# A crescendo of competent coding (c3) contains the Standard Genetic Code

MICHAEL YARUS

Department of Molecular, Cellular and Developmental Biology, University of Colorado, Boulder, Colorado 80309-0347, USA

## ABSTRACT

**The Standard Genetic Code (SGC) can arise by fusion of partial codes evolved in different individuals, perhaps for differing prior tasks. Such code fragments can be unified into an SGC after later evolution of accurate third-position Crick wobble. Late wobble advent fills in the coding table, leaving only later development of translational initiation and termination to reach the SGC in separated domains of life. This code fusion mechanism is computationally implemented here. Late Crick wobble after C3 fusion (c3-lCw) is tested for its ability to evolve the SGC. Compared with previously studied isolated coding tables, or with increasing numbers of parallel, but nonfusing codes, c3-lCw reaches the SGC sooner, is successful in a smaller population, and presents accurate and complete codes more frequently. Notably, a long crescendo of SGC-like codes is exposed for selection of superior translation. c3-lCw also effectively suppresses varied disordered assignments, thus converging on a unified code. Such merged codes closely approach the SGC, making its selection plausible. For example: Under routine conditions, ≈1 of 22 c3-lCw environments evolves codes with ≥20 assignments and ≤3 differences from the SGC, notably including codes identical to the Standard Genetic Code.**

**Keywords: HGT; anticodon; codon; origin; triplet**

## INTRODUCTION

The Standard Genetic Code associates 22 functions (20 amino acids plus initiation and termination) with the 64 possible ordered RNA triplets in a way reproduced with appreciable accuracy throughout Earth's biota. This implies that the SGC preexisted in predecessors of all modern Earth creatures. Thus, the SGC's derivation offers information about early biology before the common ancestor of modern organisms, and during divergence into present (Zhou et al. 2018) domains of life.

Here, such information is sought by quantitative modeling of SGC emergence, using arguably general assumptions (Yarus 2021b). It is assumed only that codon assignment, capture and decay (and added here: new coding tables and code fusion) occur at characteristic rates. SGC existence is attributable to the joint effects of those rates within the 64 triplet space of a coding table.

In order to embody events whose complexity may be great, but unknown at the start, a computable model is used (Fig. 7; Yarus 2021b). This envisions SGC evolution as a set of shorter intervals, called passages, during which one event only (a codon assignment, for example, or possibly no event) occurs. Encoding events differ in probability during a passage. The virtue of this formulation is that it allows explicit programming of hypotheses about code evolution, even for histories of great complexity. Implied coding tables are explicitly computed. Moreover, using different probabilities during an interval is equivalent to assignment of different rate constants (Yarus 2021b), and timing of events may therefore be compared. For example, when the real-world time to assign a codon is known, these calculations will convert to early Earth times (Yarus 2021d).

These inquiries take recent form (Yarus 2021c) as the idea that the code was composed by fusing independently arising partial codes, perhaps combining primitive compartments with differing coding competencies. Code fusion is common in Biology, having been observed many times, for example, between mitochondrial and nuclear codes (Duchêne et al. 2009).

Creation of the SGC by fusion of separate partial codes was thought (Yarus 2021c) to have specific advantages; for example, realizing the SGC within a smaller code

**Corresponding author: yarus@colorado.edu**

Article is online at http://www.rnajournal.org/cgi/doi/10.1261/rna.079275.122. Freely available online through the *RNA* Open Access option.

population. Such hypothetical fusion advantages are tested below.

## RESULTS

### Individual coding tables

Developing primordial codes (Yarus 2021b) may assign unassigned triplets (with probability Pinit), using either SGC-like assignments or random assignments (probability Prand), or capture unassigned triplets related by single mutations from their existing assignments (Pmut). Such assignments, however, can decay and be lost (Pdecay). Probabilities have the same relative values for passages here as in prior studies, so present codes resemble those earlier ones. However, probabilities have also been reduced in proportion so that two events in one interval are less probable. This makes the present model somewhat more accurate, though more passages elapse during code evolution.

### Fusion of codes

In addition, new events (Fig. 7) occur in this work: with constant probability/passage, new coding tables appear (Ptab). New codes begin with a single arbitrary assignment (Pinit), then evolve using the same rules as the initiating coding table. Newly originated codes accumulate; once these exist, they may fuse with other codes (Pfus) with a probability that increases with the number of possible partners. A fused compartment can gain assignments from both fusees, or it can be unchanged, if both happen to use overlapping prior encodings. However, fusion can also be disastrous, if fusing codes conflict. A simple rule is used: if fused codes give a triplet more than one meaning, this will be damaging, and both participants are lost.

### The evolutionary goal

The assumption is: there was a functional advantage to SGC coding, and codes more like it increasingly possess that advantage. To avoid unnecessary hypotheses, superiority of the SGC is unspecified. Instead, code selection is more probable as the distance to the SGC decreases. This is implemented by seeking codes that are sufficiently complete: they encode ≥20 of the 22 possible functions (recognizing the late development of definitive initiation and termination, Burroughs and Aravind 2019). In addition, codes must be accurate: they vary from the SGC by the fewest misassignments, abbreviated "misx," where x is the number of differences from the SGC. Codes closest to SGC completeness and SGC assignments (called SGC-like codes) are most likely to have been selected, whatever (yet unknown) selection may have applied.

### Fusion must be a major evolutionary event

Coding evolution is altered by fusion only if it occurs significantly. A significance requirement is both elementary and profound; we discuss fusion rates first.

Code fusion requires two successive events. Firstly, new coding tables arise to create a population of codes in the initial code's environment. This happens at a fixed probability per passage (Ptab); a passage being mean time for completion of one evolutionary step for a coding table.

In the second step, tables, once multiple ones exist, may fuse their codes at random, with probability Pfus per passage, Pfus*(others). (others) is the number of codes existing alongside each fusion candidate; thus (others) = (total codes − 1). Fusions become increasingly probable with time.

Time is measured in passages, which simultaneously host evolution in all existing codes (Materials and Methods). Either a new assignment, capture of a sequence-related codon, assignment decay, or a code fusion may occur within a passage, perhaps accompanied by creation of a new coding compartment that begins with one assigned codon, and evolves alongside the initial code.

As an environment's code population increases, it acquires more complex coding, and the program records these events in any detail desired. Every change and every intermediate code can be recorded and delivered to the experimenter. But because only a small part of total change is usually of interest, only partial data are routinely reported. The first is a summary of the important properties of every code in an environment at every passage. This allows study of average events for all codes. The second kind of report tracks only the most complete codes (e.g., most functions assigned). This emphasizes advanced coding, useful if progress to SGC-like codes is being studied.

### A majority of fusions

Figure 1 plots the fraction of codes with ≥20 assigned functions that have benefitted from a fusion contribution, versus the probability of new tables (Ptab in Fig. 1A, Pfus is fixed and favorable), or versus the probability of fusion (Pfus in Fig. 1B, Ptab is fixed and favorable). More tables and more likely fusion, as expected, increase the fraction of SGC-like (≥20 function) codes that acquire fused assignments. Because this ms concerns change produced by code fusions, Ptab = 0.08 and Pfus = 0.002 (Fig. 1, rightward, yielding frequent fusion) are used below. A requirement for elevated fusion agrees with other studies (Aggarwal et al. 2016), and this work provides a simple rationale: fusion must be a frequent route to final codes.

