An evaluation of codes more compact than the natural genetic code

Royal Truman

Various researchers have claimed that the genetic code is highly optimized according to various criteria. It is known to minimize the deleterious effects of base-pair mutations and translational mistakes, and is said to be optimally compact. However, this view is based on the assumption that the quaternary code requires three nucleotides per amino acid, which the Kraft Inequality and McMillan's Theorem show is not correct. We demonstrate the existence of theoretical codes up to 23% more compact, but these would have been very error-prone. This analysis demonstrates the implausibility of evolving a simpler doublet code into the natural genetic code.

What would the characteristics of an ideal genetic code be? In his latest book,[1] Dr Werner Gitt suggests considerations like space, material and energy efficiency, plus robustness.[2] Furthermore, it must also incorporate its own error-correction features. Specifically,

1. storage in a living cell must be done within the smallest possible space. The choice of code should be one that uses the least materials. As the letter length (L) per word increases, the required material and storage space would increase
2. furthermore, as the number of characters (n) in the alphabet increases, the complexity of the execution machinery will also increase. This would require more material and result in more errors during replication, transcription and translation
3. because during DNA replication the double helix is unzipped and each of the single strands receives 'complementary' letters, the number of different letters in the alphabet must be even
4. In order to reduce errors during the many copying processes, it is necessary to incorporate redundancy.[2]

Gitt examined various possible block codes, based on alphabets of two to six symbols and codeword[3] ('cw') lengths between two and six. Based on the four principles above and the fact that twenty amino acids must be coded for, it was argued that

> "From an engineering point of view, and under the criteria that were considered here, the code system used in living organisms for protein synthesis—the Quaternary Triplet Code—is the best of all possible codes considering the four requirements that must be met."[4]

It seems worthwhile to look more closely into this claim given that both the Kraft Inequality and McMillan's Theorem (discussed below) demonstrate that point 1 (above) is not met. Specifically, coding for 20 amino acids and a stop instruction can actually be designed using shorter codewords.

In an oft-cited paper, Freeland and Hurst[5] examined a million codes semirandomly generated by a computer program and found only one claimed to be superior to the natural genetic code. This conclusion took into account the similarity (polarity or hydrophobicity) of the amino acids produced following a base-pair mutation, and that transitions tend to occur more frequently than transversions for mutations and mistranslations.[6] This paper was discussed in this journal,[7] where I concluded that the one in a million estimate is too high! Not all the reasons for this conclusion were mentioned. Here are two more. Firstly, Freeland and Hurst limited their analysis to alternative codes possessing the same high level of redundancy shown in the natural code[8] (table 1), which is highly improbable.

Why should one accept as a reasonable given multiple different aminoacyl-tRNA synthetase enzymes for most amino acids? Statistically more likely would have been many copies of the same synthetase to charge only one amino acid. These copies would mutate afterward in the three anticodon positions to produce many synonymous codons for just one or very few amino acids.

Secondly, an assignment of three codons to stop is also a good but not random choice. Too many such codons would accidentally terminate translation following random mutational or translational errors.[9]

Table 1. Pattern of codon redundancy in the standard genetic code. For example, amino acids Ala, Gly, Pro, Thr and Val are each represented by four codons: GCA, GCC, GCG, GCU.

Nr. of Synonymous codons	Coding for amino acid
1	Met, Trp
2	Cys, Asp, Glu, Phe, His, Lys, Asn, Gln, Tyr
3	Ile, Stop
4	Ala, Gly, Pro, Thr, Val
6	Leu, Arg, Ser

Interestingly, Freeland and Hurst took for granted that three characters are necessary for a quaternary code.

> "The length of codons in the genetic code is also optimal, as three is the minimal nucleotide combination that can encode the twenty standard amino acids."[10]

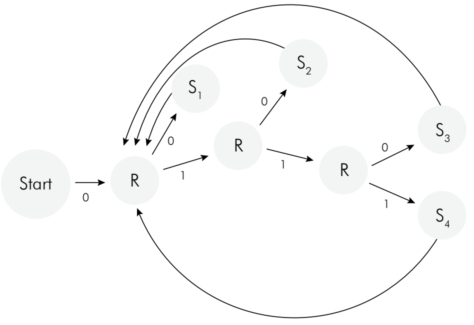
So we see that most have concluded with too little reflection that a shorter genetic code cannot be devised.

Instantaneous Codes

We will show that perfectly legitimate, uniquely decodable schemes can be developed which also satisfy the desirable feature of being *instantaneously decodable codes (IDCs)*. Otherwise code characters would need to be buffered and analyzed upon being decoded later. An example of a valid but non-IDC is shown in figure 1. Here the alphabet is based on only two characters (*0*,*1*) which are combined to form four codewords which represent unambiguously four symbols we shall call $s_1 \ldots s_4$ (these codewords could represent, for example, North, South, East, West).

The logic tree shown in figure 1 illustrates how the decoder often cannot know upon processing a letter (*0* or *1*) whether a full codeword has been received and must look ahead one, or sometimes two, characters. For instance, if *0* then *1* was received, perhaps codeword s_2 has been sent. But to know, the next letter must be checked. A *0* would confirm this. If instead a *1* is read, the decoder would now have to continue looking ahead to decide between s_3 and s_4. This is inefficient; s_2 has been received but the fact is not immediately known.

Performing this look-ahead logic is easy enough with electronic equipment, but would be very difficult for cells, which would have to manufacture the mechanical components for the processing equipment.

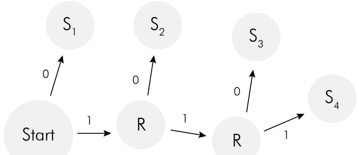


R: instruction to read the next symbol

Symbol	Codeword
S_1	0
S_2	01
S_3	011
S_4	0111

Figure 1. Logic tree for an example of a non-instantaneously decodable code based on four codewords, $s_{1.} \ldots s_4$.

Symbol	Codeword
S_1	0
S_2	10
S_3	110
S_4	1110



R: instruction to read the next symbol

Figure 2. Logic tree of an instantaneously decodable code. Each symbol possible $s_1 \ldots s_4$ is knowable upon reading the last codeword's character.

Therefore, we will demand that a more compact genetic code also be instantaneously decodable[11] as is the natural genetic code.[12] An IDC is *unique* only if all codewords can be interpreted without having to buffer future symbols. An example of such a code is shown in figure 2. The codeword is known as soon as its last character is read. No look-ahead is necessary (nor look-back, like other codes require). After each codeword is processed, the logic tree is then reset to read the rest of the message.

A shorter genetic code must exist

We will now examine genetic code candidates using IDC using codewords of different lengths. This implies that no codeword must contain another one embedded as a prefi.

The Kraft Inequality[13] expresses a necessary and sufficient condition for instantaneous codes to exist, based on wordcode lengths ranging from l_1 to l_{max}:

$$K = \sum_{i=1}^{q} r^{-l_i} \le 1 \qquad (1)$$

where q is the number of different codewords, r is the number of characters used in the alphabet, and l_i is the of length of each codeword. To illustrate, for the standard genetic code,[14] which has $l_i = 3$, we find using (1):

$$K = \sum_{i=1}^{21} 4^{-l_i} = 21x4^{-3} = 21/64 \le 1 \qquad (2)$$

Therefore, an IDC for a quaternary alphabet using fixed codeword lengths of three must exist, which indeed is the case (the natural genetic code).

Remarkably McMillan's Theorem[15] implies that for every non-instantaneous uniquely decodable code an *instantaneous* code with the same code word length can always be found.

