Journal of Molecular Evolution (2021) 89:19–44
https://doi.org/10.1007/s00239-020-09983-9

ORIGINAL ARTICLE

Evolution of the Standard Genetic Code

Michael Yarus[1]

Received: 11 July 2020 / Accepted: 11 December 2020 / Published online: 24 January 2021

Abstract

A near-universal Standard Genetic Code (SGC) implies a single origin for present Earth life. To study this unique event, I compute paths to the SGC, comparing different plausible histories. Notably, SGC-like coding emerges from traditional evolutionary mechanisms, and a superior route can be identified. To objectively measure evolution, progress values from 0 (random coding) to 1 (SGC-like) are defined: these measure fractions of random-code-to-SGC distance. Progress types are **spacing/distance/d**elta **P**olar **R**equirement, detecting space between identical assignments/mutational distance to the SGC/chemical order, respectively. The coding system is based on selected RNAs performing aminoacyl-RNA synthetase reactions. Acceptor RNAs exhibit SGC-like Crick wobble; alternatively, non-wobbling triplets uniquely encode 20 amino acids/start/stop. Triplets acquire 22 functions by stereochemistry, selection, coevolution, or at random. Assignments also propagate to an assigned triplet's neighborhood via single mutations, but can also decay. A vast code universe makes futile evolutionary paths plentiful. Thus, SGC evolution is critically sensitive to disorder from random assignments. Evolution also inevitably slows near coding completion. The SGC likely avoided these difficulties, and two suitable paths are compared. In *late wobble*, a majority of non-wobble assignments are made before wobble is adopted. In *continuous wobble*, a uniquely advantageous early intermediate yields an ordered SGC. Revised coding evolution (limited randomness, late wobble, concentration on amino acid encoding, chemically conservative coevolution with a chemically ordered elite) produces varied full codes with excellent joint progress values. A population of only 600 independent coding tables includes SGC-like members; a Bayesian path toward more accurate SGC evolution is available.

Keywords Coding table · Codon · Triplet · Evolution · Distribution fitness

Introduction

The Object of the Investigation

Figure 1a initiates analysis by depicting its goal. The figure contains the SGC, connecting codon triplets and standard abbreviations for encoded functions, like the 20 standard amino acids. Woese (1965) discovered that the chromatographic mobility of amino acids in organic heterocycle/water mixed solvents could be used to classify the amino acids in a way relevant to the genetic code. In particular, the dependence of chromatographic mobility on the mole fraction water in the mixed solvent, called the 'polar requirement,' has been attached in parentheses to the amino acid abbreviations in Fig. 1a. Here, polar requirements are not Woese's original chromatographic values, but these quantities were corrected (Mathew and Luthey-Schulten 2008) by molecular dynamics distribution studies, which can circumvent chromatographic

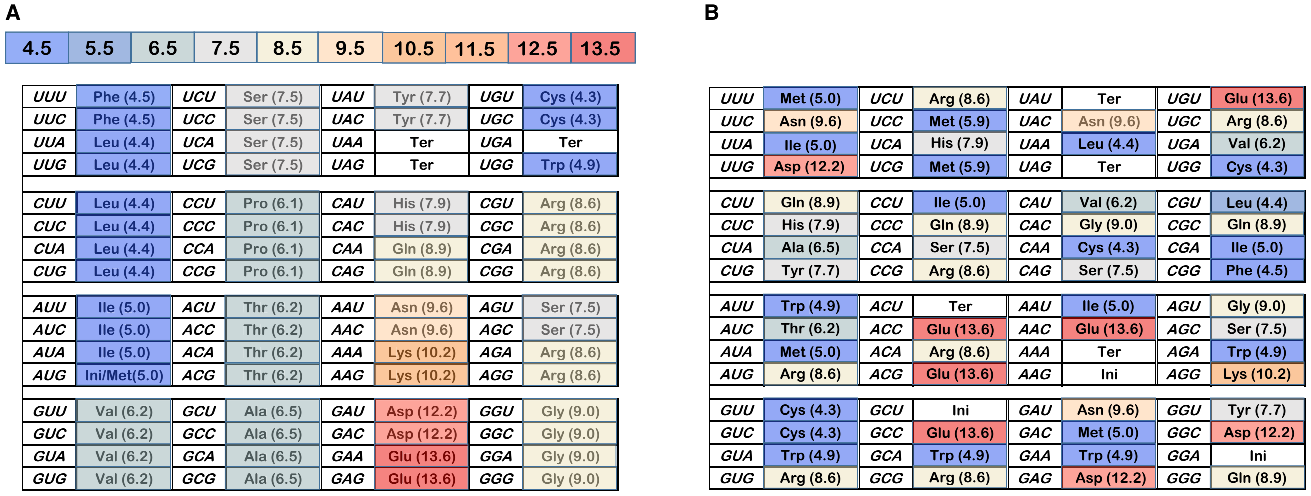
Handling editor: **Michelle Meyer**.

Work subsequent to this manuscript is also available online: Yarus (2021) Optimal evolution of the Standard Genetic Code, J Mol Evol https://doi.org/10.1007/s00239-020-09984-8, and Yarus (2021) Crick wobble and superwobble in Standard Genetic Code evolution, J Mol Evol https://doi.org/10.1007/s00239-020-09985-7.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00239-020-09983-9.

✉ Michael Yarus
yarus@colorado.edu

[1] Department of Molecular, Cellular and Developmental Biology, University of Colorado Boulder, Boulder, CO 80309-0347, USA



A

4.5	5.5	6.5	7.5	8.5	9.5	10.5	11.5	12.5	13.5

UUU	Phe (4.5)	*UCU*	Ser (7.5)	*UAU*	Tyr (7.7)	*UGU*	Cys (4.3)
UUC	Phe (4.5)	*UCC*	Ser (7.5)	*UAC*	Tyr (7.7)	*UGC*	Cys (4.3)
UUA	Leu (4.4)	*UCA*	Ser (7.5)	*UAA*	Ter	*UGA*	Ter
UUG	Leu (4.4)	*UCG*	Ser (7.5)	*UAG*	Ter	*UGG*	Trp (4.9)
CUU	Leu (4.4)	*CCU*	Pro (6.1)	*CAU*	His (7.9)	*CGU*	Arg (8.6)
CUC	Leu (4.4)	*CCC*	Pro (6.1)	*CAC*	His (7.9)	*CGC*	Arg (8.6)
CUA	Leu (4.4)	*CCA*	Pro (6.1)	*CAA*	Gln (8.9)	*CGA*	Arg (8.6)
CUG	Leu (4.4)	*CCG*	Pro (6.1)	*CAG*	Gln (8.9)	*CGG*	Arg (8.6)
AUU	Ile (5.0)	*ACU*	Thr (6.2)	*AAU*	Asn (9.6)	*AGU*	Ser (7.5)
AUC	Ile (5.0)	*ACC*	Thr (6.2)	*AAC*	Asn (9.6)	*AGC*	Ser (7.5)
AUA	Ile (5.0)	*ACA*	Thr (6.2)	*AAA*	Lys (10.2)	*AGA*	Arg (8.6)
AUG	Ini/Met(5.0)	*ACG*	Thr (6.2)	*AAG*	Lys (10.2)	*AGG*	Arg (8.6)
GUU	Val (6.2)	*GCU*	Ala (6.5)	*GAU*	Asp (12.2)	*GGU*	Gly (9.0)
GUC	Val (6.2)	*GCC*	Ala (6.5)	*GAC*	Asp (12.2)	*GGC*	Gly (9.0)
GUA	Val (6.2)	*GCA*	Ala (6.5)	*GAA*	Glu (13.6)	*GGA*	Gly (9.0)
GUG	Val (6.2)	*GCG*	Ala (6.5)	*GAG*	Glu (13.6)	*GGG*	Gly (9.0)