### Altered population history

Figure 2 contains averages for every coding table in 1000 code populations carried to 750 passages. Beginning with

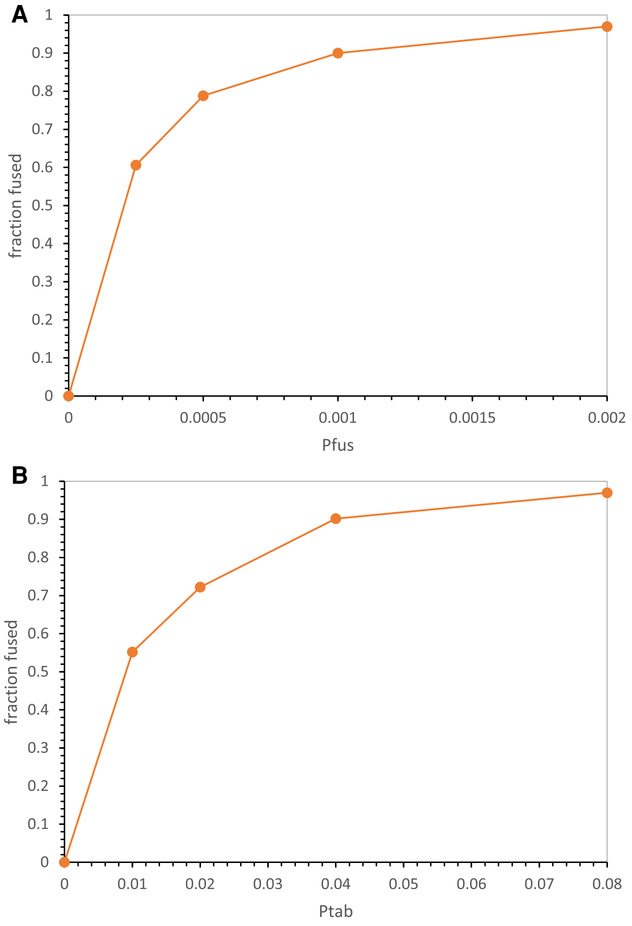


**FIGURE 1.** (*A*) Complete fused codes versus probability of fusion. Evolution was stopped when the first code with ≥20 functions appeared in an environment. The fraction, among 500 environments, of such ≥20 function codes that acquired assignments from fusion is plotted versus the probability of fusion during a passage, Pfus. Pmut = 0.00975, Pdecay = 0.00975, Pinit = 0.150, Prand = 0.050, Ptab = 0.08. (*B*) Complete fused codes vs probability of coding table initiation. Evolution was stopped when the first code with ≥20 functions appeared in an environment. The fraction, among 500 environments, of such ≥20 function codes that acquired assignments from fusion is plotted versus the probability that a new coding table is initiated during a passage, Ptab. Pmut = 0.00975, Pdecay = 0.00975, Pinit = 0.150, Prand = 0.050, Pfus = 0.002.

a single initial code, this ranges from 5412 potential codes at 60 passages to 60,790 potential codes at 750 passages. Thus, accurate mean values for code kinetics are available. In Figure 2, all coding tables are partitioned into four classes—they have had no fusion, have been lost in fusion, have been annihilated by incompatible fusion, or have received successful fusion (Fig. 7).

These times allow unfused tables to become infrequent (only about 6% are unfused at 750 passages). The predominant fate of coding tables is loss in fusion, and this is true from early times, just before 140 passages. Codes are lost because they fuse into others, and a significant minority were annihilated by trying to fuse to codes with incompatible assignments. Only 9.4% of once-existent codes still exist at 750 passages (those with no fusion or successful fusions). Successful fusions themselves have a peak around 110 passages, after which they are also lost in later destructive events. Figure 7, which sketches calculations in a simplified environment, may help conceptualize fusion losses. We will return to the early successful fusion maximum (Fig. 2), and to its later decline, below.

## Superior codes follow fusion

We now examine later codes in Figure 2; this minority of fusion survivors includes codes that closely approach the SGC. Figure 3 shows the properties of the most complete codes from 10,000 code populations evolved throughout the interesting era of Figure 2, from 150 to 750 passages, when fusions emerge, then increase and decisively shape an environment's codes.

## Competence

Figure 3A depicts abundance of SGC-like codes. These codes have either experienced no fusion, or a complementary fusion that adds assignments. They have assigned codons to ≥20 of the standard functions, and so are almost complete. Moreover, plotted compartments encode functions very similarly to the SGC, having assignments completely overlapping (blue line), a single differing assignment (red), two differences (gray), or three differently assigned triplets (yellow).

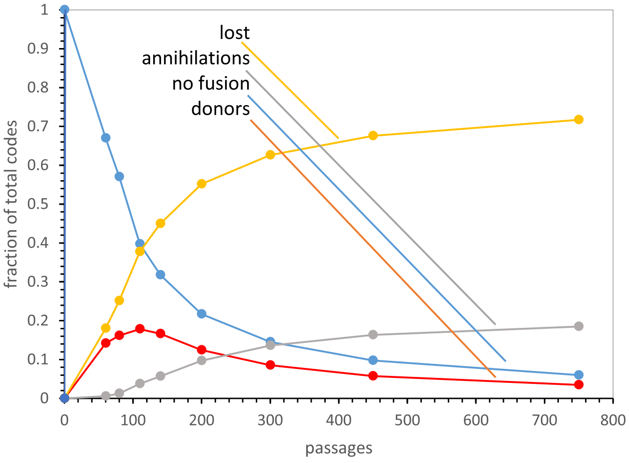
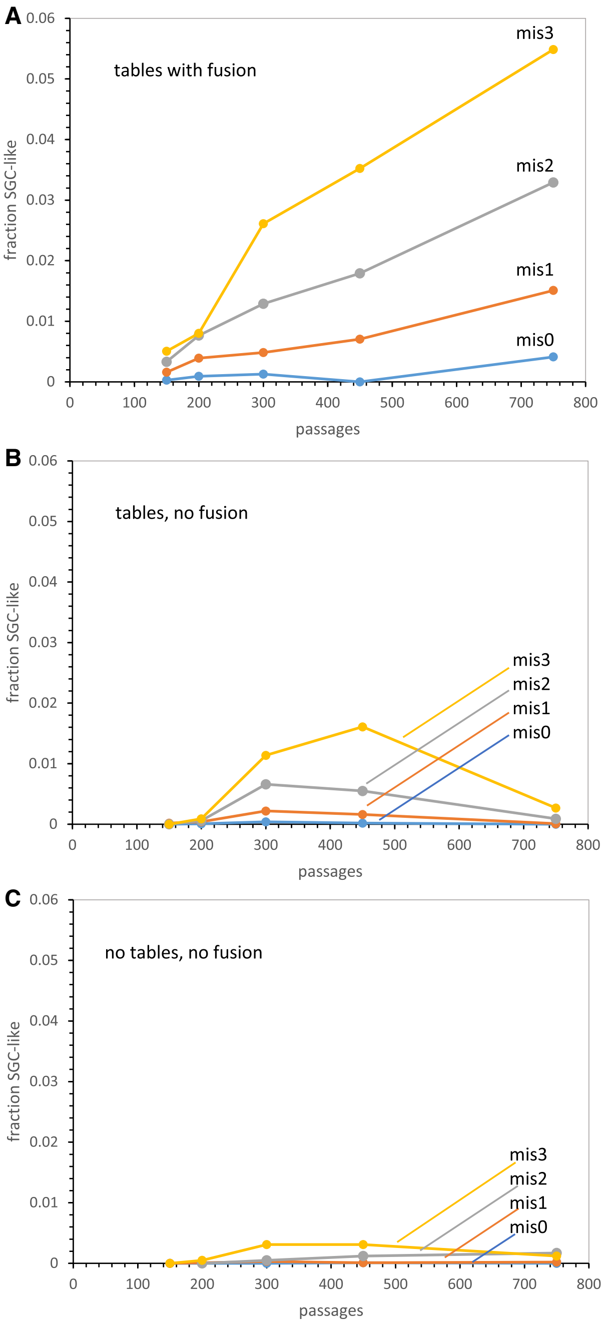


**FIGURE 2.** Code fate versus time. Mean fractions of 1000 total codes are plotted versus passages (time). Kinetics for several fates are shown: donors—codes that successfully fused/annihilations—codes lost in incompatible fusions/no fusion—unfused codes/lost—donors lost in fusion. Pmut = 0.00975, Pdecay = 0.00975, Pinit = 0.150, Prand = 0.050, Pfus = 0.00200, Ptab = 0.08.