Forewarned by these mathematical tools, can we find a shorter genetic code which is unique and instantly decodable? Let us try using C codewords of length $l = 2$ and the remaining with $l = 3$. Such a code must be shorter than the natural genetic code, which uses only $l = 3$. The Kraft Inequality and McMillan's Theorem demand that:

$$K = C \text{ x } (4^{-2}) + (21 - C) \text{ x } 4^{-3} \le 1 \qquad (3)$$

One solution is C = 14. In other words, a quaternary IDC for 20 amino acids plus a stop codon can be produced using 14 codewords two characters long and the remaining three characters long, leading to an average length of

$$((14 \times 2) + (7 \times 3))/21 = 2.3 \text{ letters per codeword} \qquad (4)$$

Once such a code is found (see below) the best strategy would be to assign the most frequently used amino acids to the shorter codewords, thereby leading to an average of less than 2.3 nucleotides for most genes. This additional compression is not available to the natural genetic code since all codewords have $l = 3$.

Construction of a shorter instantaneous genetic code

Instead of using code characters (*A,C,G,T*)[16] for the nucleotides of the genetic code, we will use numbers (*0,1,2,3*) for convenience, as is usual in coding theory. This reinforces the generality of our approach from a coding point of view; we make no assumptions about the chemical implementations. The principle is to begin with the shortest codewords and ensure none appear as a prefix in the longer ones. But to obtain enough codewords for the intended requirements, one may need to avoid some of the shorter patterns.[17,18] A more compressed genetic code ('Compact Code') is shown in table 2. Other coding conventions could have been chosen[19] which won't affect the conclusions we'll reach in this paper. Figure 3 helps identify it as an IDC, with every codeword being unique.

The Compact Code is *2.3*/3 or 23.3% more compact than the natural genetic code and would require fewer adaptor molecules (tRNAs), DNA and mRNA. The amount of resources saved would be dramatic.

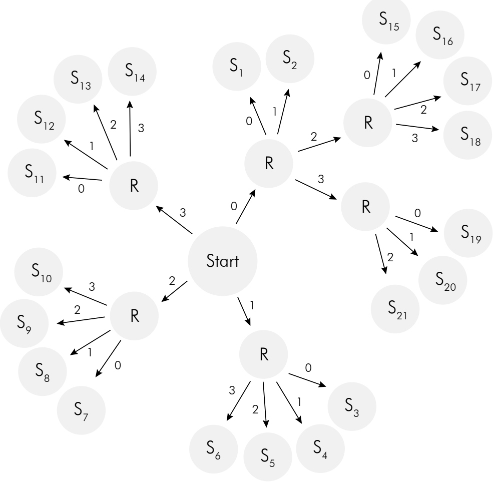


Figure 3. Logic tree for a compressed zero-memory genetic code ('Compact Code') shown in table 2, using two- and three-letter codewords. R: instruction to read the next symbol.

This prompts us to re-examine the claim made by Gitt *et al.*:

> "From an engineering point of view, and under the criteria that were considered here, the code system used in living organisms for protein synthesis—the Quaternary Triplet Code—is the best of all possible codes considering the four requirements that must be meet."[20]

It is actually not the shortest code possible. But what about the remaining criteria?

Robustness of a compressed genetic code

Robustness to mutations on DNA or errors in transcription and translation is an important criterion in addition to code compactness. Single- or double-base-pair ('bp') insertions or deletions ('indels') result in erroneous codewords for both the compact and natural genetic code.[21]

Far more common than indels are single bp mutations. It would seem that a code based on $4^2 = 16$ two-character codons plus 64 – 16 three-character codons should be able to provide many synonyms and thus neutral effects to single bp mutations. The opposite turns out to be the case.

Table 2. Compressed zero memory genetic code ('Compact Code'), using two- and three-letter codewords. Characters 0 … 3 represent nucleotides **A,C,T,G** and symbols s_1 .. s_{21}, 20 amino acids and a stop instruction. Average codeword length: (14 x 2 + 7 x 3)/21 = **2.33**.

Symbol	Codeword
S_1	00
S_2	01
S_3	10
S_4	11
S_5	12
S_6	13
S_7	20
S_8	21
S_9	22
S_{10}	23
S_{11}	30
S_{12}	31
S_{13}	32
S_{14}	33
S_{15}	020
S_{16}	021
S_{17}	022
S_{18}	023
S_{19}	030
S_{20}	031
S_{21}	032

Mutations in DNA

Of the potential $4^2 = 16$ two-character codons, only 14 can be used in a more compact code (like the one shown in table 2), and the two which don't, *02* and *03*, would be prefixes to existing codewords (*020*, *021*, *022*, *023*, *030*, *031,* and *032*). Therefore, no additional redundancy can be offered for the shorter codewords. Replacing any character cannot fail but code for a different amino acid.

How many (larger) codons could be added? The only pattern still available which will not include a prefix codeword is *033*. To put it to maximum use we would assign it as a synonym for one of the symbols s_{18}–s_{21} (table 2). However, this offers insignificant benefits and for only one codeword in the whole coding scheme: of nine possible bp mutations only one, 033, would be neutral.[22]

We conclude that the Compact Code permits almost zero protection to single, double or triple bp mutations (except for one

Table 3. Synonymous codons in the standard genetic code resulting from a single base-pair (bp) mutation. For each of three positions, a bp mutation can lead to any of three variations (e.g. A → C | G | T). Total possibilities are thus 3 x 3 x 64 = 576, of which 138 code for the same amino acid. 138 / 576 = 24% protection against single bp mutations.

AA	Codon	Synon. Codons[1)]
Ter	TAA	2
Ter	TAG	1
Ter	TGA	1
Ala	GCT	3
Ala	GCC	3
Ala	GCA	3
Ala	GCG	3
Cys	TGT	1
Cys	TGC	1
Asp	GAT	1
Asp	GAC	1
Glu	GAA	1
Glu	GAG	1
Phe	TTT	1
Phe	TTC	1
Gly	GGT	3
Gly	GGC	3
Gly	GGA	3
Gly	GGG	3
His	CAT	1
His	CAC	1
Ile	ATT	2
Ile	ATC	2
Ile	ATA	2
Lys	AAA	1
Lys	AAG	1
Leu	TTA	2
Leu	TTG	2
Leu	CTT	3
Leu	CTC	3
Leu	CTA	4
Leu	CTG	4
(Continued)		
Met	ATG	0
Asn	AAT	1
Asn	AAC	1
Pro	CCT	3
Pro	CCC	3
Pro	CCA	3
Pro	CCG	3
Gln	CAA	1
Gln	CAG	1
Arg	CGT	3
Arg	CGC	3
Arg	CGA	4
Arg	CGG	4
Arg	AGA	2
Arg	AGG	2
Ser	TCT	3
Ser	TCC	3
Ser	TCA	3
Ser	TCG	3
Ser	AGT	1
Ser	AGC	1
Thr	ACT	3
Thr	ACC	3
Thr	ACA	3
Thr	ACG	3
Val	GTT	3
Val	GTC	3
Val	GTA	3
Val	GTG	3
Trp	TGG	0
Tyr	TAT	1
Tyr	TAC	1

Synonymous mutations: **138**

negligible codeword). In contrast, for the standard genetic code 24% of single bp mutations lead to a synonymous codon (see table 3).

But an important factor must not be overlooked. In many cases a mutation which introduces a different amino acid in a protein can be tolerated. However, mutations in the longer codewords of a compressed genetic coding scheme can match the two first positions of a two-character codeword, leading to a reading-frame error. This would generally be devastating and lead to a worthless gene, since from that point on in the sequence the wrong amino acids would be linked together. Table 4 shows that this occurs in 5/9 of all bp mutations on the longer codewords for the compressed code shown in table 2.

The net effect is that about 5/9 x 7/21 = 18.5% of all bp mutations of the compressed code would be deadly.

Table 4. Single base-pair mutations in the longer codewords of a compressed genetic code will often produce two-letter codons (highlighted). This creates a reading-frame shift error: the third position is misinterpreted as being the first for the next codeword. Here 0 … 3 represent nucleotides such as **A,C,T,G** or a genetic system based on other chemistries.