B

UUU	Met (5.0)	*UCU*	Arg (8.6)	*UAU*	Ter	*UGU*	Glu (13.6)
UUC	Asn (9.6)	*UCC*	Met (5.9)	*UAC*	Asn (9.6)	*UGC*	Arg (8.6)
UUA	Ile (5.0)	*UCA*	His (7.9)	*UAA*	Leu (4.4)	*UGA*	Val (6.2)
UUG	Asp (12.2)	*UCG*	Met (5.9)	*UAG*	Ter	*UGG*	Cys (4.3)
CUU	Gln (8.9)	*CCU*	Ile (5.0)	*CAU*	Val (6.2)	*CGU*	Leu (4.4)
CUC	His (7.9)	*CCC*	Gln (8.9)	*CAC*	Gly (9.0)	*CGC*	Gln (8.9)
CUA	Ala (6.5)	*CCA*	Ser (7.5)	*CAA*	Cys (4.3)	*CGA*	Ile (5.0)
CUG	Tyr (7.7)	*CCG*	Arg (8.6)	*CAG*	Ser (7.5)	*CGG*	Phe (4.5)
AUU	Trp (4.9)	*ACU*	Ter	*AAU*	Ile (5.0)	*AGU*	Gly (9.0)
AUC	Thr (6.2)	*ACC*	Glu (13.6)	*AAC*	Glu (13.6)	*AGC*	Ser (7.5)
AUA	Met (5.0)	*ACA*	Arg (8.6)	*AAA*	Ter	*AGA*	Trp (4.9)
AUG	Arg (8.6)	*ACG*	Glu (13.6)	*AAG*	Ini	*AGG*	Lys (10.2)
GUU	Cys (4.3)	*GCU*	Ini	*GAU*	Asn (9.6)	*GGU*	Tyr (7.7)
GUC	Cys (4.3)	*GCC*	Glu (13.6)	*GAC*	Met (5.0)	*GGC*	Asp (12.2)
GUA	Trp (4.9)	*GCA*	Trp (4.9)	*GAA*	Trp (4.9)	*GGA*	Ini
GUG	Arg (8.6)	*GCG*	Arg (8.6)	*GAG*	Asp (12.2)	*GGG*	Gln (8.9)

Fig. 1 **a** Top: Color coding for polar requirement in Fig. 1. Each number indicates the midpoint PR for that color. So the 10.5 box spans 10.01–11.0. **a** Bottom: The standard genetic code (SGC), with parenthetical polar requirements (Mathew and Luthey-Schulten 2008). The SGC has progress values: spacing = 1.0, distance = 1.0; dPR = 1.0. **b** A randomized genetic coding table; each triplet assigned to one of 22 functions by randomized number. Colors visually represent polar requirement, as in **a**. The example shown is representative of randomized coding tables: its progress values: spacing = 0.009, distance = − 0.017, dPR = − 0.195

artifacts, such as amino acids with affinity for a paper chromatographic support.

Woese pointed out (Woese et al. 1966) that the genetic code assigned similar codons to amino acids with similar polar requirements. In Fig. 1a, each triplet has been colored, with hydrophobic polar requirements blue, intermediate ones gray to beige, and very polar side chains red. The SGC is exceedingly highly ordered with respect to the polar requirement, with large coherent domains for hydrophobic, intermediate, and polar amino acids. The single isolated chemical domain is also the smallest; at the upper right, containing the unusual amino acids Cys and Trp. Chemical order spans the coding table, with its division into a few coherent regions especially striking. This coherence makes it obvious why the code's development can be accurately directed by maximizing similarity in polar requirements as a guide (Freeland and Hurst 1998b). This has been attributed to similar roles for chemically similar amino acids within proteins (but see below).

To illustrate extent of SGC order by contrast, Fig. 1b is a coding table that has none. Triplets were assigned using randomized numbers, then the table was colored using the polar requirement scheme of Fig. 1a. The distinctive, pervasive chemical order of the SGC is strikingly evident in the dissimilarity of Fig. 1a and b.

A Model for Calculations

To investigate SGC appearance, we desire the fewest, least specific assumptions, in order to maximally respect limited knowledge of the early code. They are as follows: there was an era in which 22 meanings (20 amino acids and start and stop signals) became assigned to 64 possible triplets. This era begins with the first triplet assignment, and ends with a near-fully assigned coding table that resembles the **S**tandard **G**enetic **C**ode (SGC). Meaningful average rates of coding assignment, which includes both enabling mutation and ensuing events that fix a new meaning, are assumed to exist ("Methods", Fig. 12).

These assumptions emphasize mode and kinetics during SGC approach, and de-emphasize mechanistic detail. Therefore, this analysis foregoes some kinds of analysis, to emphasize other kinds. I argue that new resolution of a likely route to the SGC results, without requiring still-unknown mechanistic detail.

Relations Between Identical and Similar Functions

Examination of triplets occupied by similar or identical amino acids in the SGC suggests regular relations between multiple assignments for similarly encoded functions.

Third Codon Position

As has long been evident (Woese 1965), third codon positions often vary without changing coding, producing XY **A/G**, XY **U/C**, XY **U/C/A** or XY **U/C/A/G** blocs with similar assigned functions and polar requirements. This is not likely due to mutational uniqueness in third-position triplet nucleotides, which presumably mutate as do other nucleotides. Instead, similarity is attributable to wobble

(Crick 1966), which assigns versatile base pairing to third codon positions, reading them by ambiguous pairing with the same molecule. So, the code easily expands to accommodate third-position mutational variation, immortalizing many such easy SGC expansions (Fig. 1a). SGC structure implies that contemporary acceptors wobbled because wobble (Crick 1966) assignments are frequent.

Such order extends to amino acids that are not identical, but similar chemically, judged by polar requirement. Whenever a code box containing XY **U/C/A/G** also contains different amino acids, the amino acids have similar polar requirements but varied chemistry. This is true for chemically varied amino acids: hydrophobics like Phe and Leu, weakly polars like Ser and Arg, and very polar side chains like Asp and Glu (Fig. 1a).

First Position

Less frequently, mutational variation in the first position appears to have been captured, for example, with identical residues as for Leu UU **A/G** and CU **A/G**, or similarly, Arg CG **A/G** and AG **A/G**. Again, vertical columns of the same color (similar PR) often join otherwise chemically different amino acids by first-position change. Gly-Arg, Tyr-His and Ser-Arg are examples (Fig. 1a).

Second Position

Least frequently, the SGC suggests capture of second-position variation for an identical function, the clearest possibility being UAA/UGA terminators. However, relations between chemically similar amino acids via second-position change are common in the SGC, as for Ser-Tyr (Fig. 1a).

The Formative Influence of Mutational Neighborhoods

These observations are consolidated by supposing that code evolution was guided by likely mutational pathways. A triplet with a given function might transfer function to a triplet related to it by single mutation. Thus, there are three possible triplets that might be captured at the first, second, and third triplet positions; nine possible captures in total. These nine changes comprise a triplet's "mutational neighborhood." When neighborhood mutations were readily accommodated, as at wobble positions, the code frequently expanded by that route.