## The crescendo

The fraction of competent codes tends to rise from an origin just after the appearance of population-wide fusions (Fig. 2) to the end of calculation (Fig. 3A; the crescendo). In addition, these triply unusual codes, with completeness, fusion contributions and accuracy that are all exceptional, are quite frequent, seemingly well within the reach of a search for SGC-like translation. For example, the completely SGC-like class (mis0) are detected in 10,000 environments early, at 150 passages, and are 1/250 among the best codes at 750 passages. Even supposing a demanding selection, requiring precise SGC mimicry, such codes appear relatively early, and require only selecting superior translation among 250 environments. This seems achievable.

Further, if selection for superior translation extends to all codes similar to the SGC, ≥20 function, mis0 to mis3 codes exist in 1/100 environments at 150 passages, and more frequently than 1/10 environments at 750 passages. Late selection would not seek far for SGC-like results.



## The crescendo evolves

Existing codes (Fig. 3A–C) quickly acquire additional assignments. Less quickly, new codes arise and existing codes fuse. Less frequently yet, new codons are captured for existing assignments and assignments decay. Thus, a large flux of change is absorbed in a code environment. The implication is that the competent code population is constantly changing on a timescale comparable to its initial appearance. The coding crescendo's tables are constantly evolving, with new codes replacing the previously competent: compare successful fusions lost to later events (Fig. 2). Thus, the crescendo offers a changing face to selection, as well as increasingly frequent competent codes. Selection can occur when a particularly effective code appears from the crescendo's jumble.

## The crescendo and its competence come from fusion I

The crescendo of competence is produced by code fusion.

**FIGURE 3.** (*A*) SGC-like codes versus time: tables and fusion. The fraction of almost complete codes (≥20 functions) with cited levels of misassignment (relative to the SGC) is plotted for 10,000 environments that have both new code initiation and fusion. mis0 = no misassignments/mis1 = 1 misassignment, and so on. Ten thousand environments evolved to the times/passages shown, and codes with ≥20 assigned functions were characterized. Pmut = 0.00975, Pdecay = 0.00975, Pinit = 0.150, Prand = 0.050, Pfus = 0.002, Ptab = 0.08. (*B*) SGC-like codes vs time: tables, no fusion. The fraction of almost complete codes with cited levels of misassignment (relative to the SGC) is plotted for 10,000 environments that have new code initiation, but no fusion. mis0 = no misassignments/mis1 = 1 misassignment, and so on. Ten thousand environments evolved to the times/passages shown, and codes with ≥20 assigned functions were characterized. Pmut = 0.00975, Pdecay = 0.00975, Pinit = 0.150, Prand = 0.050, Pfus = 0.000, Ptab = 0.08. (*C*) SGC-like codes vs time: no new tables, no fusion. The fraction of almost complete codes with cited levels of misassignment (relative to the SGC) is plotted for 10,000 environments that have no new code initiation, and no fusion. mis0 = no misassignments/mis1 = 1 misassignment, and so on. Ten thousand environments with single tables evolved to the times/passages shown, and codes with ≥20 assigned functions were characterized. Pmut = 0.00975, Pdecay = 0.00975, Pinit = 0.150, Prand = 0.050, Pfus = 0.000, Ptab = 0.00.

Figure 3B shows code evolution similar to Figure 3A, but without fusion (Pfus = 0). As in other Figure 3 panels, fractions of the most complete codes from 10,000 environments (Fig. 7) are plotted versus passages (time). Having multiple codes itself facilitates the evolution of more complex coding. Thus, environments with parallel coding tables as in Figure 3A, but with no fusions between them (Fig. 3B), present a useful comparison. Figure 3B has the same *y*-axis to facilitate comparison with Figure 3A: no crescendo exists. In fact, for multiple tables without fusion, all levels of completeness with assignment accuracy arise later, achieve lower frequencies among most complete codes, and do not persist. Competence ultimately declines instead of increasing (Fig. 3A).

## The crescendo and its competence come from fusion II

Multiple coding tables facilitate evolution of the SGC without fusing. Figure 3C completes controls for Figure 3A; it describes a similar set of code evolutions, but lacking both multiple tables and fusion (Ptab = 0, Pfus = 0). This resembles the system previously analyzed (Yarus 2021b), where code evolution takes place in a single initial coding table in each environment, each evolving until it resembles the SGC. However, Figure 3C is useful here because its individual codes are the same as those of Figure 3A,B. Figure 3C also has the same time scale, frequency scale and colors as Figure 3A,B. Thus, single codes without fusion gain SGC resemblance later than in Figure 3A, restrict such competence to lower levels even than for multiple tables in Figure 3B, and again show no crescendo (Fig. 3A). In fact, SGC-like coding is everywhere lower than for multiple tables without fusion (Fig. 3B). For example, complete resemblance to the SGC (mis0) is not detected among 10,000 environments until 450 passages and then at too low a frequency (≈$10^{-4}$) to be deciphered on Figure 3C's ordinate.

## Origin of competence

It is clear why environments that fuse code compartments, primitive cells or partial coding tables are superior. Figure 4A compares mean misassignments for the most complete codes (assigned ≥20 functions) from 10,000 environments. Multiple codes with fusion (red, Fig. 4A) are similar to multiple codes without fusion (green, Fig. 4A) and to codes without multiple tables and fusion (blue, Fig. 4A) *until* fusion becomes predominant in code evolution (see Altered population history, above). After this time (≈150 passages, Fig. 4A), errors during different evolutionary modes diverge greatly. Strikingly, other modes almost double mean misassignment in codes with fusion.