Symbol	Code-word	Frameshifts created [a)]	Position mutated	New codon (9 possibilities)
s_{15}	020	5	1	**12**0, **22**0, **32**0,
			2	**00**0, **01**0, 030,
			3	021, 022, 023
s_{16}	021	5	1	**12**1, **22**1, **32**1,
			2	**00**1, **01**1, 031,
			3	020, 022, 023
s_{17}	022	5	1	**12**2, **22**2, **32**2,
			2	**00**2, **01**2, 032,
			3	020, 021, 023
s_{18}	023	5	1	**12**3, **22**3, **32**3,
			2	**00**3, **01**3, 033,
			3	020, 021, 022
s_{19}	030	5	1	**13**0, **23**0, **33**0,
			2	**00**0, **01**0, 020,
			3	031, 032, 033
s_{20}	031	5	1	**13**1, **23**1, **33**1,
			2	**00**1, **01**1, 021,
			3	030, 032, 033
s_{21}	032	5	1	**13**2, **23**2, **33**2,
			2	**00**2, **01**2, 022,
			3	030, 031, 033

[a)] Number of single-base-pair mutations which would produce an existing shorter codeword.

This consideration must be added to the complete lack of protection against mutations discussed above which do not change the codeword length (e.g. 00 → 01; or 031 → 032).

Translational errors

The vulnerability towards mutations in DNA across generations is one difficulty compressed genetic codes face. But transcription and translation errors also occur, allowing the first two letters of the longer codewords to often be misinterpreted, leading to frame shift reading errors.

We conclude that a compressed genetic code would indeed permit shorter messages to be used but at an inacceptable loss of reliability. And all other variations of compressed codes also result in inacceptable errors rates (Appendix). In the on-line Addendum, we point out the formidable, perhaps insurmountable, difficulties of implementing such a scheme biologically.[23]

Evolution from a quaternary codeword of length two to three

It is known that the third position of codons provides little discriminatory information about the intended amino acid, and that often the same tRNA can recognize multiple synonymous codons. One popular theory[24] speculates that a doublet genetic code may have preceded the current triplet one. Given that a doublet quaternary code could specify 16 events (16 amino acids, or 15 plus a stop codon), the reasoning is that it would have been easier for evolution to have produced a simpler organism requiring only 15/20 as many amino acids.

One merit of this proposal from an evolutionary point of view is the decreased number of possible codes which would result. Here is why this matters. Before a code could arise on its own some kind of chemical interaction would have to exist between amino acids and portions of DNA or RNA (although no unique, direct interactions are known between specific amino acids and specific codons, which is why tRNA adaptor molecules are needed). But suppose some weak interactions did exist in a precursor system. We pointed out earlier[25] that there are *1.5 x 10*^{84} ways to assign 20 amino acids plus a stop instruction to 64 codons.[26] Whatever changes are needed to make the decoding unambiguous and precise enough to be reliable would be faced with a hopeless number of unguided directions to explore. But to evolve a 15-amino-acid system using a doublet quaternary code would face a smaller space of possibilities (1.6 x 10^{14} different codes).[27]

There are different proposals as to which 15 or 16 amino acids could have been used in the initial doublet code and different arguments which attempt to justify believing such a code existed.[28] How persuasive is the argument that the third codon position serves little purpose? Examination of the anticodon region of tRNAs and their conjugate codons on mRNA suggests the relevant hydrogen bonds may often not be ideally lined up. After all, the chemical interaction must be only temporary and easily broken to allow the next codon to be processed by ribosomes in the cell. The first position is most important, since the triplet reading-frame must be firmly established. Therefore, the weakest interactions are expected to be in the third codon position, thereby being most susceptible to misinterpretation and requiring the greatest protection by allowing synonymous codons to differ at that position.

Let us re-examine the first two positions of the natural genetic code, table 5. Only seven of the 20 amino acids can actually be associated unambiguously to a doublet. There are five cases in which the doublet cannot distinguish between two amino acids, and four cases of different doublets coding for the same amino acid. Evidence for a preceding doublet code would have been considerably more persuasive had we observed that 15 amino acids and a stop instruction in the natural genetic code could be coded for by the first two positions instead of only so few. The justification to propose a preceding doublet code is not persuasive, on statistical grounds, considering that in the natural genetic code there are three different ways to place doublets within the three positions of the codons, increasing the chances of discovering something interesting merely by chance.[29] The reason simpler genetic systems have been proposed is prompted not by observed data but the discomfort that natural processes could

Table 5. Left: Importance of the first two positions of the natural genetic code to identify amino acids. Only 7 of the 20 amino acids are unambiguously assigned to a single codon. **Right**: Amino acid coding communicated by first two position on codons.

Implied by first two codon positions	Amino acid coded for
AC	Thr
CC	Pro
GC	Ala
GG	Gly
GT	Val
TA	Tyr
TT	Phe
AA	Lys or Asn
AT	Met or Ile
CA	Gln or His
GA	Asp or Glu
TG	Trp or Cys
CG or AG	Arg
TA or TG	Ter
TC or AG	Ser
TT or CT	Leu

AA	Codon	
Ter	TAA	TA or TG
Ter	TAG	
Ter	TGA	
Ala	GCT	GC
Ala	GCC	
Ala	GCA	
Ala	GCG	
Cys	TGT	TG
Cys	TGC	
Asp	GAT	GA
Asp	GAC	
Glu	GAA	GA
Glu	GAG	
Phe	TTT	TT
Phe	TTC	
Gly	GGT	GG
Gly	GGC	
Gly	GGA	
Gly	GGG	
His	CAT	CA
His	CAC	
Ile	ATT	AT
Ile	ATC	
Ile	ATA	
Lys	AAA	AA
Lys	AAG	
Leu	TTA	TT,CT
Leu	TTG	
Leu	CTT	
Leu	CTC	
Leu	CTA	
Leu	CTG	
Met	ATG	AT
Asn	AAT	AA
Asn	AAC	
Pro	CCT	CC
Pro	CCC	
Pro	CCA	
Pro	CCG	
Gln	CAA	CA
Gln	CAG	
Arg	CGT	CG, AG
Arg	CGC	
Arg	CGA	
Arg	CGG	
Arg	AGA	
Arg	AGG	
Ser	TCT	TC, AG
Ser	TCC	
Ser	TCA	
Ser	TCG	
Ser	AGT	
Ser	AGC	
Thr	ACT	AC
Thr	ACC	
Thr	ACA	
Thr	ACG	
Val	GTT	GT
Val	GTC	
Val	GTA	
Val	GTG	
Trp	TGG	TG
Tyr	TAT	TA
Tyr	TAC	

have initiated life based on something as complex as the natural genetic system.

Error protection in an evolving doublet genetic code

If 16 doublet codewords are used to code for different amino acids and a stop instruction, then there would be no protection against mutations: every mutation or translation error must lead to a different amino acid in the protein. Coding for fewer amino acids to provide redundancy offers minimal protection against single mutations. For example, codewords *10* and *11* could be used to code for the same amino acid. Then of 16 codewords, two would have a 50:50 chance that the mutation occurred on the second letter, itself with a 1/3 chance of producing a synonym given there are three mutational alternatives. Only 2 x 1/16 x 1/2 x 1/3 = 2% of the mutations would be harmless, at the price of losing the ability to code for an amino acid. Even devoting two synonyms for every codeword would lead to harmless single bp mutations only 17% of the time, and then only by being able to code for just seven amino acids and a stop instruction.[30]

The evolutionary compressed doublet scenario

Suppose a doublet code had existed. The message to code a protein would look something like: *12 21 01 32 01 33 13 10 21* … but without spaces as separators.

To evolve to the natural genetic code (table 5), at some point each doublet must no longer be a cw. This means the original tRNA would no longer recognize it. The problem is that a new evolving tRNA, which is supposed to read three nucleotides, would now 'steal' the first one from the next doublet, ruining the reading-frame and coding for a random pattern of amino acids thereafter.

