many are there, and what are they doing?. Annual review of genetics, 44, 167-188.
109. Georg, J., & Hess, W. R. (2011). cis-antisense RNA, another level of gene regulation in bacteria. Microbiology and Molecular Biology Reviews, 75(2), 286-300.
110. Štambuk, N., Konjevoda, P., Turčić, P., Kövér, K., Kujundžić, R. N., Manojlović, Z., & Gabričević, M. (2018). Genetic coding algorithm for sense and antisense peptide interactions. BioSystems, 164, 199-216.
111. Keese, P. K., & Gibbs, A. (1992). Origins of genes: "big bang" or continuous creation?. Proceedings of the National Academy of Sciences, 89(20), 9489-9493.
112. Willis, S., & Masel, J. (2018). Gene birth contributes to structural disorder encoded by overlapping genes. Genetics, 210(1), 303-313.
113. Sabath, N., Wagner, A., & Karlin, D. (2012). Evolution of viral proteins originated de novo by overprinting. Molecular biology and evolution, 29(12), 3767–3780.
114. Carter Jr, C. W. (2021). Simultaneous codon usage, the origin of the proteome, and the emergence of de-novo proteins. Current Opinion in Structural Biology, 68, 142-148.
115. Yockey, H. P. (1979). Do overlapping genes violate molecular biology and the theory of evolution?. Journal of Theoretical Biology, 80(1), 21-26.
116. Collins, D. W., & Jukes, T. H. (1994). Rates of transition and transversion in coding sequences since the human-rodent divergence. Genomics, 20(3), 386-396.
117. Kumar, S. (1996). Patterns of nucleotide substitution in mitochondrial protein coding genes of vertebrates. Genetics, 143(1), 537-548.
118. Moriyama, E. N., & Powell, J. R. (1997). Synonymous substitution rates in Drosophila: mitochondrial versus nuclear genes. Journal of Molecular Evolution, 45(4), 378-391.
119. Morton, B. R. (1995). Neighboring base composition and transversion/transition bias in a comparison of rice and maize chloroplast noncoding regions. Proceedings of the National Academy of Sciences, 92(21), 9717-9721.
120. Friedman, S. M., & Weinstein, I. B. (1964). Lack of fidelity in the translation of synthetic polyribonucleotides. Proceedings of the National Academy of Sciences of the United States of America, 52(4), 988.
121. Freeland, S. J., & Hurst, L. D. (1998). Load minimization of the genetic code: history does not explain the pattern. Proceedings of the Royal Society of London. Series B: Biological Sciences, 265(1410), 2111-2119.
122. Mushegian, A. (2008). Gene content of LUCA, the last universal common ancestor. Front Biosci, 13(4657), 66.
123. Grantham, R. (1974). Amino acid difference formula to help explain protein evolution. Science, 185(4154), 862-864.
124. Dayhoff, M. O., Schwartz, R. M., & Orcutt, B. C. (1972). A model of evolutionary change in proteins, in``Atlas of Protein Sequence and Structure''(MO Dayhoff, Ed.).
125. Bennet, S. A., Cohen, M. A., & Gonnet, G. H. (1994). Amino acid substitution during functionally constrained divergent evolution of protein sequences. Protein Engineering, Design and Selection, 7(11), 1323-1332.
126. Freeland, S. J. (2002). The Darwinian genetic code: an adaptation for adapting?. Genetic Programming and Evolvable Machines, 3(2), 113-127.