There are two sources of misassignment in present code environments. The more straightforward is that random

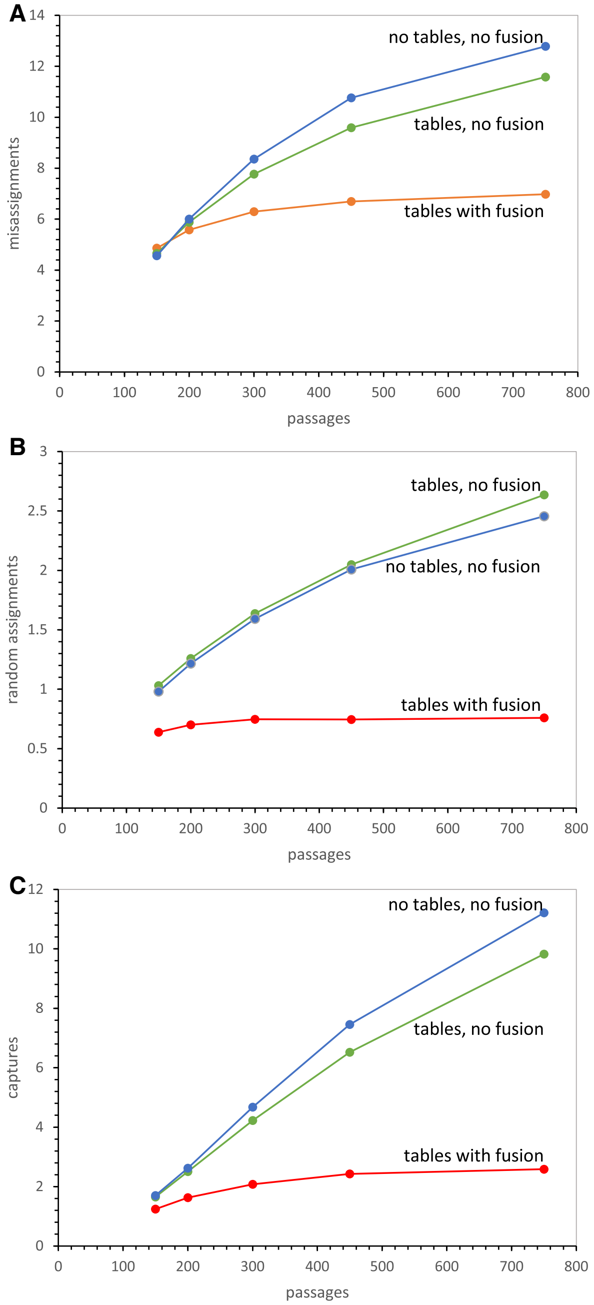


**FIGURE 4.** (*A*) Mean misassignments versus time: tables and fusion, tables no fusion, no tables no fusion. Ten thousand environments were evolved to the times shown, and misassignments relative to the SGC were counted among most complete codes in each environment. Probabilities are the same as in Figure 3 for the three kinds of evolution. (*B*) Randomly assigned codons versus time: tables and fusion, tables no fusion, no tables no fusion. Ten thousand environments were evolved to the times shown, and mean randomly assigned codon triplets were counted among most complete codes in each environment. Probabilities are the same as in Figure 3 for the three kinds of evolution. (*C*) Capture of mutationally related codons versus time: tables and fusion, tables no fusion, no tables no fusion. Ten thousand environments were evolved to the times shown, and mean capture of triplets one mutation distant from assigned codons were counted among most completely assigned codes in each environment. Probabilities are the same as in Figure 3 for the three kinds of evolution.

assignment is allowed by a variable probability of random association between functions and triplets (Prand). Because these assignments are accurately randomized, they are unlikely to be the same as for the SGC.

## Fate of random assignments

Figure 4B shows the mean number of randomized assignments in the most complete codes from 10,000 coding environments (Prand = 0.05) versus time. The pattern strikingly reproduces overall accuracy in Figure 4A. That is, codes derived by fusing multiple tables (blue, Fig. 4B) have made two- to threefold fewer random assignments than without fusion (red, Fig. 4B) or without both multiple tables and fusion (gray, Fig. 4B).

## Fate of captured triplets

A second source of misassignment is that related triplets (one mutation away from an assigned triplet) can be captured for an existing related function. The new assignment can be to a chemically related amino acid (having similar polar requirement, Woese 1965; Mathew and Luthey-Schulten 2008), or even the same as the previously assigned function (Yarus 2021b). Chemically related amino acids are sometimes, but not always, assigned to mutationally related triplets in the SGC, and there are several choices for the "chemically related" one (see Materials and Methods). So, capture also frequently yields encoding unlike the SGC.

Figure 4C shows that codes using fusion more strongly discriminate against captures of mutationally related triplets for related functions. Again, the pattern follows that in Figure 4A: before prevalent fusion, the three modes of code evolution are similar. Afterward, they progressively diverge under fusion; at 750 passages mean fused codes (blue, Fig. 4C) utilize four- to fivefold fewer error-prone captures than do unfused multiple codes (red, Fig. 4C) or single codes (gray, Fig. 4C).

## Codes with misassignments are rejected

How do fused codes become superior? Fusion tests codes against each other because unlike codes are incompatible. Fusions between like codes are more likely to succeed; fusions between unlike codes are more likely to be lost because of the toxic effects of codons with multiple meanings. Thus, when highly complete fused codes are characterized above (Figs. 3, 4), they are intrinsically less heterogeneous than the partial codes from which they have been derived. As a fusing environment progresses, with fusion more and more probable (Fig. 2), heterogeneity due to random assignment (Fig. 4B) and capture of related codons (Fig. 4C) is suppressed among the fused (Fig. 4A). Therefore, the same number of unfused codes in one environment (Fig. 3B), or single codes without potential fusion (Fig. 3C), cannot compete with the accuracy of fusing tables evolving together (Fig. 3A).

## Acceptable randomness

The SGC is strikingly ordered, that is, nonrandom (Woese 1965). An important question for any evolutionary path is therefore: how much random assignment can be tolerated? Increased competence via fused nascent codes specifically raises the possibility that fusion increases the latitude for assignment in early coding. Figure 5 thus presents data for random assignment from none (Prand = 0) to about 2.2 random assignments/code on average (Prand = 0.1).

Data in Figure 5 are for tens of thousands of environments at 300 passages, a time when all modes are evolving SGC-like codes (Fig. 3). The frequencies of near-complete, accurately assigned codes are plotted logarithmically versus the linear probability of random assignment (Prand). Log frequencies of such SGC-like codes tend to decrease linearly with Prand (see also Yarus 2021c), with decrease somewhat more rapid as the rigor of requirement increases. Thus, inerrant codes (mis0) decrease somewhat more than those with three misassignments (mis3). But even if Prand = 0 other sources of error remain, like capture of mutationally related triplets for similar assignments (Fig. 4). Frequencies for good coding shown are higher than we have previously observed. These origin hypotheses still have substantial access to the SGC even if they randomly assign triplet functions in roughly one of 10 cases. Thus, the prior rule-of-thumb (Yarus 2021b) need not change

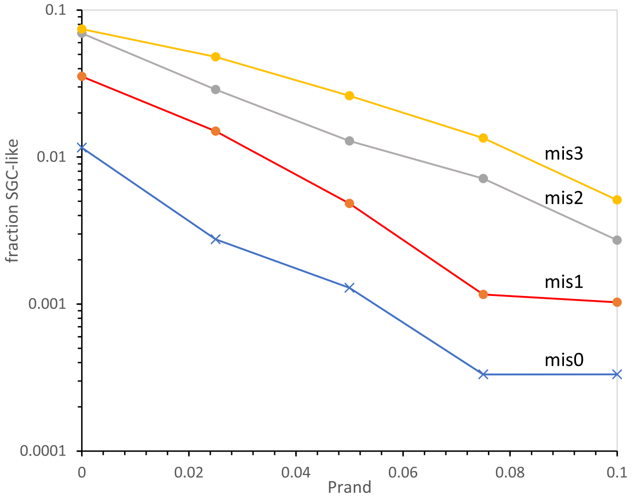


**FIGURE 5.** Fraction SGC-like codes versus probability of random assignments. A total of 10,000 (or 20,000 for greatest Prand) environments were evolved to 300 passages. Among substantially complete codes (≥20 assigned functions), fractions with different levels of misassignment were counted. mis0 = no misassignments relative to the SGC/mis1 = 1 misassignment, and so on. Pmut = 0.00975, Pdecay = 0.00975, Pinit = 0.150, Pfus = 0.00200, Ptab1 = 0.08.

for fused code evolution: one or two functions can have been assigned for no specific reason.