Here, we simplify by assuming that all mutations are equally likely, though there is evidence that transitions (pyrimidine to pyrimidine and purine to purine) are more probable than transversions (purine to pyrimidine, or its reverse; Lehman and Joyce 1993; Vawter and Brown 1993; Collins and Jukes 1994; Kumar 1996).

A Plausible Primordial Acceptor RNA Model from Selection

Selection amplification for aa-RNA synthesis from its natural aa-adenylate precursor readily yields small aa-RNA-producing catalysts (Illangasekare et al. 1995). By selecting aa-RNA synthesis without requiring aminoacylation of an arbitrary 3′ sequence, such an RNA active center can be reduced to a 5-nt ribozyme aminoacylating a 4-nt substrate RNA (Chumachenko et al. 2009) with only 3 nucleotides conserved for aminoacyl transfer. Thus the natural aminoacyl-RNA precursor, an activated amino acid adenylate, is bound and its amino acid regiospecifically esterifies the terminal 2′ hydroxyl of a tetramer RNA within a tiny RNA active center (Yarus 2011). The dimensions of such a catalytic RNA pentamer are not large enough to surround an amino acid, and indeed the small aminoacylator is not amino acid specific (Turk et al. 2011).

Varied selection data show that sidechain-specific amino acid-binding RNAs exist, and require a minimum of 18–20 ribonucleotides (Yarus et al. 2005; Yarus 2017b). Thus, regiospecific aminoacyl transfer requires a surprisingly simple center with only three conserved ribonucleotides. Ribonucleotides therefore are unexpectedly proficient at *trans*-aminoacylation catalysis. In pronounced contrast, many more nucleotides would usually be required to add side chain specificity. Therefore, amino acid specificity is not expected in the very earliest, small aminoacylation catalysts (but see Illangasekare and Yarus 1999).

Accordingly, selection amplification suggests that the simplest, therefore earliest, ribozymic aminoacyl-RNA synthetase would catalyze RNA-specific acylation, via 3 or more specific base-pairs to an oligonucleotide acceptor, but would transfer multiple amino acids. The small aminoacyl-ribonucleotide product, using its pairing nucleotides, could also base pair relatively specifically with a subset of codons (Illangasekare and Yarus 2012), thus acting as a primordial anticodon. An aminoacyl-RNA would thereby associate its triplet codon(s) with a set of amino acid sidechains. Base pairing nucleotides that bind RNA substrate to ribozyme can be changed with only small effects on activity (Illangasekare and Yarus 2012). So, mutation of a base pairing, proto-anticodon nucleotide would allow the acceptor oligonucleotide to base pair with a new set of codons. New codon specificity therefore requires only a synthetase duplication and a single base-pairing mutation. Such mutating aminoacyl-RNAs associate their amino acids with neighboring triplet(s), the event here termed mutational capture (see "Methods" section, Fig. 12).

Further, ribonucleotides can be added to the small, unspecific aminoacylation active center. Extensions at both ribozyme and acceptor termini permit continued catalytic activity (Illangasekare and Yarus 2012; Xu et al. 2014). Such nucleotide additions might permit a new fold that allows amino acid sidechain specificity. For example, sidechain-proximal nucleotides potentially restrict large amino acids, making aminoacylation selective for small side chains. So, with two sequence changes (a proto-anticodon change and one proximal to the sidechain), previously untranslatable triplets might acquire a novel meaning, chemically related to a pre-existing assignment.

Aminoacyl-RNA Summary

Existing molecular data suggest a primitive manifold of specific acceptors, reading restricted codon groups, but using a single common aminoacyl transferase catalytic center, whose ribozyme can be as small as 5 ribonucleotides (Illangasekare and Yarus 2012). This RNA can be elaborated to add amino acid selectivity. Acquisition of new triplet specificities without losing aminoacylation activity permits such an aminoacyl transfer center to readily explore its triplet neighborhood, capturing the nine codons in its mutational neighborhood; that is, those a single mutation away.

Simplified Crick Wobble

Early coding must be minimal, independent of complex nucleotide modifications which can only arise later (Grosjean and Westhof 2016). To model wobble, I use a potentially primitive system (Crick 1966), requiring only natural nucleotides. In particular, third-position G:U wobble pairs are allowed. Acceptor (anticodon): coding (codon) pairs include A:U, G:U, G:C, U:G, U:A, and C:G. Table 1 lists these and allows visualization of mutational transitions, and therefore the evolutionary routes that simplified wobble coding most likely will follow. Thus, for example, one cannot assign XYA or XYC specifically; such functional triplets exist only as members of wobble pairs. If a wobble or non-wobble choice is made, as for codon XYU, wobble occurs with probability Pwob.

Table 1 The simplified Crick wobble system

Acceptor	$^{3'}$ **U**	$^{3'}$ **C**	$^{3'}$ **A**	$^{3'}$ **G**
Coding 1st	$^{5'}$ **AYZ**	$^{5'}$ **GYZ**	$^{5'}$ **UYZ**	$^{5'}$ **CYZ**
Acceptor	$^{3'}$ **U**	$^{3'}$ **C**	$^{3'}$ **A**	$^{3'}$ **G**
Coding 2nd	$^{5'}$ **YAZ**	$^{5'}$ **YGZ**	$^{5'}$ **YUZ**	$^{5'}$ **YCZ**
Acceptor	$^{3'}$ **U**	$^{3'}$ **C**	$^{3'}$ **A**	$^{3'}$ **G**
Coding 3rd	$^{5'}$ **YZA**	$^{5'}$ **YZG**	$^{5'}$ **YZU**	$^{5'}$ **YZC**
Coding 3rd	$^{5'}$ **YZG**			$^{5'}$ **YZU**

Y and **Z** are arbitrary nucleotides. Third acceptor complement position **G** may specify coding **C** or **U**; third acceptor complement position **U** may specify either coded **A** or **G** (Fig. 12). Coding triplets are permitted only one acceptor

However, Crick wobble is not clearly primordial. While G:U pairing itself is ancient, modern ribosomes use complex rRNA conformational changes to limit codon:anticodon complexes to third-position wobble (Moazed and Noller 1990; Ogle et al. 2001). Moreover, the tRNA anticodon loop-and-stem structure has a complex role in translational efficiency (Yarus 1982), including suppression of errors (Ledoux et al. 2009). Thus, it is plausible that simpler base pairing was primordial, and that evolutionary advances in both ribosomes and adaptor RNAs were required to make efficient and accurate third-position wobble (Table 1) possible. Thus, in these calculations, wobble sometimes appears later, after simpler codon:anticodon base pairing.

Quantitative Detection of Evolutionary Progress

To compare evolving coding tables, objective measurement of differences like those between Fig. 1a and b is essential. With the SGC (Fig. 1a) and the above discussion in mind, code order is measured using three progress indices.

Mean Mutational Spacing Between Identical Assignments (Spacing)

We are interested in grouping of identical functions because SGC coding occurs in compact groups (Fig. 1a). Progress toward this condensed goal is measured by counting mutations required to superpose triplets for identical functions (amino acids and start/stops). This distance (termed "spacing") is ≤ 3 mutations for every triplet comparison; and if 3 if all three coding nucleotides must be changed. Further, each pair of triplets must be counted only once, not duplicated by starting from both participants. In practice, it is useful to normalize distances for the number of pairs, calculating mean distance/triplet pair. Normalization makes spacing resilient when tables with varying numbers of unassigned triplets are compared. In 1000 random complete coding tables, identical functions are 2.284 (mean) ± 0.002 (sem) mutations apart. The SGC has a mean distance of 1.30 mutations between identical functions by the same criterion. Thus, spacing progress captures the SGC's exceptional compaction—tracking progress from random tables (spacing 2.284) toward the condensed SGC (spacing 1.30).