127. Taylor, F. J. R., & Coates, D. (1989). The code within the codons. Biosystems, 22(3), 177-187.
128. Fisher, R.A., 1930. The Genetical Theory of Natural Selection. Oxford University Press, Oxford.
129. Zhu, W., & Freeland, S. (2006). The standard genetic code enhances adaptive evolution of proteins. Journal of theoretical biology, 239(1), 63-70.
130. Jiménez, J. I., Xulvi-Brunet, R., Campbell, G. W., Turk-MacLeod, R., & Chen, I. A. (2013). Comprehensive experimental fitness landscape and evolutionary network for small RNA. Proceedings of the National Academy of Sciences, 110(37), 14984-14989.
131. Ferrada, E., & Wagner, A. (2012). A comparison of genotype-phenotype maps for RNA and proteins. Biophysical journal, 102(8), 1916-1925.
132. Wang, B., Papamichail, D., Mueller, S., & Skiena, S. (2005, June). Two proteins for the price of one: The design of maximally compressed coding sequences. In International Workshop on DNA-Based Computers (pp. 387-398). Springer, Berlin, Heidelberg.
133. Inouye, M., Ishida, Y., & Inouye, K. (2017). Designing of a single gene encoding four functional proteins. Journal of theoretical biology, 419, 266-268.
134. Opuu, V., Silvert, M., & Simonson, T. (2017). Computational design of fully overlapping coding schemes for protein pairs and triplets. Scientific reports, 7(1), 1-10.
135. Kryshtafovych, A., Schwede, T., Topf, M., Fidelis, K., & Moult, J. (2019). Critical assessment of methods of protein structure prediction (CASP)—Round XIII. Proteins: Structure, Function, and Bioinformatics, 87(12), 1011-1020.
136. Kryshtafovych, A., Schwede, T., Topf, M., Fidelis, K., & Moult, J. (2021). Critical assessment of methods of protein structure prediction (CASP)—round XIV. Proteins: Structure, Function, and Bioinformatics, 89(12), 1607-1617.
137. Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., ... & Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. Nature, 596(7873), 583-589.
138. Blazejewski, T., Ho, H. I., & Wang, H. H. (2019). Synthetic sequence entanglement augments stability and containment of genetic information in cells. Science, 365(6453), 595-598.
139. Wichmann, S., Scherer, S., & Ardern, Z. (2021). Biological factors in the synthetic construction of overlapping genes. BMC genomics, 22(1), 1-17.
140. Fernandes, J. D., Faust, T. B., Strauli, N. B., Smith, C., Crosby, D. C., Nakamura, R. L., ... & Frankel, A. D. (2016). Functional segregation of overlapping genes in HIV. Cell, 167(7), 1762-1773.
141. Safari, M., Jayaraman, B., Yang, S., Smith, C., Fernandes, J. D., & Frankel, A. D. (2021). Functional and Structural Segregation of Overlapping Helices in HIV-1. bioRxiv.
142. Sonnhammer, E. L., Eddy, S. R., & Durbin, R. (1997). Pfam: a comprehensive database of protein domain families based on seed alignments. Proteins: Structure, Function, and Bioinformatics, 28(3), 405-420.
143. Durbin, R., Eddy, S. R., Krogh, A., & Mitchison, G. (1998). Biological sequence analysis: probabilistic models of proteins and nucleic acids. Cambridge university press.
144. Eddy, S. R. (1998). Profile hidden Markov models. Bioinformatics (Oxford, England), 14(9), 755-763.

145. Katoh, K., & Standley, D. M. (2013). MAFFT multiple sequence alignment software version 7: improvements in performance and usability. Molecular biology and evolution, 30(4), 772-780.
146. Edgar, R. C. (2004). MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic acids research, 32(5), 1792-1797.
147. Rost, B. (1999). Twilight zone of protein sequence alignments. Protein engineering, 12(2), 85-94.
148. Torrisi, M., Kaleel, M., & Pollastri, G. (2019). Deeper profiles and cascaded recurrent and convolutional neural networks for state-of-the-art protein secondary structure prediction. Scientific reports, 9(1), 1-12.
149. Kabsch, W., & Sander, C. (1983). Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers: Original Research on Biomolecules, 22(12), 2577-2637.
150. Joosten, R. P., Te Beek, T. A., Krieger, E., Hekkelman, M. L., Hooft, R. W., Schneider, R., ... & Vriend, G. (2010). A series of PDB related databases for everyday needs. Nucleic acids research, 39(suppl_1), D411-D419.
151. Pauling, L., Corey, R. B., & Branson, H. R. (1951). The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain. Proceedings of the National Academy of Sciences, 37(4), 205-211.
152. Tian, P., Louis, J. M., Baber, J. L., Aniana, A., & Best, R. B. (2018). Co-Evolutionary Fitness Landscapes for Sequence Design. Angewandte Chemie International Edition, 57(20), 5674-5678.
153. Zhao, F., Yu, C. H., & Liu, Y. (2017). Codon usage regulates protein structure and function by affecting translation elongation speed in Drosophila cells. Nucleic acids research, 45(14), 8484-8492.
154. Miyata, T., & Yasunaga, T. (1978). Evolution of overlapping genes. Nature, 272(5653), 532-535.
155. Decrulle, A. L., Frénoy, A., Meiller-Legrand, T. A., Bernheim, A., Lotton, C., Gutierrez, A., & Lindner, A. B. (2021). Engineering gene overlaps to sustain genetic constructs in vivo. PLoS computational biology, 17(10), e1009475.
156. Benenson, Y. (2012). Biomolecular computing systems: principles, progress and potential. Nature Reviews Genetics, 13(7), 455-468.
157. Lapique, N., & Benenson, Y. (2018). Genetic programs can be compressed and autonomously decompressed in live cells. Nature nanotechnology, 13(4), 309-315.
158. Woller, A (2020). Artificial Overlapping Genes: Possibilities and Boundaries. [Unpublished master's thesis]. TUM School of Life Sciences, Technische Universität München.
159. Steinegger, M., & Söding, J. (2018). Clustering huge protein sequence sets in linear time. Nature communications, 9(1), 1-8.
160. Gudyś, A., & Deorowicz, S. (2017). QuickProbs 2: towards rapid construction of high-quality alignments of large protein families. Scientific reports, 7(1), 1-12.
161. Chen, D., & Texada, D. E. (2006). Low-usage codons and rare codons of Escherichia coli. Gene Ther. Mol. Biol, 10, 1-12.