## DISCUSSION

### The Standard Genetic Code

Because all modern Terran organisms possess the SGC or a close relative, the most economical hypothesis is that the SGC existed in the last common ancestor. The SGC's origin has therefore attracted an immense literature that defies concise summary. However, a brief synopsis follows, to put present findings in context.

### Optimization of the SGC

A substantial quantitative literature exists on SGC optimization, much of it initiated by the finding that code structure appears to effectively reduce destructive effects of likely mutational or translational errors (Freeland and Hurst 1998). However, the SGC is only partially optimized (Novozhilov et al. 2007), standard optimization routes require many steps (Massey 2010), apparent optimization to errors readily occurs as a by-product of other goals (Massey 2008; Błażej et al. 2018), and full optimization is not physically plausible (Yarus 2021b). Thus, no code optimization exists in c3-lCw, other than that intrinsic to code fusion.

### Late Crick wobble (lCw)

Accurate third-codon-position wobble (Crick 1966) is unlikely to be a primordial form of genetic coding. Compare modern wobble: tRNA–rRNA interaction on the modern ribosome is extensive, spanning both tRNA molecules and distant regions in both large rRNAs (Moazed and Noller 1986, 1989). Some of these contacts appose rRNA nucleotides with codon-anticodon triplets to check their conformations (Ogle et al. 2001). Such checks determine that the first two base pairs are Watson–Crick (Demeshkina et al. 2012), as well as whether wobble positions lie within the multiple conformations allowed for normal, tautomeric and charged wobble pairs (Westhof et al. 2019). Such sophisticated three-dimensional error checking is unlikely for initial encoding, but could evolve later in coding history. Thus, here coding is assumed to involve normal base pairing until a later time when wobble becomes possible, probably in a ribonucleopeptide proto-ribosome. Given that Crick wobble readily fills the coding table, simplified Crick wobble (using only standard nucleotides, Yarus 2021b) is assumed to be adopted quickly throughout the early code, once evolved. Others have also treated wobble as a significant later coding development (Lei and Burton 2021).

### Summary of c3-lCw

Fusion joins partial codes which have diverse encodings. Fusion must occur frequently if fusion is to alter code distributions (Fig. 1), because even a single unfused coding table can evolve to resemble the SGC (Fig. 3C; Yarus 2021b). Fusion profoundly alters coding environments (Fig. 2). Fusions will likely be undirected, because fused codes cannot express a phenotype until after fusion itself. Thus, fusions that expand coding toward the SGC will be accompanied by those that do not—and incompatible coding annihilates both participants (Figs. 2, 7).

A fusing population shrinks as fusion expands. Here, the major long-term fate of codes is entry into fusions: effective, innocuous or disastrous. Only 9.4% of ≈61,000 codes that once existed survive in this late environment (750 passages, Fig. 2).

Such losses change the code population (Figs. 3, 4). A fusing population shows a crescendo of individuals (Fig. 3A) with near-complete codes (≥20 functions) that also resemble majority encoding (zero, one, two, or three differences from the SGC). Crescendo is a specific fusion product, absent for similar coexisting multiple codes lacking fusion (Fig. 3B). In addition, SGC-like codes are fewer for a system in which no similar fusion partners exist (Fig. 3C). Moreover, competence is delayed in both nonfusing systems (Fig. 3). Code fusions will be more readily selected, presenting better codes sooner, more frequently, in a smaller population and for a longer time.

Crescendo competence has a clear source: misassignment is suppressed (Fig. 4A). Variant codes are selectively extinguished in unproductive fusion attempts (Figs. 2, 6). A fused population has fewer erroneous random assignments (Fig. 4B), as well as fewer error-prone captures of related triplets (Fig. 4C). This requires only that selection favor codons with unique meanings. Thus, rise of a dominant encoding will be only slightly dependent on evolutionary details. Reliance on specific molecular mechanisms is minimized here because, in part, ancient coding machineries are still uncertain. C3-iCw fusion advantages would likely appear similarly for any encoding in which parts of the coding table can develop separately.

Environments evolving via code fusions produce competent codes more frequently (Yarus 2021b) than prior models. For example, ≈1% to ≈11% of the environments in Figure 3A have ≤3 misassignments with ≥20 functions—these frequencies seemingly imply easy selection for superior SGC translation, especially given a planet-sized sample. Or more specifically, exact (mis0) SGC codes exist in >1 in 1000 environments (Prand = 0.05, Fig. 5), though only selection intrinsic to fusion has been applied. For this SGC-identical class, the only selection required is harvesting exact SGC's among other codes. This is distribution fitness (Yarus 2021b), where a distributed property is selected from an

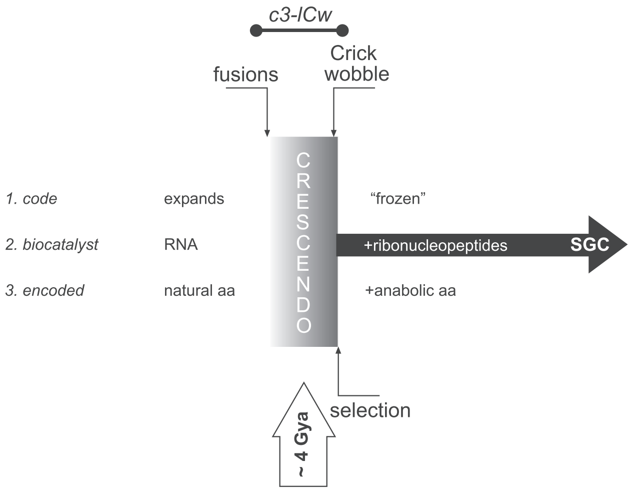


**FIGURE 6.** A coding crescendo separates two early code eras. Three numbered definitions for divided early code evolution (see text) are interpreted as outcomes of c3-lCw. Initiation of code fusion ("fusions"), evolution of simple wobble ("Crick wobble") and selection of the SGC ("selection") are marked relative to the crescendo (Fig. 3A). Here, wobble evolves just before SGC selection; most simply, wobble existed in the nascent SGC. However, simple Crick wobble could have emerged later in the ribonucleopeptide era. The last common ancestor lies off to the right of Figure 6. The c3-lCw's approximate time before the present (upward arrow) reflects that of the most ancient fossil biota (see text).

excellent but atypical minority. But, c3-lCw self-refinement still requires accurate assignments (Yarus 2021c), plausibly estimated as ≤10% random (Fig. 5).

## The surprising crescendo

Simple assumptions (fusions between incompatible partial codes are deleterious) yield striking code convergence (Figs. 3, 4). Convergence suppresses deviations of different origins (Fig. 4A–C), and thus is likely to be broadly applicable. Most significantly, complete and accurate codes accumulate continuously during prevalent fusion (Fig. 3A). Particular accurate codes present at one time (Fig. 3A) differ from those earlier and later. Varying SGC-like codes potentially increase until selection of a superior one. It is difficult to imagine a more effective SGC search.

## C3-lCw is consistent with previous conjecture

Fusion was initially suggested in order to allow SGC incorporation of different forms of ordered assignments (Yarus 2021c); for example, related amino acids in the same coding table column or row. It was also suggested that code fusion could both speed appearance of the SGC and allow it to appear in a smaller population—predictions borne out in Figure 2 and Figure 3.