Distance from the SGC (Distance)

Progress to any code is of interest, but most particularly, progress toward the SGC. Distance to the SGC is quantified by totaling the total number of mutations required to move from triplets in a novel table to triplets for identical functions in the SGC. Again, only identical functions are compared, all possible pairs are counted once, and the result is normalized to yield the mean distance per triplet comparison. One

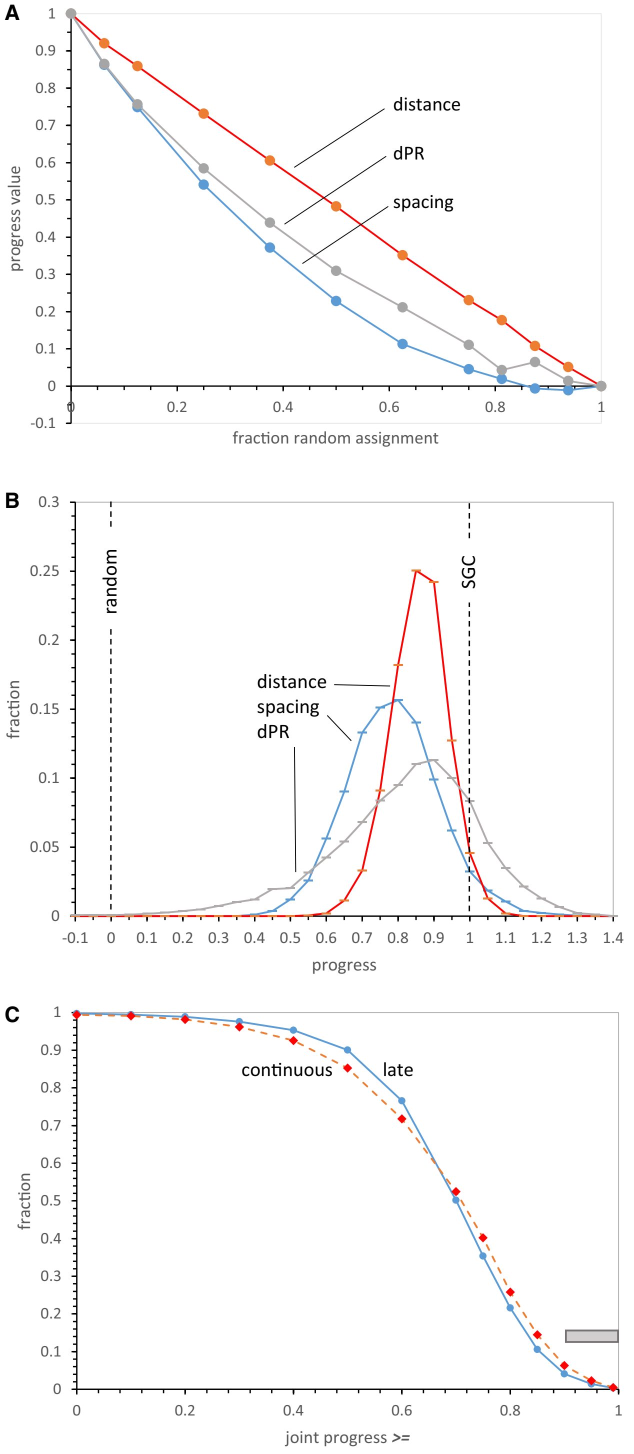


Fig. 2 **a** Progress values in randomly assembled coding tables. ▸ Computed mean spacing, distance, and dPR progress values for 250 full coding tables at each point, assembled with the abscissa's fraction of varied random assignments, complemented with SGC-like assignments for all other triplets. **b** Distributions of progress values with late wobble. Histograms of 10,000 late wobble evolutions to 20 encoded functions with $P_{rand} = 0.1$ and Coevo_PR assignments are shown. Lines mark randomized ("random") behavior and Standard Genetic Code ("SGC") behavior. $P_{init} = 0.6$, $P_{decay} = 0.04$, $P_{mut} = 0.04$. **c** Joint distributions of spacing, distance, and dPR progress values for continuous and late-wobbling code evolution. Data include that of **b**. The "SGC" line indicates all progress values are simultaneously ≥ 1. Gray bar marks the range for SGC-proximal coding tables; joint progress ≥ 0.9. 1000 continuous wobbling (to 175 passages) and 10,000 late-wobbling (to 20 encoded functions) evolutions were employed, with Coevo_PR assignments and $P_{rand} = 0.1$, $P_{init} = 0.6$, $P_{decay} = 0.04$, $P_{mut} = 0.04$, $P_{wob} = 0.5$

thousand independent completely random tables average 2.286 ± 0.002 mutations from the SGC (per pair) by this measure. Random spacing and distance are further clarified in "Methods" section.

Chemical Order (dPR)

The SGC shows exquisite, virtually comprehensive ordering of amino acids by polar requirement (colored areas, Fig. 1a). We quantitate chemical order by summing absolute polar requirement differences over all amino acid pairs in mutational neighborhoods, using corrected amino acid polar requirements (Mathew and Luthey-Schulten 2008) closely related to those measured chromatographically by Woese et al. (1966). Only neighborhood pairs that differ are counted and normalized for the number of comparisons. Thus, dPR does not overlap with spacing, dPR counts only non-identical residues. So, dPR specifically measures chemical grouping, not coding proximity. This normalized distance is 2.98 ± 0.01 per amino acid pair (in polar requirement units) for 1000 random tables versus 2.069 for the SGC, thereby allowing dPR to report chemical order (Fig. 1a). dPR is the only progress index that explicitly utilizes the notion of mutational neighborhood.

Indices of Coding Order: Progress Values

In order to make progress indices transparent indicators for SGC proximity, they are used in a form which does not require comparison to other numbers. This "progress value," is 0.0 for unordered, random coding tables and 1.0 when order equivalent to the SGC is attained. Thus progress from random coding to SGC order appears as decimal zero to one, respectively; progress value is the fraction of mutational or chemical distance to the SGC covered.

$$\text{spacing, distance, dPR progress} = \frac{\text{random index} - \text{system index}}{\text{random index} - \text{SGC index}}$$

Progress values can be < 0 or > 1 because systems can be more disperse than random coding tables or more frequently, more ordered than the SGC itself. Still, mean decimal spacing, distance, and dPR allow assessment of a calculation yielding tens of thousands of numbers, indicating whether greater or lesser mean SGC likeness was attained. Many such discriminations were the crux of the present inquiry. While progress might be differently defined, below these definitions do locate the SGC.

Progress Values Respond to Random Assignment

To clarify progress values, Fig. 2a plots mean spacing, distance, and dPR for groups of 250 full coding tables constructed with varied numbers of randomly chosen SGC assignments. Unassigned triplets were filled with random assignments with no set relation to the SGC. These coding table populations therefore are otherwise random, but have a specified fraction of SGC ordered assignments—the latter fraction is plotted across Fig. 2's *x*-axis. Distance progress is accurately proportional to the fraction of random triplet assignments, starting from the SGC at upper left. dPR and spacing progress are more sensitive to random assignment, declining to near-random values before all triplets are randomized. Spacing is most sensitive to random triplet intercalations, but all three progress indices respond progressively to small deviations from the SGC, rationalizing their use to assess SGC proximity.