Somehow a nucleotide would need to be inserted right after the evolving doublet codon and this particular codon would need to be identified by a new tRNA.

Unrealistic as this demand is, the expanding code would also be susceptible to a high proportion of deadly nonsense and reading-frameshift causing mutations (table 4), which increase as more triplet codons get produced. If a doublet cw like *33* evolved to a triplet *330*, then a 'vacancy', *33*, would be created by the discontinued doublet. Doublets such as *03*, *13*, *23*, *30*, *31* or *32* could mutate to *33* in one mutation, producing a nonsense, or undecodable doublet. In addition, the *third* position of triplet *330* could mutate to *331*, *332* or *333* with no meaning as double or triplet cws. Worse, the new *1*, *2,* or *3* in the third position could become a legitimate starting points for double cw, so we'd have another frameshift opportunity.

Of course, the *first* and *second* positions of *330* could also mutate, leading to *030, 130, 230, 300, 310, 320*, for which the first two positions would now be interpreted as valid doublets and a reading-frame would occur here also. It is clear that a compressed doublet could not have evolved into the natural triplet genetic code.

The evolutionary doublet scenario with a spacer

Given the difficulties of expanding from a simple doublet to a triplet quaternary code, an *ad hoc* assumption is found in the literature: although only the two first positions are used by the code, a third position must always have been there, serving as a kind of spacer.[24,31] This prepared the way for evolution (although it lacks foresight!) to develop the triplet code millions of years later.

Wong's coevolution hypothesis[32] proposes that a genetic code began with only seven amino acids: Glu, Asp, Val, Ser, Phe, Ala, Gly, and Stop. The proposed amino acids are based on known biosynthetic pathways, which leads to amino acids derived from other ones: Tyr from Phe; Cys and Trp from Ser; Lys, Thr, and Asn from Asp; Arg, Pro, and Gln from Glu; His from Gln; Met and Ile from Thr; and Leu from Val.[32,33] Note that the initial list has no basic amino acids and would not be able to form proteins as we know them!

Attention is drawn to the fact that there are small variations in the natural code sometimes in mitochondria and some single-cell organisms.[14] This means that a given codon in the natural code sometimes leads to a different amino acid in the variant code. Some therefore reason that if the code is not static it could evolve, perhaps dramatically. The most popular mechanism revolves around a codon capture notion:

> "This theory proposes a temporary disappearance of an amino acid codon (or stop codon) from coding frames by conversion to another synonymous codon and a loss of the corresponding tRNA that translates the codon. This produces an unassigned codon. For a stop codon, the release factor must simultaneously change so as not to recognize the stop codon. The codon reappears later by conversion of another codon and emergence of a tRNA that translates the reappeared codon with a different assignment. As a result, the nucleotide sequences change while the amino acid sequences of proteins do not change."[34]

In many cases the same tRNA molecule can identify synonymous codes. To illustrate, in the natural code both the U and C base at the codon's 3rd position can pair with a G in the anticodon; and A and G at the codon's 3rd position can pair with U in the anticodon.[35] Therefore, although there are 64 codons in the natural code, it is not necessary to have a unique tRNA for each codon. The number of different tRNAs present varies according to organism between 22 and 55.[36,37] However, tRNAs with the same anticodons often differ in other regions nearby, implying their use is sophisticated and regulated and not fully understood.

> "Unexpectedly, the number of tRNA genes having the same anticodon but different sequences elsewhere in the tRNA body (defined here as tRNA

isodecoder genes) varies significantly (10–246). "tRNA isodecoder genes allow up to *274* different tRNA species to be produced from *446 genes* in humans, but only up to 51 from 275 genes in the budding yeast."[36]

In theory, the unassigned codon capture could occur by different mechanisms: (i) by a change in the anticodon; (ii) by aminoacylating a different amino acid to an existing tRNA molecule; or (iii) in mitochondria, by a change in codon-anticodon pairing.[38] It is argued that sometimes there is a strong tendency towards forming a high proportion of G+C *vs.* A+T bases in the genomes of some organisms[39] which could lead to some synonymous codons disappearing (the genes coding for the tRNAs would be affected and also various codons.). G+C in eubacteria can vary between 25–75%.[39] Sometimes highly questionable assumptions must be made to explain how codons appear and disappear from a genome, such as a species first heading towards a high G+C proportion over time and later switching in the opposite direction[40]. Such conceptual models are lacking in evidence and purposely guide natural selection to make the concept seem feasible.

Others disagree with Wong's suggested genetic code.[33] Jukes proposed[24,41] an earlier code consisted of 15 amino acids and one Stop, each 16 outcomes being represented by four codons pairing with a single tRNA molecule. The nature of the third position would be irrelevant for coding purposes. Using numbers 0...3 to represent a base *A, C, G, U* for generality, the codewords would be 00, 01, 02, 03, 10, 11, 12, 13, 20, 21, 22, 23, 30, 31, 32, 33 plus any of the four nucleosides in the third position as a spacer to add stability to the codon-anticodon interactions. When the vertebrate mitochondrial code was discovered later, it was found that there were indeed such tRNA molecules.

This form of translation, where an unmodified U in the first position of the anticodon pairs with all four bases, *A, C, G*, and *U* in the third position of codons (four-way wobble) also takes place in *Mycoplasma* spp., and in two family boxes of the chloroplast code.[42]

Other evolutionary models deny the feasibility of a doublet genetic code and argue that a predecessor codeword was *larger* than the triplet code[43–46] and that this code shrunk over time. These proposals introduce insurmountable difficulties we won't elaborate on here.[47] Nevertheless, the notion of an earlier doublet genetic block code three nucleotides long is known and widely believed. But is it reasonable?

Jukes' model, or some variation of it, seems to be the most promising of the doublet proposals, from an evolutionary worldview. But it overlooks something fundamental. The third position on the codon can be any of the four bases, *A, C, G* or *U* and presumably adds stability to the interaction with the anticodon region. The problem is, there is therefore *no need for the third position, as a spacer at all. The first position of the next doublet would automatically serve just as well, since the third position is allowed to be any of the four bases*. The reading-frame would always be a codeword two nucleotides long although the codon-anticodon would involve stable triplet interactions. Since presumably 16 unique codewords were sufficient and evolution can't plan ahead, there would have been no reason for an extra non-informative spacer base!

Organisms evolving without the unnecessary 'spacer' nucleotide would save about a third of the energy and building materials, and the genome replication time would be much shorter. They would have an unbeatable competitive advantage and quickly out-reproduce the inefficient version proposed,[48] literally in a matter of days.[49]

Nucleotides outside the codeword reading-frame of the natural genetic code *can* interact to provide additional stability. For example, the nucleotide following the stop codon is important for efficient translational termination and the stop signal may be considered a tetranucleotide. U is by far the most highly represented of the four bases in the nucleotide position following all three stop codons in *E. coli*, whereas A and C are less frequent.[50,51]

A scheme like Jukes' offers zero protection to mutations in the 16 codewords: every mutation would generate a different amino acid or stop, and the number of mutations would be expected to be much higher than today (discussed below). Each of the 16 tRNAs are assumed to be able to interact with all four bases following the doublet codewords. For chemical reasons these four kinds of interaction will not be identically strong. In some cases the two-letter codewords will hardly benefit from the extra stability provided through the third base, so a +1 shift in the position of translation should often produce a comparably strong iteration using the three new positions. Consider, for example, an arbitrary codon *123* followed by *X* (*X = A, C, G, or U*). If the interaction of base *3* with the anticodon is weak but strong with *X*, then a frameshift to *23X* is likely to occur.[52]

How realistic would the subsequent evolutionary expansion to add new amino acids be? Several things must occur. A codon must disappear from the genome and then reappear only after a new tRNA, with a new amino acid attached to it, evolved (an untranslatable codon wouldn't be acceptable). The codon reappearance must be preceded or accompanied by two new independent metabolic processes: to create the new amino acid and a new synthetase enzyme. The eliminated codons would reappear, but must do so in a location which does not destroy an existing protein, and they must not be able to interact with the tRNA they had before. Note that only one amino acid at only one position in the whole genome would now be produced at this huge metabolic cost. Whatever the reason for disappearance of the codon must be reversed, for example, extreme high or low G+C content. Multiple copies of that codon must still be created throughout the genome, in each case offering the potential of disrupting existing proteins. The more non-informative codons are allowed in the third positions following the codewords (Jukes' model assumes all four bases), the greater the difficulty of evolving a corresponding new anticodon which will interact preferentially as it should and neglect the others.