## Three definitions for two coding eras

In several important ways, SGC history can be divided into two eras, each with its own evolutionary rules (Knight et al. 2001).

### First definition: code expansion versus code stability

The focus here is the early period of expansion of the code to its present scope, presumably selected via the ability of enlarged (more complete) codes to encode more competent peptides (Sengupta and Higgs 2015). The implied contrast is with a later period, after substantial completion of the SGC, when the code is approximately "frozen" (Crick 1968) because it must conserve a highly evolved prior proteome (Ardell and Sella 2002). However, even the later code evolves to some extent (Jukes and Osawa 1993), perhaps selectively changing late-evolved functions, like termination (Yarus 2021a).

### Second definition: RNA versus ribonucleopeptide agents

The first definition just above is approximately echoed in the distinction between an early era resting solely on the capabilities of RNA (Gilbert 1986) and a later era of ribonucleopeptide agents (Fig. 6). This second transition must lie somewhere near Crick's freezing point because aminoacyl-RNA synthetases themselves are very complex proteins (with multiple specific substrate sites, performing multiple catalyzes). A modern variety of such catalysts is only plausible when the code has become strongly constrained by extensive prior encoding.

### Third definition: primordial amino acids versus those from anabolism

Division into early and late eras is further reinforced by the prevalent belief that coding was initiated with readily available natural amino acids (Miller 1953), perhaps those most easily chemically synthesized under primitive conditions (Higgs and Pudritz 2009). The earliest amino acids are frequently specified as G, A, D, and V (Ikehara 2009), all encoded by GNN codons (N is any nucleotide) in the final SGC (Higgs 2009). Later more complex metabolism permitted the addition of amino acids derived from evolved metabolites (Wong 1975; Taylor and Coates 1989; Di Giulio 2008). This third division is closely related to those above: while coding likely began on RNA (Yarus 2017), synthesis of the first encoded peptides provides new molecules with novel conformations and chemical groups for intermolecular interactions. Thus, in the period before freezing (first definition), early codes can rapidly expand their interactions by making peptides that can enhance further code evolution. Progress toward codes so competent that they have hindered their own further evolution (first definition) is speeded by the appearance of structured riboucleopeotides (second definition). These might have been noncovalent (Carter 2015), covalent ribozymic (Turk et al. 2011), or chemical (Müller et al. 2022); ribonucleopeptide abilities are an evolutionary subject of great experimental interest. Finally, a near-complete SGC arises in the era of the ribonucleopeptide translation apparatus (Fig. 6). This code participates in the biochemistry of the last common ancestor, including its varied assortment of specific aminoacyl-RNA synthetases, required to encode protein catalysts for a complex metabolism (Ribas de Pouplana 2020; Xavier et al. 2021).

## All two-era definitions may rely on the crescendo

Figure 6 suggests that c3-lCw might delimit all eras. During the c3-lCw, the code became so complex it might be called frozen; the c3-lCw is also when capable ribonucleopeptide catalysts become possible, and the c3-lCw also inaugurates the first epoch when an almost modern set of nucleopeptide monomers might exist.

Current evidence suggests that the most ancient organisms took the form of close laminations, with different organisms densely layered within 3.43 Gya stromatolites (Allwood et al. 2006), or compact bundles of filaments 3.75 to 4.28 Gya in seafloor jasper (Papineau et al. 2022). Such colonial populations encourage association of cells of different origins and competency, and potentially encourage fusion of their codes. Given these microscopic fossils, c3-lCw might have occurred ≈4 billion years ago, as proposed in Figure 6.

## Late assignments to late amino acids

SGC coding triplets likely include sequences extracted from ancient RNA binding sites for cognate amino acids (Yarus and Christian 1989; Rodin et al. 2011; for review, see Yarus 2017). Such RNA-amino acid interactions probably underlie assignments in the earliest coding tables. Initial partial codes can differ, but then converge to the SGC as suggested here (Fig. 6). An objection sometimes offered (e.g., Koonin and Novozhilov 2017) is that metabolically complex, late-appearing amino acids like arginine (Janas et al. 2010) and tryptophan (Majerfeld et al. 2010) are among amino acids associated with triplet-containing RNA sites. But likely ancient amino acids like isoleucine also show prominent cognate triplets (Lozupone et al. 2003). Moreover, assignment of RNA triplets from binding sites would probably continue in a later ribonucleopeptide era (Fig. 6). RNA sites for later, complex amino acids like arginine (Yarus and Christian 1989) would be used when advantageous. So, ribonucleopeptide catalysts could expand anabolism to complex amino acids, while their SGC assignments still utilized fits to RNA binding sites.

## Fine-tuning wobble

Late Crick wobble is implemented here (Figs. 6, 7), but plays little part in this discussion of code capabilities. Completeness (all functions encoded) is the usual criterion for code progress; completeness is unaffected by simplified Crick wobble, which only extends existing assignments (Yarus 2021b, 2021d).

But modern coding goes beyond simple Crick wobble. For example: there are three isoleucine codons; AUU, AUC, and AUA. Two, AUU/AUC, are accessible by simple wobble. But standard bases should not wobble to read AUA without also pairing with AUG, a Met codon. Crick accommodated isoleucine (Crick 1966) by base modification, deaminating A to I (inosine) to pair with U, C, and wobble A. Bacteria instead modify tRNA$^{\mathrm{Ile}}$ with L-lysine to make lysidine at the anticodon's wobble position. Lysidine-modified tRNA can pair with AUA specifically (Nakanishi et al. 2009). Other tRNA$^{\mathrm{Ile}}$ anticodon arm modifications may also aid AUA translation by tRNA$^{\mathrm{Ile}}$ (Köhrer et al. 2014). Similarly, subtle anticodon refinements for accurate wobble have been reviewed (Grosjean and Westhof 2016): complex amino acid substrates (Higgs and Pudritz 2009) and complex anticodon arm modifications enhanced simple wobble during the ribonucleopeptide era (Fig. 6). Accordingly, much code history has occurred after the crescendo.

## HGT and code universality

Vetsigian et al. (2006) expresses Carl Woese's conviction that early HGT (horizontal gene transfer) was a crucial "innovation-sharing protocol." Only creatures possessing similar genetic codes could share innovations evolved independently. Thus, communal innovation-sharing selected code uniformity before the unified genome and efficient vertical inheritance evolved. A learning model confirms that mutual genetic intelligibility via HGT, without vertical inheritance, could have universalized the SGC (Froese et al. 2018). Present work suggests that when earlier partial codes prevailed, HGT would speed the assembly of, purify the population of, and facilitate the selection of, SGC-like codes.