Progress Values Have Varied Relations to SGC Order

In Fig. 2b is shown spacing, distance, and dPR progress for 10,000 coding tables evolved by late wobble, a code history we will later dissect in detail. All distributions are roughly symmetrical single peaks and well described by means and standard errors used here. Almost all evolved coding tables have progress value distributions highly shifted from random assignment (0.0), with means just below the SGC (1.0), indicating overall effectiveness for a late-wobbling evolution.

Full distributions in Fig. 2b also show that distance is the sharpest peak and finds the fewest codes at or exceeding SGC behavior. Spacing is broader and has an intermediate-size foot at or beyond the level of SGC grouping of identical functions. The broadest and least symmetrical distribution is for dPR and chemical order among closely related triplets. As one result, chemical order is often the most frequently evolved in this population.

SGC-Like Codes Require Three Upper-Tail Properties

Moreover, Fig. 2b shows that access to realistic coding tables is very sensitive to underlying coding order. Tables resembling the SGC are those simultaneously in the upper tails of three progress distributions. Fraction of evolved tables with three progress values near 1, the fraction with SGC order, will therefore vary rapidly and non-linearly when change in history shifts or spreads underlying spacing, distance, and dPR distributions (Fig. 2b), even slightly.

Thus, the joint distribution of progress values (Fig. 2c) quantitatively implements the present biological goal; a code that possesses SGC-like grouped assignments, SGC-related and chemically ordered (Fig. 1a). In Fig. 2c, the fraction of coding tables with joint progress greater than the abscissa value is calculated. For example, ≈ 50% of evolved coding tables have spacing, distance, and dPR simultaneously ≥ 0.7. More specifically, when the “vicinity-of-the-SGC” is desired, the region under the rightward gray bar will be utilized. In other words, those coding tables with spacing, distance, and chemical order simultaneously covering ≥ 90% of the distance from random codes to the SGC will be cited, to target discussion. Solid and dashed curves in Fig. 2c represent two ways to evolve a genetic code. Below, we will return to this difference between late and continuous wobble.

The Code Evolution Model Evolves Finished Codes

A biologist may be slightly interested in averaged behavior for coding table populations. Such an aggregate accurately follows underlying kinetic rules, but for example, never finishes a coding table, persisting forever in an average near steady state with unassigned triplets. In contrast, a coding table that evolves to a finished code is of immediate interest. Finished codes assign all 64 triplets (termed “full” tables) or encode all 22 functions (termed “complete” tables).

Coding history is computed (described in “Methods” section, Fig. 12) by following one coding table at a time. A triplet is randomly chosen. Subsequent events occur at random on the basis of probabilities for initial triplet assignment (P_{init}), mutational capture of a nearby triplet by existing assignments in an assigned triplet (P_{mut}) or assignment decay (P_{decay}). Using randomized numbers to choose chance events with specified probabilities, a chosen unassigned triplet can be allocated to one of 22 essential functions (P_{init}). If the randomly chosen triplet has already been assigned, it can capture new triplets for its function, or related functions, via neighborhood mutations (P_{mut}). Alternatively, its function can decay, with the triplet losing its previously assigned meaning, to become unassigned again (P_{decay}). Probabilities are chosen to limit outcomes to a total probability of 1.0. Repeating such chance events, only one of which occurs in a passage, ultimately builds a code with desired properties, for example, full (64-assignment) codes, or a complete (22 function) code. Repetition of such computation using different assumptions compares evolutionary routes by determining the probability of an SGC-like outcome.

Kinetics by Choosing a Series of Random Triplets

Because it is not usual to compare rates by performing a succession of random events, we first show that this yields expected kinetic behaviors. Figure 3 exhibits velocities for

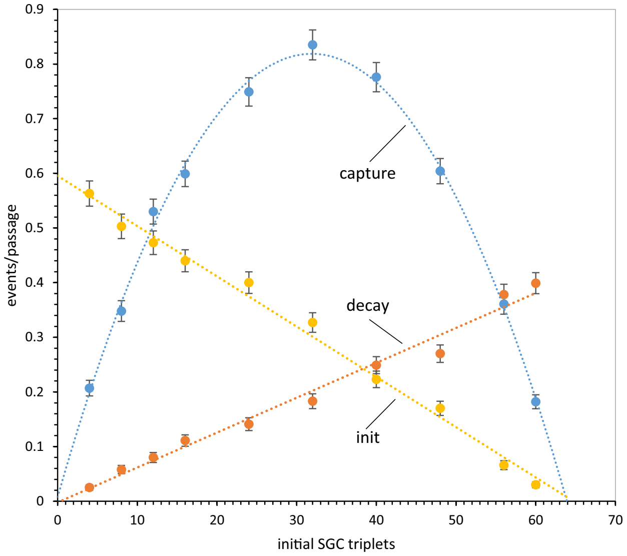


Fig. 3 Observed rates of initiation, decay, and mutational capture in coding tables with varied numbers of randomly chosen SGC triplets. Measured mean rates (events/passage ± sem) are shown for 1000 randomly composed non-wobbling tables. Fitted least squares dotted lines portray expected kinetic behavior for initiation, decay, and mutational capture. Decays and captures have been enlarged tenfold to visualize them on the same scale as initiations. $P_{mut} = 0.04$, $P_{decay} = 0.04$, $P_{init} = 0.6$, but $P_{rand} = 1.0$

initial triplet assignments, mutational captures, and assignment decays as a function of the number of assigned triplets (randomly chosen from the SGC in 1000 repetitions). First-order reactions should be proportional to reactant availability, second-order reactions to the product of two reactant availabilities.

Initiation

Initial assignment linearly declines in rate as triplets are filled with random SGC assignments. Initiation is a maximum with no triplets filled (at left, where the least squares line extrapolates to the complete table's $P_{init} = 0.6$), decreases linearly as triplets become assigned, and extrapolates near zero when 64 triplets are occupied, so that no initiation can exist. Thus, accurate first-order initiation is seen.

Decay

Assignment decay should also be first order in assigned triplets. It is zero at left (where there are no assignments to decay), increases linearly as assigned triplets increase, and extrapolates to a maximum reflecting the probability of table decay itself (0.04/passage) when 64 triplets are occupied, and therefore full probability of decay is expected. Thus, accurate first-order decay is observed.