162. Hyatt, D., Chen, G. L., LoCascio, P. F., Land, M. L., Larimer, F. W., & Hauser, L. J. (2010). Prodigal: prokaryotic gene recognition and translation initiation site identification. BMC bioinformatics, 11(1), 119.
163. Hayashi, K., Morooka, N., Yamamoto, Y., Fujita, K., Isono, K., Choi, S., ... & Horiuchi, T. (2006). Highly accurate genome sequences of Escherichia coli K-12 strains MG1655 and W3110. Molecular systems biology, 2(1), 2006-0007.
164. Baba, T., Ara, T., Hasegawa, M., Takai, Y., Okumura, Y., Baba, M., ... & Mori, H. (2006). Construction of Escherichia coli K-12 in-frame, single-gene knockout mutants: the Keio collection. Molecular systems biology, 2(1), 2006-0008.
165. Shenhav, L., & Zeevi, D. (2020). Resource conservation manifests in the genetic code. Science, 370(6517), 683-687.
166. Xu, H., & Zhang, J. (2021). Is the genetic code optimized for resource conservation?. Molecular biology and evolution, 38(11), 5122-5126.
167. Rozhoňová, H., & Payne, J. L. (2021). Little evidence the standard genetic code is optimized for resource conservation. Molecular Biology and Evolution, 38(11), 5127-5133.

Abbreviations

OLG - overlapping genes
DNA - deoxyribonucleic acid
RNA - ribonucleic acid
tRNA - transfer ribonucleic acid
mRNA - messenger ribonucleic acid
SGC - standard genetic code
A - adenosine
C- cytosine
G - guanosine
T - thymidine
U - uridine
AA - amino acid
MG - mother gene
ORF - open reading frame
HMM - Hidden Markov model
RBS - ribosome binding site

List of Figures

List of Tables

Acknowledgements

First and foremost I want to thank my doctoral supervisor Prof. Dr. Siegfried Scherer for his trust in giving me, as a physicist, the chance to graduate at the department of microbial ecology. Thank you very much for always having an open ear and your support in scientific and private matters.
Special thanks to my mentor Dr. Zachary Ardern for the many technical, scientific and philosophical discussions, which helped me to grow professionally and personally. Your help in proofreading was always greatly appreciated.
Furthermore I want to thank all my colleagues at the OLG working group, namely Dr. Barbara Zehentner, Dr. Michaela Kreitmeier, Dr. Alina Glaub, Dr. Sonja Vanderhaeghen, Franziska Graph, Anika Wahl and Dr. Christopher Huptas for the friendly and supportive work environment. I am also grateful to all co-workers in the department for all the good time we spent on breaks and excursions.
Lastly I want to thank my parents and brothers who were always ready to help me out. Special thanks goes to my wife Su, who always gave me honest feedback and supported me all the way to the end in good and bad times.