No one has actually demonstrated that autonomously living organisms could survive using proteins based on a drastically smaller number of amino acids. The ferrodoxin are sometimes used as an example[53] of proteins which don't require all amino acids. Lys, Tyr, His, Met, and Trp are missing or very rare for them, but ferrodoxins are considerably smaller than the average protein,[54] are only useful together with other far more complex proteins and require a complex multiple iron-sulphur catalytic center.

Mutating genetic codes very error-prone

There are various scenarios one could envision to convert a doublet code to the natural code. In one evolutionary variant, a streamlined doublet (without spacers) would lose a codon like *33*; bp insertions and back-mutations presumably occurred later and the *33* then reappeared as a triplet.

Another variant would be that a doublet code existed with spacers. One codon became dependent on a particular spacer due to improvements in the original tRNA, so that at some point *33* would no longer be recognized, but a codon like *330* would.

In other words, *331*, *332,* and *332* would have disappeared temporarily from the genome, or at least be only weakly recognized, and so error-prone as to be practically undecodable. (Those codons could then reappear later and be used to represent new amino acids).

There are good reasons for arguing that were a doublet code with spacers to evolve towards the natural code, 'gaps' or ambiguity (like *331* and *332*) would arise. For example, amino acids *Methionine* and *Tryptophan* permit only one of the four bases to appear in the codon's third position, and neither amino acid is found in Wong's model. Codons with a new meaning would have been part of a set of synonyms which must first lose one or more members from the genome. This is because it is more plausible that a new amino acid could be tolerated in only one position than that all cases of an existing amino acid be replaced in one grand sweep throughout all genes in the cell.

Consider Tryptophan, which is represented in the natural code by only UGG. Cystein shares the same first two positions using codons UGC and UGU, and Cystein is not part of Wong's list of initial amino acids. The fourth codon sharing the first two positions, UGA, represents a stop instruction in the natural code. A new stop codon, spread all over a genome, would be deadly. We conclude that Tryptophan could not have arisen from a primordial doublet UGX codon, evolving a new meaning.

Another consideration is that in the natural code only eight amino acids treat the 3rd position as equivalent, of which only four appear in Wong's model.[55] Most of the new amino acids would have to be introduced via a single codon. Developing a new tRNA able to simultaneously recognize two codons is unlikely. And nine of the 20 amino acids are represented in the natural code by only two codons. To illustrate, Histidine (CAC, CAU) and Glutamine (CAA, CAG) share the same first two positions although neither amino acids is found in Wong's list of initial amino acids. As a second example, Asparagine (AAC, AAU) and Lysine (AAA, AAG) also share the first two positions and also are not found in Wong's list.

In the two evolutionary variants above, doublet gaps are produced as the price of introducing a new amino acid and assigning to a unique triplet codon. However, mutations would lead to many deadly reading-frame shifts and nonsense (undecipherable) codons. For example, if *330* is recognized as a codeword but not *33*, then single bp mutations could lead to the following effects:

Mutations in a triplet codeword

a. *330 → 030, 130, 230, 300, 310, 320* leading to mutated codons, the first two letters of which define a doublet, resulting in a reading-frame shift.
b. *330 → 331, 332, 333* which have no meaning (if misinterpreted as a doublet these would also lead to a frameshift).

Mutations in a doublet codeword

c. *03 0, 13 0, 23 0, 30 0, 31 0, 32 0 → 330* misinterpreting the doublets (followed by a 0 in the third position) as the new triplet codeword, leading to a reading-frame shift.
d. *03 i, 13 i, 23 i, 30 i, 31 i, 32 i →* 33i, where *i* = 1, 2 or 3. This leads to codes like 331, 332, 333 which are also nonsense codewords, since the *33* doublet can no longer be decoded.

As more doublet codewords get transformed into triplets the potential for deadly frameshift and nonsense mutations increases as shown in table 6.

Mutations would have been more deleterious earlier

One would expect a putative primitive genetic code, based on so few amino acids, to be highly error-prone. Could these primitive proteins have tolerated countless mutations without leading to error-catastrophe? There are two reasons to argue mutational rates would have been very high:

1. In a series of studies, Axe demonstrated[56,57] that extant proteins have extra built-in robustness, permitting them to tolerate a number of mutations as long as too many don't occur, especially if not close together in the protein's folded state. This robustness is unnatural,[58] and for primitive proteins based on a reduced set of amino acids, would be even less so. This view is reinforced by the fact that many evolutionists believe the genetic code

Table 6. Evolution from a doublet to a triplet genetic code would have led to a multitude of deadly mutations. Unlike the natural code, base pair mutations would often lead to reading-frame shifts. All single point mutations are assumed to be equiprobable. Doublet to doublet mutations like *12 → 22* have been neglected. The mutations take the whole codewords into account, so a 1/3 factor was added to the probability results to calculate on a per base basis. Rate of base-pair mutations = **p**.
Codewords 1–15: *(00),(01),(02),(03),(10),(11),(12),(13),(20),(21),(22),(23),(30),(31),(32)*
Codewords 16–17: *(330),(331)*[a]
See text for evolutionary scenarios.

Nr. of 3-letter cws	Effect per 3- letter cw added	Resulting frameshift mutations [n]	Resulting amino acid conversion [b]	Resulting nonsense mutations[c]
1	(330)	1/3 x p x 5/32 [d) + e)]	0 [f]	1/3 x p x 5/32 [g) + h)]
2	(330), (331)	1/3 x p x 5/17 [i) + j)]	1/3 x p x 2/51 [k]	1/3 x p x 7/51 [l) + m)]

[a] Doublet (33) is no longer to be identified as a codeword (cw).

[b] AA = amino acid. Mutation leads to another codeword

[c] Mutation leading to a codeword which cannot be interpreted

[d] Three-letter cw → two-letter cw (frameshifts):
(**3**30) → (03) | (13) | (23) = p x (1/16) x (3/3) = p x (1/16);
(3**3**0) → (30) | (31) | (32) = p x (1/16) x (3/3) = p x (1/16); => p x (2/16)

[e] Two-letter cw → three-letter cw (frameshifts):
(03) | (13) | (23) → (**3**30) = p x (3/16) x (1/3) x (1/4) = p x 1/64;
(30) | (31) | (32) → (3**3**0) = p x (3/16) x (1/3) x (1/4) = p x 1/64 => p x (2/64)

[f] There are no three-letter cws to mutate to.