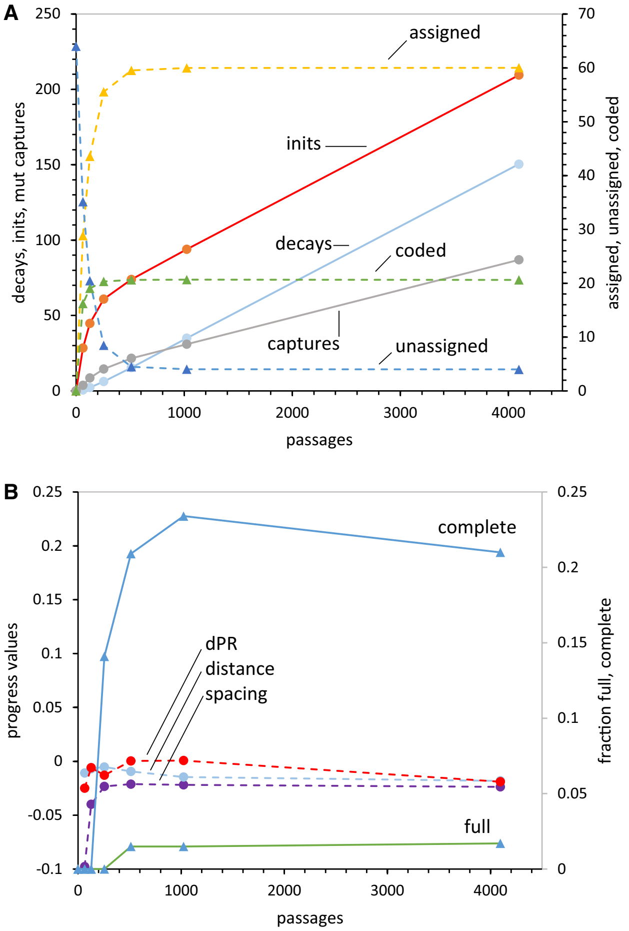


Fig. 4 **a** Kinetic evolution of non-wobbling coding tables. Means are shown for 1000 coding tables using random initial assignment and 0 ± 2 PR mutational capture through 4096 passages. Initiations, decays, and mutational captures are plotted, with triplets assigned, triplets unassigned, and functions coded on a secondary axis (rightward). Transition probabilities as in Fig. 3. **b** Progress and finished non-wobbling coding tables. Means of 1000 spacing, distance, and dPR progress values during the 4096 passages of **a** are plotted on the left ordinate, and the fraction of full and complete coding tables are on the right ordinate

Mutational Capture

Transfer of an assignment to a neighborhood triplet contrasts with initiation and decay, it requires an assigned triplet and an unassigned one, the latter to be captured for a similar assignment. Expansion of the code by mutational capture therefore should be second order, varying with the product (assigned*unassigned) triplets. The data show that the expected second-order maximum capture rate is observed when half, 32 triplets, are assigned. Moreover, the fitted rate extrapolates to zero both when assigned codons = 0 (at left) and also when unassigned codons = 0 (at right). So, mutational triplet capture behaves as a second-order reaction.

There is further quantitative support for this rate analysis in "Methods" section.

One might summarize Fig. 3 by saying that computer passages through an evolving coding table are proportional to time. But there is nevertheless a difference in definition. To respect this difference, durations are expressed in "passages" (computational transits of a nascent coding table) rather than "times."

Two Eras in a Nascent Coding Table

By putting off some details (of mutational capture mechanisms), coding table fates can now be computed. Figure 4a shows mean data for a population of 1000 coding tables without wobble (characteristics in legend), through their initial 4096 passages. There is an initial period of rapid change (≈ 0–200 passages), then a near steady state in which assigned and unassigned triplets change little. In that later era, total decays, initial assignments, and mutational captures increase at almost constant rates, but mean assigned/unassigned triplets are almost constant. Assignments are rapid initially (because many unassigned triplets exist), decays increase after a delay (in which assigned triplets accumulate), and mutational captures accelerate, then slow as requisite unassigned triplets become rare. Ultimately assignment events (initiation and mutational capture) and decay events balance, and a steady state emerges. In "Discussion" section we return to persistent unassigned triplets, and to stably incomplete coding, exemplified in Fig. 4a as a mean of 60 steadily assigned and 4 unassigned codons.

Figure 4b, however, shows new, finished code behavior. Both transient and near-steady-state behavior appear also for full and complete coding tables of biological interest. Full coding tables (all triplets assigned) appear after a delay and then are stable at about 0.016 of the population. Complete coding (22 encoded functions) both appears earlier and also is more abundant, ≈ 0.22 of all tables. This sequence reflects the fact that ≥ 22 events minimally complete a table, but ≥ 64 events are required to fill a table. Two distinct eras: early transient emergence and later stable codes, shape code evolution and accordingly, are discussed again below.

Sources for Steady-State Order

Now consider evolution of progress values. Figure 4b shows spacing, distance, and dPR progress with increasing (Fig. 4a) duration. Coding progress also has a steady state. Progress values are similar at all points in a population's history, once coding tables are substantially occupied. This is equally true for spacing, distance, and dPR. Thus progress is near-constant in time for tables with the same transformation probabilities (Fig. 4a, b). Finally, Fig. 4 evolution history employs random initiations and random later mutational captures. Figure 4b extensively documents the lasting random fate of such a table (that is; progress values ≈ 0) throughout 4096 passages.

Thus, to evolve an SGC we must include source(s) of order. Order comes from assignment of early triplets matching the SGC (as for stereochemical origins). Such initial stereochemistry will be insufficient, but its failure clarifies more successful code evolution.

Coding Tables with Initial SGC Assignments

Consider coding that begins with 16 randomly chosen SGC triplets. Using different sets of initial SGC triplets averages effects of particular dispositions. Figure 5 presents such average passages to varied levels of encoding, including ultimate completion at 22 encoded functions. The number of initial triplet assignments required to attain final levels of encoding (in addition to the initial 16) is also shown.

Because results differ greatly in wobbling (Fig. 5a, b) and non-wobbling (Fig. 5c, d) coding systems, for the first time (Fig. 5) also distinguishes coding systems. Non-wobble codes, like those treated thus far, admit any assignment to triplets, however their codon sequences may be related. In contrast, wobble (Table 1) allows G:U and A:U third-position pairs (Crick 1966), fixing some adjacent codons' meaning.

Coding Tables Initiated with 16 SGC Assignments: Wobble Coding

Figure 5a shows mean durations (passages) and number of random assignments (inits) needed to attain particular numbers of encoded functions. Notably, both time and assignments needed to reach a specified wobble code complexity increase dramatically after 20 encoded functions. In fact, starting at an initial mean of 12.35 functions (from 16 chance SGC triplets), encoding the last two functions costs more than 100-fold as much time and assignments as do the other 20 encodings. This implies a history of great complexity—a complete wobble coding table has assigned triplets an average of ≈ 25,000 times, thereby overwhelmingly making repeated, futile assignments.

This is 'completion complexity,' a reflection of the difficulty of fitting together wobbling coding boxes in a fixed space that must contain 22 of them. Many explorations, involving decay and reassignment (detected as initiations in Fig. 5a), are required to complete a wobble coding table. In addition, forces driving change weaken as full tables are approached. Initiation slows near completion because unassigned codons become rare (Fig. 3). Mutational capture also slows near completion because one participant, the unassigned triplet, also becomes rare (Fig. 3). Finally, decay of