# MATERIALS AND METHODS

The program that repeatedly evolves coding environments and reports code properties was written using the integrated development environment in Lazarus v. 2.20RC1 in its console mode, with the free Pascal FPC 3.2.2 compiler. Compiled mechanisms were run on a Dell XPS computer under 64-bit Microsoft Windows

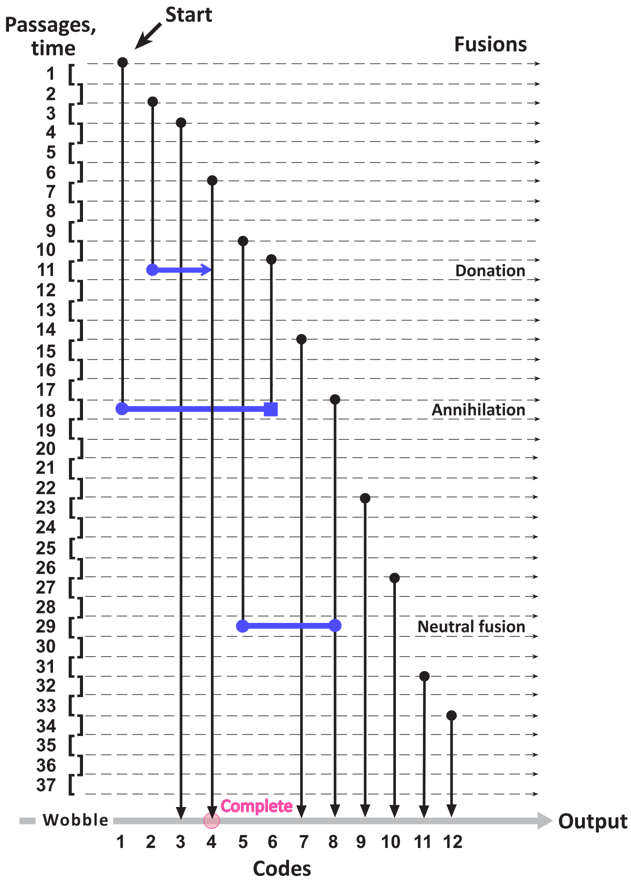


**FIGURE 7.** Code evolution in one simplified environment. For explanatory purposes, Figure 7 shows only 37 passages (environments can have thousands) and 12 codes (environments can have hundreds). Passages are time for one event in coding evolution: individual passages vary stochastically, but are shown similarly for clarity. However, the mean passage is defined, marking time reasonably precisely. Each environment begins with a single table, labeled "Start." New tables appear in Figure 7 with constant probability per passage (Ptab), starting with a single arbitrary assignment (Pinit). At each passage, all current tables evolve by one step: a new assignment (Pinit), random (Prand) or SGC-like (1—Prand), an assignment decay (Pdecay), or capture of an unassigned codon one mutation distant (Pmut). As soon as multiple tables arise, they can fuse with {probability Pfus/passage} × {number of codes − 1}. Three kinds of fusions exist, each decreasing total codes. The vanishing code can contribute assignments to a recipient ("Donation"). The vanishing code can be lost, along with its incompatible recipient ("Annihilation"). The vanishing code can cause no change if all its assignments already exist in the recipient ("Neutral fusion"). An environment is completed ("Complete") at a set time, or when a code with desired properties arises (translucent red circle), such as encoding ≥20 assigned functions. Wobble (late Crick wobble, lCw) evolves later (Yarus 2021b), after fundamental assignments are made. The program usually reports ("Output") averages of all codes, or alternatively, properties of best codes (e.g., most complete: translucent red circle). In Figure 7, this best code (Code 4) has evolved in part by receiving assignments from fusion.

10, 2.9 GHz on an Intel Core i9-8950HK CPU, using 32 GB of RAM.

The source used for all present calculations is Ctable20k.pas, available on request from the author. Results from the program, as tab-delimited files, were passed to Microsoft Excel 2016 for analysis and graphics. An example of spreadsheet analysis is also available on request.

Time is measured in cycles through code evolution, called passages. Passages, and other details of programmed action, are depicted in Figure 7. Assignment, decay and codon capture, which occur mutually exclusively during a passage through each single code, have been discussed previously (Yarus 2021b). Multiple codes and fusion are introduced in Figures 1–5 above.

## ACKNOWLEDGMENTS

Thanks to Norm Pace for an informative discussion of stromatolites, and to Bill McClain for helping to clarify a draft manuscript.

*Received May 25, 2022; accepted July 18, 2022.*

## REFERENCES

Aggarwal N, Bandhu AV, Sengupta S. 2016. Finite population analysis of the effect of horizontal gene transfer on the origin of an universal and optimal genetic code. *Phys Biol* **13:** 036007. doi:10.1088/1478-3975/13/3/036007

Allwood AC, Walter MR, Kamber BS, Marshall CP, Burch IW. 2006. Stromatolite reef from the early archaean era of Australia. *Nature* **441:** 714–718. doi:10.1038/nature04764

Ardell DH, Sella G. 2002. No accident: genetic codes freeze in error-correcting patterns of the standard genetic code. *Philos Trans R Soc Lond B Biol Sci* **357:** 1625–1642. doi:10.1098/rstb.2002.1071

Błażej P, Wnętrzak M, Mackiewicz D, Mackiewicz P. 2018. Optimization of the standard genetic code according to three codon positions using an evolutionary algorithm. *PLoS One* **13:** e0201715. doi:10.1371/journal.pone.0201715

Burroughs AM, Aravind L. 2019. The origin and evolution of release factors: implications for translation termination, ribosome rescue, and quality control pathways. *Int J Mol Sci* **20:** 1981. doi:10.3390/ijms20081981

Carter CW. 2015. What RNA World? Why a peptide/RNA partnership merits renewed experimental attention. *Life Basel Switz* **5:** 294–320. doi:10.3390/life5010294

Crick FH. 1966. Codon-anticodon pairing: the wobble hypothesis. *J Mol Biol* **19:** 548–555. doi:10.1016/S0022-2836(66)80022-0

Crick FHC. 1968. The origin of the genetic code. *J Mol Biol* **38:** 367–379. doi:10.1016/0022-2836(68)90392-6

Demeshkina N, Jenner L, Westhof E, Yusupov M, Yusupova G. 2012. A new understanding of the decoding principle on the ribosome. *Nature* **484:** 256–259. doi:10.1038/nature10913

Di Giulio M. 2008. An extension of the coevolution theory of the origin of the genetic code. *Biol Direct* **3:** 37. doi:10.1186/1745-6150-3-37

Duchêne A-M, Pujol C, Maréchal-Drouard L. 2009. Import of tRNAs and aminoacyl-tRNA synthetases into mitochondria. *Curr Genet* **55:** 1–18. doi:10.1007/s00294-008-0223-9

Freeland SJ, Hurst LD. 1998. The genetic code is one in a million. *J Mol Evol* **47:** 238–248. doi:10.1007/PL00006381

Froese T, Campos JI, Fujishima K, Kiga D, Virgo N. 2018. Horizontal transfer of code fragments between protocells can explain the origins of the genetic code without vertical descent. *Sci Rep* **8:** 3532. doi:10.1038/s41598-018-21973-y

Gilbert W. 1986. The RNA world. *Nature* **319:** 618. doi:10.1038/319618a0

Grosjean H, Westhof E. 2016. An integrated, structure- and energy-based view of the genetic code. *Nucleic Acids Res* **44:** 8020–8040. doi:10.1093/nar/gkw608

Higgs PG. 2009. A four-column theory for the origin of the genetic code: tracing the evolutionary pathways that gave rise to an optimized code. *Biol Direct* **4:** 16. doi:10.1186/1745-6150-4-16

Higgs PG, Pudritz RE. 2009. A thermodynamic basis for prebiotic amino acid synthesis and the nature of the first genetic code. *Astrobiology* **9:** 483–490. doi:10.1089/ast.2008.0280

Ikehara K. 2009. Pseudo-replication of [GADV]-proteins and origin of life. *Int J Mol Sci* **10:** 1525–1537. doi:10.3390/ijms10041525