[g] Three-letter cw → nonsense and (frameshifts if misinterpreted as some doublet):
(33**0**) → (331) | ((332) | (333) = p x (1/16) x (3/3) = p x (1/16)
(since (33) is not a codeword)

[h] Two-letter cw → nonsense, (33)**i** = (33)1 | (33)2 | (33)3:
(03) | (13) | (23) → (**3**3**i**) = p x (3/16) x (1/3) x (3/4) = p x (3/64);
(30) | (31) | (32) → (**33**i) = p x (3/16) x (1/3) x (3/4) = p x (3/64) => p x 6/64

[i] Three-letter cw → two-letter cws (frameshifts):
(**3**30) → (03) | (13) | (23) = p x (1/17) x (3/3) = p x (1/17);
(**3**31) → (03) | (13) | (23) = p x (1/17) x (3/3) = p x (1/17);
(3**3**0) → (30) | (31) | (32) = p x (1/17) x (3/3) = p x (1/17);
(3**3**1) → (30) | (31) | (32) = p x (1/17) x (3/3) = p x (1/17); => p x (4/17)

[j] Two-letter cw → three-letter cw (frameshifts):
(03) | (13) | (23) → (**3**30) = p x (3/17) x (1/3) x (1/4) = p x 1/(4x17);
(03) | (13) | (23) → (**3**31) = p x (3/17) x (1/3) x (1/4) = p x 1/(4x17);
(30) | (31) | (32) → (3**3**0) = p x (3/17) x (1/3) x (1/4) = p x 1/(4x17);
(30) | (31) | (32) → (3**3**1) = p x (3/17) x (1/3) x (1/4) = p x 1/(4x17); => p(1/17)

[k] Mutation to another three-letter cw:
(33**0**) → (331) = p x (1/17)(1/3) = p x 1/(3x17);
(33**1**) → (330) = p x (1/17)(1/3) = p x 1/(3x17) => 2/(3x17)

[l] Three-letter cw → nonsense and frameshifts:
(33**0**) → (332) | (333) = p x (1/17) x (2/3) = p x 2/(3x17) ;
(33**1**) → (332) | (333) = p x (1/17) x (2/3) = p x 2/(3x17) ; => p x 4/(3x17)
(since (33) is not a codeword)

[m] Two-letter cw → nonsense, (33)**i** = (33)2 | (33)3:
(03) | (13) | (23) → (**3**3**i**) = p x (3/17) x (1/3) x (2/4) = p x 1/(2x17);
(30) | (31) | (32) → (**33**i) = p x (3/17) x (1/3) x (2/4) = p x 1/(2x17) => p x 1/17

is almost as old as our planet[59] or at least as old as the Last Universal Common Ancestor (LUCA) about 2.5 billion years ago[60,61] (which would provide virtually no time for proteins to have been optimized).

2. It is very unlikely sophisticated, complex error-correcting machinery, as found in living organisms today, could be developed with only a handful of the 20 amino acids.

Conclusions

Various authors have claimed the quaternary natural genetic code is as compact as it can be if 20 amino acids and stop instructions are to be encoded. This is not correct since the Kraft Inequality informs us (and we demonstrated) that shorter instantly decoded codes, based on variable-length codewords are theoretically possible. Nevertheless, we show that such codes would be impractical to implement on a mechanically biological system and would be highly error-prone.

Our analysis discredits the notion that an earlier doublet genetic code could mutate into the natural triplet genetic code. A compressed version cannot evolve into a triplet version without producing catastrophic frameshifts. A doublet code which relies on a spacer base in the third position, presumably to enhance interactions with anticodons, makes no sense, since the first position of a next doublet could have produced the same effect far more efficiently.

The triplet codon code offers redundancy and considerable protection against mutational and translational errors, and the choice of a *block code* minimizes the possibility of frameshifts. A doublet code evolving into a triplet version would be too vulnerable to reading-frame shifts to have been feasible.

Appendix. Compressed genetic codes would be too vulnerable to errors

Perhaps one could optimize a trade-off between compactness and robustness to find a better code than the natural genetic one. The maximum number of unique codewords, based on a quaternary code, using any combination of codewords with lengths ranging from one to three is 64, see figure 4.

Two-thirds of the codewords in the Compact Genetic Code proposed in table 2 are two letters long, but the code offers no protection to single mutations. Suppose a code was designed by avoiding two-letter codewords by using only single and three-letter codewords.

We observe that for a code containing a single *0* as a codeword all branches beginning with *0* become disallowed to permit instantaneous decoding (the four forbidden doublets

00, *01*, *02*, *03* would eliminate 16 triplets *001, 002 …*) leading to 64 + 1 – 16 = 49 codewords (see figure 4). Clearly, the same applies if characters 0 ... 3 are used. The number of instantly decodable codewords can be calculated[62] by

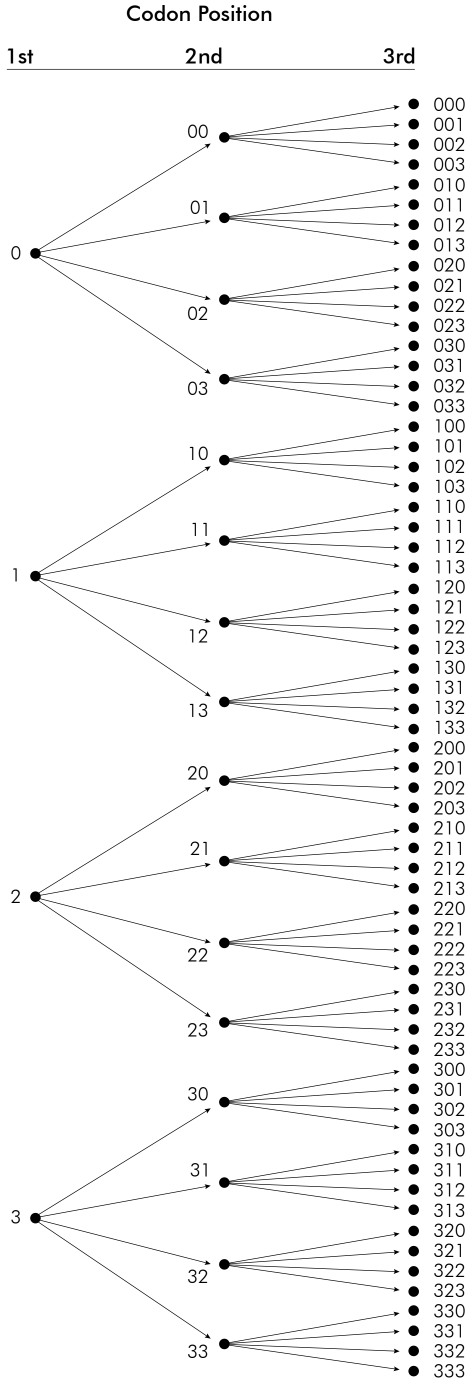


Figure 4. Maximum number of codewords based on a quaternary code. Characters 0 ... 3 can represent nucleotides (**A,C,G,T**) or other chemical structures. If a single character codeword is used, it cannot be a prefix in the second column, removing 4 x 4 = 16 three-letter codewords. Two-letter codewords cannot appear as prefixes in three-letter codewords; each two-letter codeword removes 4 possible three-letter ones. The largest number of codewords results when only three-letter codewords are used, 4 x 4 x 4 = 64.

$$64 - 15j \qquad (5)$$

where j is the number of single-letter quaternary codewords and the remaining are all triplets.

The proportion of single mutations (or mistranslations) which would lead to a reading-frame error is calculated[63] by

$$(1/3) \text{ x } (j/3) \text{ x } (64 - 16j)/64 \qquad (6)$$

for $j = 0$ to 3 single-character codewords (for $j = 4$ there can be no three-letter codewords: 64 – 16 x 4 = 0).

Equation (5) shows that using three or more single-character codewords does not permit 21 amino acids plus Stop from being represented (table 7), so we focus on only two feasible scenarios. From the summary in table 7 we see that using one single-character cw leads to an average codeword length[64] of 2.96, forfeits about a fourth of the available codewords which could have been used to add robustness to mutations through redundancy, and 8% of all mutations lead to a frame-shift.

Using two single-character codewords leads to an average cw length of 2.88, forfeits almost half of the available codewords, and 11% of all mutations lead to a frameshift (table 7). Since only 34 codewords would be available, this code could only provide 34 – 21 = 13 synonyms for the twenty amino acids and Stop, unlike the genetic code which on average offers 64/21 = 3-fold redundancy.