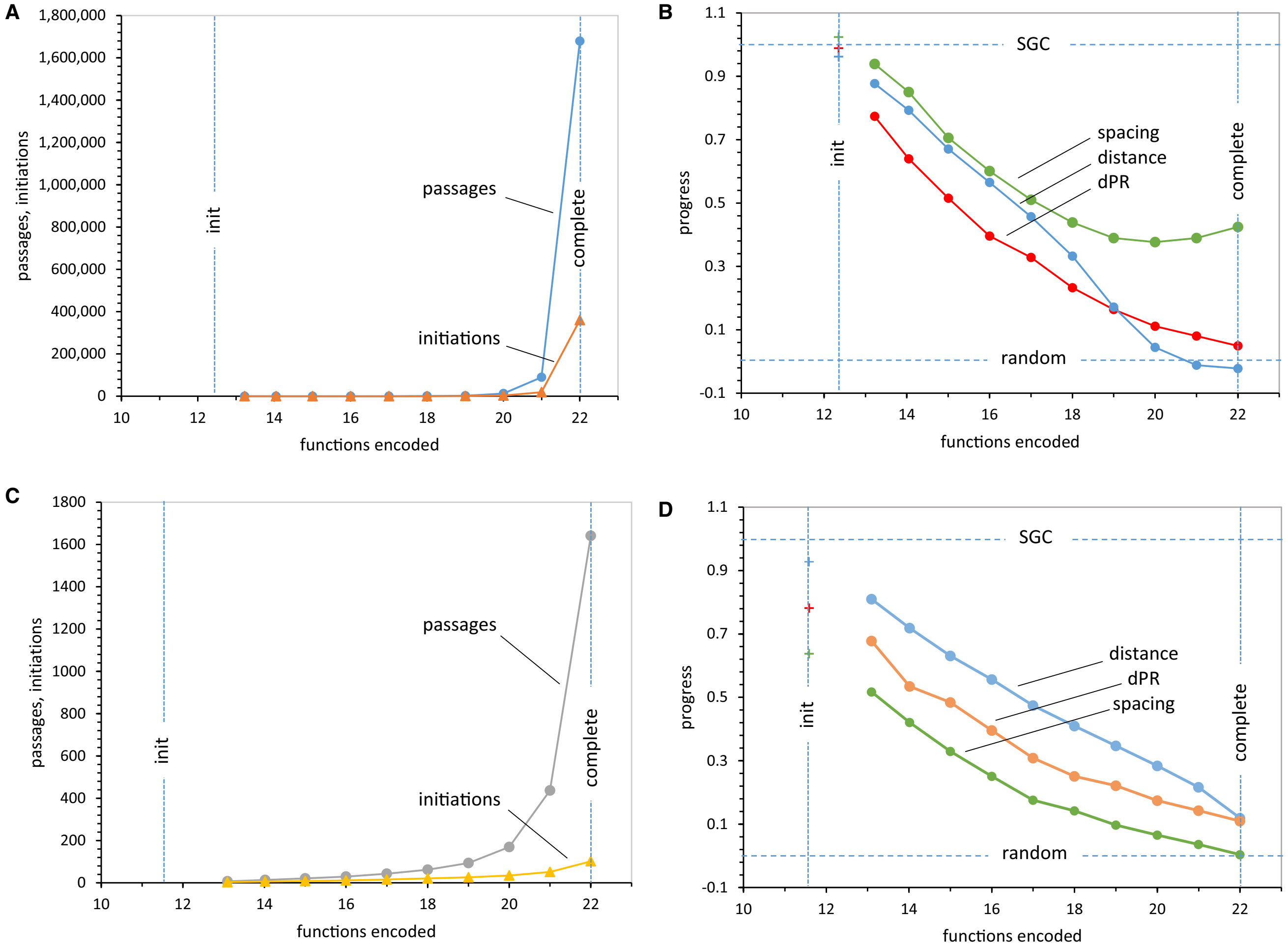


Fig. 5 **a** Mean passages and initiations required for random wobble encoding of varied numbers of functions, after 16 random SGC wobble initiations. 1000 evolutions using random wobble coding and Coevo_PR mutational capture were stopped at numbers of encoded functions on the abscissa. Lines labeled "init" are just after 16 initial random SGC triplets are chosen, but before further evolution. The "complete" line marks acquisition of 22 encoded functions. $P_{\mathrm{mut}}=0.04$, $P_{\mathrm{decay}}=0.04$, $P_{\mathrm{init}}=0.6$, $P_{\mathrm{wob}}=0.5$, $P_{\mathrm{rand}}=1.0$. **b** Mean spacing, distance, and dPR progress values for random wobble encoding after 16 random SGC wobble initiations. Data from calculations in **a**. "Init" and "complete" lines as in **a**. "SGC" line indicates SGC-like progress values, "random" line indicates progress for randomized assignments. **c** Mean passages and initiations required for random non-wobble encoding, after 16 initial random SGC non-wobble initiations. 1000 evolutions using random wobble coding and Coevo_PR mutational capture were stopped at numbers of encoded functions on the abscissa. Lines labeled "init" indicate positions just after initial 16 random SGC triplets are chosen. Lines labeled "complete" indicate acquisition of 22 encoded functions. Transition probabilities as in **a**. **d** Mean spacing, distance, and dPR progress values for random non-wobble encoding after 16 initial random SGC non-wobble initiations. Data from calculations in **c**. Labeling as in **b**

assignments will be maximal near completion, also opposing completion (Fig. 3).

Moreover, there are accompanying effects on wobble coding order. As random assignments are added, progress decays (Fig. 5b) and coding tables move away from SGC order. However, there exists a partial exception; spacing. Because wobble initiations make closely spaced identical assignments, wobble's spacing progress uniquely resists erosion, persisting indefinitely at ≈ 40% of SGC levels. However, wobble's spacing order occurs without parallel effects on distance or dPR, which descend to indistinguishability from random coding (line labeled 'random').

Note the contrasting situation *before* any evolution. Random groups of 16 initial SGC-like assignments (at 'init') have average progress values (points at upper left) that approximate the SGC itself (line labeled SGC). Initiation with wobble particularly improves spacing and dPR order (compare Fig. 5b, d). Sixteen such wobble initiations immediately make an incomplete coding table with spacing, distance, and dPR similar to the SGC.

Coding Tables Initiated with 16 SGC Assignments: Non-wobble Coding

Figure 5c shows mean duration and number of random non-wobbling assignments to reach particular numbers of encoded functions. Completion complexity exists for non-wobble codes, but is much less obstructive than with wobble. To pass from 20 to 22 non-wobbling functions (Fig. 4c), only 2.9-fold more initiations and 9.7-fold more durations are required. In addition, even if non-wobbling completion at 22 functions is mandated, only ≈ 1 additional assignment per triplet must occur.

However, code completion again undermines progress values and coding order. Figure 5d shows that spacing progress is generally reduced because the close spacing enforced by wobble assignments does not exist (compare spacing lines, Fig. 5d and b). Keeping in mind that non-wobble evolution is far shorter (Fig. 5c vs 5a), requiring 1/1000 the 20 → 22 function assignments for wobble—spacing, distance and chemical order still descend to near that of randomly formed coding tables (Fig. 5d).

Wobble Summary

Non-wobble completion is quicker and simpler than for a wobbling code, but if supplied by initial SGC assignments, code order still decays decisively. The evolutionary history modeled in Fig. 5 (initial stereochemistry, with and without wobble) is improbable, even if one's goal is coding that only faintly resembles the SGC. One must avoid the decay of SGC-like order supplied by initial wobble assignments (Fig. 5b, d), and also mitigate related effects of delay (Fig. 5a, c) during progress from a near-complete to a complete wobble code.

An Apparent Solution for Completion Complexity

Dramatic delays are confined to the era between 20 encoded functions and completion (Fig. 5a). Accordingly, it is possible that a minority of encoded functions evolved later than the majority, perhaps via a different route. This is an appealing notion for independent reasons.

Coding of translational initiation differs greatly in bacteria and eukaryotes (Kozak 1999). Bacteria initiate internally, using mRNA–rRNA complementarity as a guide, while eukaryotes scan from a 5′ mRNA end to a first favorable AUG (Hinnebusch and Lorsch 2012). These fundamental differences suggest that definitive translation initiation evolved late, after divergence of the major domains of life.