Janas T, Widmann JJ, Knight R, Yarus M. 2010. Simple, recurring RNA binding sites for L-arginine. *RNA* **16:** 805–816. doi:10.1261/rna.1979410

Jukes T, Osawa S. 1993. Evolutionary changes in the genetic code. *Comp Biochem Physiol B* **106:** 489–494. doi:10.1016/0305-0491(93)90122-L

Knight RD, Freeland SJ, Landweber LF. 2001. Rewiring the keyboard: evolvability of the genetic code. *Nat Rev Genet* **2:** 49–58. doi:10.1038/35047500

Köhrer C, Mandal D, Gaston KW, Grosjean H, Limbach PA, RajBhandary UL. 2014. Life without tRNA$^{Ile}$-lysidine synthetase: translation of the isoleucine codon AUA in *Bacillus subtilis* lacking the canonical tRNA$_2^{Ile}$. *Nucleic Acids Res* **42:** 1904–1915. doi:10.1093/nar/gkt1009

Koonin EV, Novozhilov AS. 2017. Origin and evolution of the universal genetic code. *Annu Rev Genet* **51:** 45–62. doi:10.1146/annurev-genet-120116-024713

Lei L, Burton ZF. 2021. Evolution of the genetic code. *Transcription* **12:** 28–53. doi:10.1080/21541264.2021.1927652

Lozupone C, Changayil S, Majerfeld I, Yarus M. 2003. Selection of the simplest RNA that binds isoleucine. *RNA* **9:** 1315–1322. doi:10.1261/rna.5114503

Majerfeld I, Chocholousova J, Malaiya V, Widmann J, McDonald D, Reeder J, Iyer M, Illangasekare M, Yarus M, Knight R. 2010. Nucleotides that are essential but not conserved; a sufficient L-tryptophan site in RNA. *RNA* **16:** 1915–1924. doi:10.1261/rna.2220210

Massey SE. 2008. A neutral origin for error minimization in the genetic code. *J Mol Evol* **67:** 510–516. doi:10.1007/s00239-008-9167-4

Massey SE. 2010. Searching of code space for an error-minimized genetic code via codon capture leads to failure, or requires at least 20 improving codon reassignments via the ambiguous intermediate mechanism. *J Mol Evol* **70:** 106–115. doi:10.1007/s00239-009-9313-7

Mathew DC, Luthey-Schulten Z. 2008. On the physical basis of the amino acid polar requirement. *J Mol Evol* **66:** 519–528. doi:10.1007/s00239-008-9073-9

Miller SL. 1953. Production of amino acids under possible primitive earth conditions. *Science* **117:** 528–529. doi:10.1126/science.117.3046.528

Moazed D, Noller HF. 1986. Transfer RNA shields specific nucleotides in 16S ribosomal RNA from attack by chemical probes. *Cell* **47:** 985–994. doi:10.1016/0092-8674(86)90813-5

Moazed D, Noller HF. 1989. Interaction of tRNA with 23S rRNA in the ribosomal A, P, and E sites. *Cell* **57:** 585–597. doi:10.1016/0092-8674(89)90128-1

Müller F, Escobar L, Xu F, Węgrzyn E, Nainytė M, Amatov T, Chan C-Y, Pichler A, Carell T. 2022. A prebiotically plausible scenario of an RNA-peptide world. *Nature* **605:** 279–284. doi:10.1038/s41586-022-04676-3

Nakanishi K, Bonnefond L, Kimura S, Suzuki T, Ishitani R, Nureki O. 2009. Structural basis for translational fidelity ensured by transfer RNA lysidine synthetase. *Nature* **461:** 1144–1148. doi:10.1038/nature08474

Novozhilov AS, Wolf YI, Koonin EV. 2007. Evolution of the genetic code: partial optimization of a random code for robustness to translation error in a rugged fitness landscape. *Biol Direct* **2:** 24. doi:10.1186/1745-6150-2-24

Ogle JM, Brodersen DE, Clemons WM, Tarry MJ, Carter AP, Ramakrishnan V. 2001. Recognition of cognate transfer RNA by the 30S ribosomal subunit. *Science* **292:** 897–902. doi:10.1126/science.1060612

Papineau D, She Z, Dodd MS, Iacoviello F, Slack JF, Hauri E, Shearing P, Little CTS. 2022. Metabolically diverse primordial microbial communities in Earth's oldest seafloor-hydrothermal jasper. *Sci Adv* **8:** eabm2296. doi:10.1126/sciadv.abm2296

Ribas de Pouplana L. 2020. The evolution of aminoacyl-tRNA synthetases: from dawn to LUCA. *Enzymes* **48:** 11–37. doi:10.1016/bs.enz.2020.08.001

Rodin AS, Szathmáry E, Rodin SN. 2011. On origin of genetic code and tRNA before translation. *Biol Direct* **6:** 14. doi:10.1186/1745-6150-6-14

Sengupta S, Higgs PG. 2015. Pathways of genetic code evolution in ancient and modern organisms. *J Mol Evol* **80:** 229–243. doi:10.1007/s00239-015-9686-8

Taylor FJ, Coates D. 1989. The code within the codons. *BioSystems* **22:** 177–187. doi:10.1016/0303-2647(89)90059-2

Turk RM, Illangasekare M, Yarus M. 2011. Catalyzed and spontaneous reactions on ribozyme ribose. *J Am Chem Soc* **133:** 6044–6050. doi:10.1021/ja200275h

Vetsigian K, Woese C, Goldenfeld N. 2006. Collective evolution and the genetic code. *Proc Natl Acad Sci* **103:** 10696–10701. doi:10.1073/pnas.0603780103

Westhof E, Yusupov M, Yusupova G. 2019. The multiple flavors of GoU pairs in RNA. *J Mol Recognit JMR* **32:** e2782. doi:10.1002/jmr.2782

Woese CR. 1965. Order in the genetic code. *Proc Natl Acad Sci* **54:** 71–75. doi:10.1073/pnas.54.1.71

Wong JT-F. 1975. A co-evolution theory of the genetic code. *Proc Natl Acad Sci* **72:** 1909–1912. doi:10.1073/pnas.72.5.1909

Xavier JC, Gerhards RE, Wimmer JLE, Brueckner J, Tria FDK, Martin WF. 2021. The metabolic network of the last bacterial common ancestor. *Commun Biol* **4:** 413. doi:10.1038/s42003-021-01918-4

Yarus M. 2017. The genetic code and RNA-amino acid affinities. *Life* **7:** 13. doi:10.3390/life7020013

Yarus M. 2021a. Crick Wobble and superwobble in standard genetic code evolution. *J Mol Evol* **89:** 50–61. doi:10.1007/s00239-020-09985-7

Yarus M. 2021b. Evolution of the standard genetic code. *J Mol Evol* **89:** 19–44. doi:10.1007/s00239-020-09983-9

Yarus M. 2021c. Fitting the standard genetic code into its triplet table. *Proc Natl Acad Sci* **118:** e2021103118. doi:10.1073/pnas.2021103118

Yarus M. 2021d. Optimal evolution of the standard genetic code. *J Mol Evol* **89:** 45–49. doi:10.1007/s00239-020-09984-8

Yarus M, Christian EL. 1989. Genetic code origins. *Nature* **342:** 349–350. doi:10.1038/342349b0

Zhou Z, Liu Y, Li M, Gu J-D. 2018. Two or three domains: a new view of tree of life in the genomics era. *Appl Microbiol Biotechnol* **102:** 3049–3058. doi:10.1007/s00253-018-8831-x