It becomes apparent that the negligible gain in compression would be accompanied by considerable loss in error tolerance (bp mutations and mistranslations) compared to the natural genetic code. The same conclusions are reached upon experimenting with other versions of a compressed code.

References

Table 7. Average codeword length and percentage point mutations leading to frameshifts as a function of one-character codewords in a theoretical genetic code

One-character cw[a)], j	Three-char. cw[b)]	Total cw used[a),c)]	% usage of maximum cw[d)]	Average length of cw [a),e)]	% Point mutations leading to frame-shift [f)]
0	64	64	100	3	0
1	48	49	76.6	2.96	8.3
2	32	34	53.1	2.88	11.1
3	16	19	29.7	2.68	8.3
4	0	4	6.25	1	---

a) cw = “codeword(s)”

b) 64 – 16j

c) Number of one-char + three-char cw

d) Fraction of maximumum codewords possible (64) made use of: 100 x Total cw used/64

e) (1 x one-char. cw + 3 x three-char. cw)/total cw used

f) 100 x (1/3) x (j/3) x (64 – 16j)/64. See main text.

1. Gitt, W., Compton, B. and Fernandez, J., *Without Excuse*, Creation Book Publishers, Atlanta, GA, 2011.
2. Gitt *et al.*, ref. 1, p. 164.
3. A code groups combinations of characters into *codewords*. For example, the Morse code uses specific combinations of dots and dashes to define the English alphabet. The ASCII code combines seven or eight patterns of zeros and ones to define codewords which are letters, numbers and some special symbols.
4. Gitt *et al.*, ref. 1, p. 166.
5. Freeland, S. and Hurst, L.D., The Genetic Code Is One in a Million, *J. Mol. Evol.* **47**:238–248, 1998.
6. Nucleotides *C* and *T/U* are chemically pyridines, which contain one ring (*T* is used by DNA and *U* by RNA). *A* and *G* are purines, which have two rings, and are therefore larger than *C* and *T/U*. Mutations or mistranslations (misinterpretation of a codon leading to the wrong amino acid being added to the protein) are known to occur more readily between 'like with like': *C* ↔ *T/U* and *A* ↔ *G* (called transitions) than mutations of the kind *C*, *U* ↔ *A*, *G* (called transversions).
7. Truman, R. and Terborg, P., Genetic code optimisation: Part 1, *J. Creation* **21**(2):90–100, 2007; see p. 97.
8. Freeland and Hurst, ref. 5, p. 239.
9. Having more than one Stop codon can also be advantageous. If an accidental insertion or deletion of a base occurs (leading to an incorrect reading frame and linking the wrong amino acids thereafter), the chances of accidently stumbling on a Stop codon increases, quickly terminating the worthless efforts. This saves energy and building materials.
10. Baranov, P.V., Venin, M. and Provan, G., Codon Size Reduction as the Origin of the Triplet Genetic Code, *PLoS ONE* **4**(5):e5708, 2009. doi:10.1371/journal.pone.0005708; www.plosone.org/article/info:doi/10.1371/journal.pone.0005708.
11. Yockey, H.P., *Information Theory, Evolution, and the Origin of Life*, Cambridge University Press, MA, 2005; see p. 108.
12. A genetic code must be designed using mechanical parts like ribosomes, DNA and mRNA. Logic programming as performed in a computer's memory is not relevant. Although non-IDCs are perfectly valid in principle, considerable engineering disadvantages result if used in cells. It would be necessary to position the polymer carrying the message in a very special manner to extract each codeword, and a strategy would be needed to disfavour codewords which are prefixes of longer ones (see Addendum). Thus, the requirement that a genetic code be an IDC is reasonable.
13. Togneri, R. and deSilva, J.S.C., *Fundamentals of Information Theory and Coding Design*, Chapman & Hall/CRC, Boca Raton, FL, 2003; see pp. 115–118.
14. There are some minor variants to the standard genetic code, found mostly in the mitochondria genomes of some organisms. See www.ncbi.nlm.nih.gov/Taxonomy/taxonomyhome.html/index.cgi?chapter=tgencodes#thetop.
15. Togneri and deSilva, ref. 13, pp. 118–120.
16. *A* = Adenine, *C* = Cytosine, *G* = Guanine, and *T* = Thymine.
17. Alternative strategies are not as effective. For example:

    Three single-character cws: we could begin ambitiously and try using three single-character cws followed by doublets, avoiding prefixes as we proceed. This provides codewords: *0,1,2,30,31,32,33*. We cannot generate more cws without duplicating a prefix but we don't have the minimum of 21 cws needed (20 amino acids and a Stop). We could replace instead the doublets (*30,31,32,33*) by 16 triplets (*0,1,2,300 … 333*) but the total of 3 + 16 = 19 cws is still too small. No solution works with more than two single-character cws.

    Two single-character cws: we could limit ourselves to two single-character cws and as many two-character cws as possible (*0,1, 20 … 23,30 … 33*). This only provides a total of 2 + 8 = 10 cws. Replacing the doublets by triplets *(0,1,200 … 233,300 … 333)* gives 2 + 32 = 34 cws, enough for a genetic code, and would have an average cw length (assuming equiprobable assignments) of (2 x 1 + 19 x 3)/21 = *2.81*. We can improve on this by replacing as many triplets as possible by doublets. The code (*0,1,20 … 23,300 … 333*) delivers 2 + 4 + 16 = 22 cws with average length (2 x 1 + 4 x 2 + 15 x 3)/21 = *2.62*.

    One single-character cw: we can test one single-character cw and as many doublets as possible before relying on triplets to reach at least 21 cws: (*0,10…13,20…23,30,310…333*), which provides 1 + 9 + 12 = 22. This code would have an average cw length of (1 x 1 + 9 x 2 + 11 x 3)/22=*2.48*.

    To find the code with the shortest average-length cw, no single-character cws should be used, as done in the main text to permit a far greater number of doublets to be generated. The most compressed code has an average cw length of *2.33* (see main text).
18. Personal communication, information theory class notes by Professor Roberto Togneri.
19. As an alternative coding convention we could have chosen for the necessary 21 symbols: (*00,01,02,03,10,11,12,13,20,21,22,23,30,31)* for the 14 two-letter codewords and (*320,321,322,323,330,331,332)* for the seven remaining three-letter codewords.
20. Gitt *et al.*, ref. 1, p. 166.
21. Although some cellular processes to corrrect −1 and +1 frameshift reading errors have been discovered, these are relatively rare and can only correct a small minority of such errors.
22. For example, suppose *033* is assigned as a synonym for s_{18}: then a mutation from *023* to *033* would not matter but all other eight bp mutations of *023* would lead to a different two- or three-letter codeword: *023* → *123*; *223*; *323*; *003*; *013*; *020*; *021; 022*.
23. http://creation.com/evaluation-compact-codes
24. Jukes, T.H., Coding triplets and their possible evolutionary implications, *Biochem. and Biophys. Res. Commun.* **19**:391–396, 1965.
25. Truman, R.T. and Terborg, P., Genetic code optimisation—Part 2, *J. Creation* **21**(3):84–92, 2007.
26. This can be calculated by 21! x S (64,21), where S is a Stirling's number of the second kind. See ref. 25 for a dynamic programming boot-strap algorithm.
27. 15! x S(16,15) = 1.6 x 10^{14}, using the program described in ref. 25 to find S(16,15).
28. Trifonov, E.N., Consensus temporal order of amino acids and evolution of the triplet code, *Gene* **261**:139–151, 2000.
29. The first position could have been present as a spacer only, or to anchor the start of the codon, and only the last two positions would be informative. But a doublet based on the 1st and 3rd position would be far more interesting: now the starting and ending points of the codon would be uniquely identified (avoiding the reading frameshift problem of later mutating to a tertiary code!) and an extra position left conveniently available in the middle for evolution to work on. Suppose we had found that 15 amino acids could be coded for in the natural genetic code with such a doublet: this would have been a strong argument for a predecessor doublet code!
30. To illustrate, let each of the following pairs code for the same amino acid or a Stop instruction: (00;01) (02;03) (10;11) (12;13) (20;21) (22;23) (30;31) (32;33). Only mutations in the second position are neutral, and of the three possible mutational outcomes only one is protected: 1/2 x 1/3 = 1/6, or about 17%.
31. Crick, F.H.C., The origin of the genetic code, *J. Mol. Biol.* **38**:367–379, 1968.
32. Wong, J. T.-F., The evolution of a universal genetic code, *Proc. Natl.*

Acad. Sci. USA **73**:2336–2340, 1976.

33. Osawa, S., Jukes, T.H., Watanabe, K. and Muto, A., Recent evidence for evolution of the genetic code, *Microbiol. Rev.* **56**(1):229–264, 1992, www.ncbi.nlm.nih.gov/pubmed/1579111; see p. 257.
34. Osawa *et al.*, ref. 33, p. 247.
35. Osawa *et al.*, ref. 33, p. 230.
36. en.wikipedia.org/wiki/Transfer_RNA#cite_note-15.
37. Goodenbour, J.M. and Tao, P., Diversity of tRNA genes in eukaryotes, *Nucl. Acid Res.* **34**(21):6137–6146, 2006; nar.oxfordjournals.org/content/34/21/6137.full.
38. Osawa *et al.*, ref. 33, p. 249.
39. Osawa *et al.*, ref. 33, p. 235: According to a theory developed by Sueoka, the G + C content is determined by the base conversion rate u (G • C to A • T) and v (A • T to G • C); the G + C content at equilibrium (p) is v/(u + v). 'AT pressure' indicates more A + T is favoured, and 'GC pressure', more G + C is favoured. The most probable cause of such differences is copy errors during DNA replication. Among mutator genes (mut) of *E. coli*, a mutation of the *mutT* gene specifically induces transversions from A • T to C • G pairs at a high rate and that of mutY does the same from G • C to T • A pairs. The mut gene products are mainly components functioning in DNA replication or repair.
40. Osawa *et al.*, ref. 33, p. 251.
41. T.H., Jukes, *Molecules and Evolution*, Columbia University Press, New York, 1966.
42. Osawa *et al.*, ref. 33, p. 233.
43. Baranov, P.V., Venin, M. and Provan, G., Codon size reduction as the origin of the triplet genetic code, *PLoS ONE* **4**(5):e5708, 2009; doi:10.1371/journal.pone.0005708, www.plosone.org/article/info:doi/10.1371/journal.pone.0005708.
44. Dunham, C.M, Selmer, M., Phelps, S.S., Kelley, A.C, Suzuki,T., Joseph, S. and Ramakrishnan, V., Structures of tRNAs with an expanded anticodon loop in the decoding center of the 30S ribosomal subunit, *RNA* **13**:817–823, 2007; doi: 10.1261/rna.367307, rnajournal.cshlp.org/content/13/6/817.long.
45. Tuohy, T.M.F., Thompson, S., Gesteland, R.F. and Atkins, J.F., Seven, eight and nine-membered anticodon loop mutants of tRNA2Arg which cause +1 frameshifting: Tolerance of DHU arm and other secondary mutations, *J. Mol. Biol.* **228**(4):1042–1054, 1992; www.sciencedirect.com/science/article/pii/0022283692903139.
46. Kothe, U. and Rodnina, M.R., Codon reading by tRNAAla with modified uridine in the wobble position, *Molecular Cell* **25**(1):167–174, 2007; www.cell.com/molecular-cell/abstract/S1097-2765%2806%2900784-2.
47. For example, the number of four-character quaternary genetic codes is overwhelming, and converging down to the natural triplet code cannot occur. The number of alternative codes, 21! x S(256,21), could not be calculated using the computer program from ref. 25. However, 21! x S(248,21) = 8.1 x 10^{327} could be handled, so the number must be greater than this.
48. Truman, R. and Terborg, P., Genome truncation vs mutational opportunity: can new genes arise via gene duplication?—Part 1, *J. Creation* **22(1)**: 99–110, 2008; Truman, R. and Terborg, P., Genome truncation vs mutational opportunity: can new genes arise via gene duplication?—Part 2, *J. Creation* **22(1)**:111–119, 2008.
49. The streamlined organisms would have a selectivity factor of about s = 0.3, every generation, by the arguments discussed in ref. 48. With a reproduction time in the minutes-to-days range, the whole world's population would converge to the simpler genetic version in virtually no time.
50. Osawa *et al.*, ref. 33, p. 240.
51. Brown, C.M., Stockwell, P.A., Trotman, C. N. A. and Tate, W.P., Sequence analysis suggests that tetra-nucleotides signal termination of protein synthesis in eukaryotes, *Nucleic Acids Res.* **18**:6339–6345, 1990.
52. There are a wide variety of tRNAs used to decode codons today and many variants, hundreds in many organisms, ref. 36. Chemical modifications on the tRNAs deliberately enhance or prevent specific interactions with specific codons. All this would be lacking in models like those Jukes proposes.
53. Yockey, ref. 11, p. 105.
54. Ferredoxins are less than a third of the size of an average protein, en.wikipedia.org/wiki/Ferredoxin.
55. Eight amino acids in the natural code treat the codon's third position as equivalent: Ala, Arg, Gly, Leu, Pro, Ser, Thr, Val. Wong's model assumes the following amino acids were used in a doublet code: Ala, Asp, Glu, Gly Phe, Ser, Val, meaning that 16 amino acids out of 20 would have to treat the 3rd position as no longer equivalent over time by evolutionary reasoning.
56. Axe, D.D., Extreme Functional sensitivity to conservative amino acid changes on enzyme exteriors, *J. Mol. Biol.* **301**:585–595, 2000.
57. Axe, D.D., Estimating the prevalence of protein sequences adopting functional enzyme folds, *J. Mol. Biol.* **341**:1295–1315, 2004.
58. Evolution can't select to avoid future problems like random mutations. And unlike natural proteins, those designed by biochemists are typically vulnerable to just one mutation, even though they fold properly: Taverna, D.M. and Goldstein, R.A., Why Are Proteins So Robust to Site Mutations? *J. Mol. Biol.* **315**:479–484, 2002.
59. Eigen, M., Lindemann, B.F., Tietze, M., Winkler Oswatitsch, R., Dress, A. and Haeseler, A., How old is the genetic code? Statistical geometry of tRNA provides an answer, *Science* **244**:673–679, 1989.
60. Knight, R.D., Freeland, S.J. and Landweber, L.F., Rewiring the Keyboard: Evolvability of the Genetic Code, *Nature Reviews* **2**:49–58, 2001.
61. Gu, X, The Age of the Common Ancestor of Eukaryotes and Prokaryotes: Statistical Inferences, *Mol. Biol. Evol.* **14**(8):861–866, 1997.
62. There are 64 possible three-letter codewords. Adding *j* one-letter codewords provides *j* more codewords at the expense of 16*j* three-letter codewords: 64 + *j* – 16j = 64 – 15*j*.
63. The expression (1/3) x (*j*/3) x (64 – 16*j*)/64 consists of three parts. (1/3): There is a 1/3 chance the mutation will occur in the 1st position. (*j*/3): given that a mutation did occur and that there are four characters in the alphabet (*0*,*1*,*2* and *3*), this reflects the probability of matching an existing one-letter codeword. (64 – 16*j*)/64: this is the probability the mutation occurred in a three-letter codeword.
64. Assuming the use of each amino acid is equiprobable.

Royal Truman has bachelor's degrees in chemistry and in computer science from State University of New York; an MBA from the University of Michigan (Ann Arbor); a Ph.D. in organic chemistry from Michigan State University; and a two-year post-graduate 'Fortbildung' in bioinformatic from the Universities of Mannheim and Heidelberg. He works in Germany for a European-based multinational.