Translation termination also differs in bacteria and eukaryotes, much more than encoding of amino acids, which is similar throughout Earth biota. Protein release factors have different evolutionary origins in different domains of life (Vestergaard et al. 2001), and auxiliary factors, like those that recycle the joined ribosomal subunits after termination, are also of independent evolutionary origin (Zavialov et al. 2005). Moreover, definitive termination factors are sophisticated protein catalysts (e.g., Adio et al. 2018) that cannot exist until translation itself is sophisticated. Such considerations suggest that translation termination also took its final form late, after separation of life's domains (Burroughs and Aravind 2019). Thus, the suggestion of a majority of quickly encoded functions (≈ 20) and a small number added later by a different logic (≈ 2) has extensive, long-standing molecular support.

Rapidly Evolved Codes with Wobble

Figure 6a shows that average coding behavior (as in Fig. 5) conceals a possible resolution for wobble's completion complexity. Figure 6a plots the distribution of times to acquire 20 coded functions, for wobbling and non-wobbling codes, in successive 50-passage windows. Firstly, evolution to 20 functions (Fig. 6a) makes wobble less burdensome: mean times (signpost-shapes) to code completion are 28-fold greater for 20-function wobble codes than without wobble, instead of 1000-fold (Fig. 5) at 22 encoded functions. Modes, most probable completion times, do not actually differ greatly for wobble and non-wobble codes encoding 20 functions. Instead, wobble requires longer mean evolutionary times because of a long tail of tortuous histories, in which the many assignment decays and re-initiations (Fig. 5a) mentioned above in '…: **simple wobble coding**' gradually occur. So, if most probable routes (left hand peak in Fig. 6a) are taken instead of average ones, codes that exhibit SGC wobble, but also appear quickly can evolve. Figure 6b reinforces this discussion, showing that complete coding tables do not possess substantial early completions. A 22-function coding goal makes rapidly completed coding tables rare (Fig. 6b), instead of common (Fig. 6a).

Coding with Late Wobble

So, two short paths to wobble coding appear. In the first, 20 functions are encoded without wobble, exploiting the easy access non-wobble coding has to nearly complete tables (Fig. 5c). Afterward, late wobble innovation is quickly adopted—pre-existing 20-function codes quickly add wobble wherever possible. These events will be called "late wobble."

Coding with Continuous Wobble

A second rapid route to SGC-like wobble coding, called "continuous wobble," allows wobble assignments (Table 1) at initiation of coding and throughout. This path seeks access

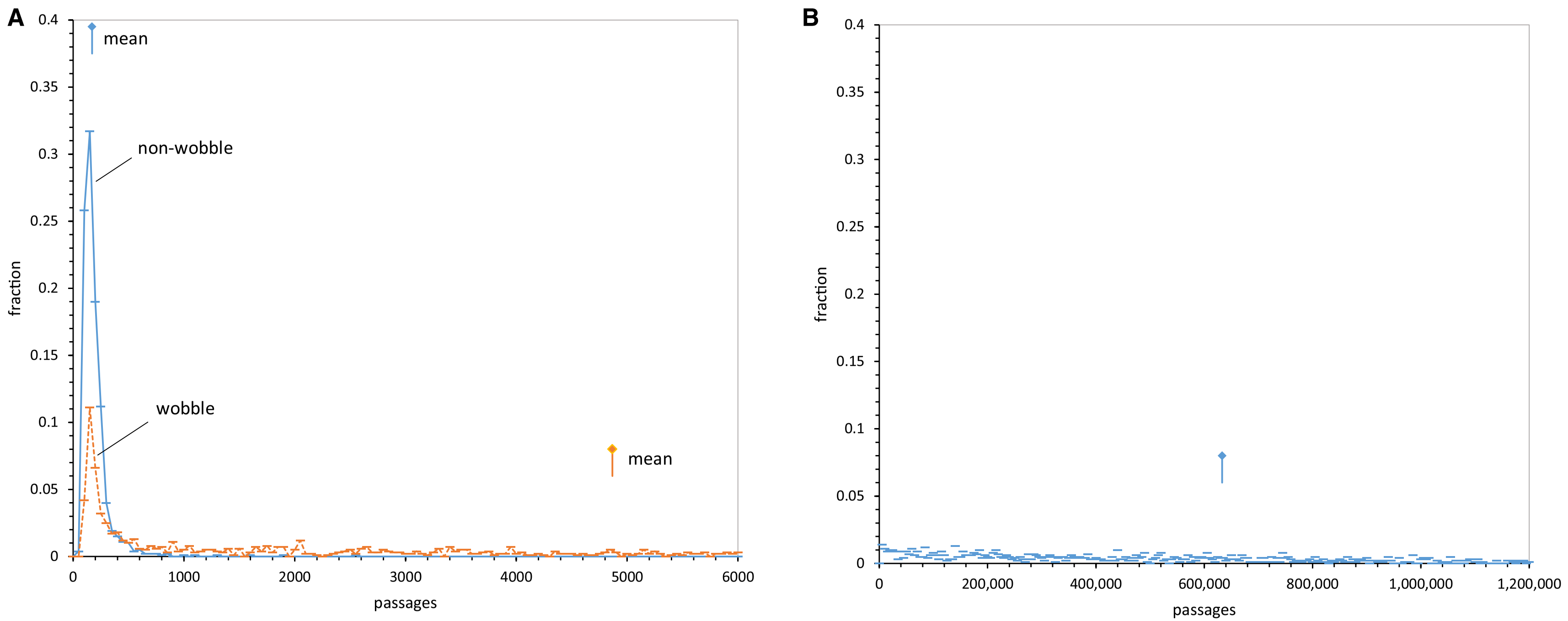


Fig. 6 **a** Fraction of near-complete 20-function codes after varied durations. 1000 evolutions with Coevo_PR captures and 10% random assignment were applied to wobble and non-wobble coding. Probabilities except for $P_{\text{rand}}=0.1$, as in Fig. 5a. Signpost symbols are distribution means. All non-wobble evolutions are plotted, but 0.216 of wobble evolutions required longer durations and are off-scale to the right. **b** Fraction of complete 22-function wobble codes at varied durations. 1000 evolutions with Coevo_PR assignments and $P_{\text{rand}}=0.1$ were used. Other probabilities as for **a**. Signpost symbol marks the mean. 0.142 of complete wobble evolutions required longer durations, off-scale to the right

to the SGC via the early subset of 20-function wobble codes (Fig. 6a). An SGC evolved via this minority is readily accessible—about ¼ of all code evolutions are eligible (Fig. 6a). However, SGC evolution via a subpopulation, even via a substantial minority like 25%, has consequences that return in "Discussion" section below.

A Second Barrier: Coding Table Order

Decline of progress values in Fig. 5b and d implies that the exquisitely ordered SGC (Fig. 1a) will require specific, persistent organizing influences. Therefore, we now compare often-cited sources of order. Calculations below compare six ordering mechanisms utilizing coevolution and paralogous selection, adaptation, and neutral mechanisms. These six mechanisms (termed Coevo, Coevo_PR, 0 ± 1 PR, 0 ± 2 PR, 0 ± 3 PR, 0 ± 4 PR) shape capture specificities for new triplets acquiring assignments related to existing ones.

Sources for Code Order

SGC non-randomness (Fig. 1a) is frequently attributed to stereochemical and/or historical causes (Knight et al. 1999).

Stereochemistry

Stereochemistry implies that the amino acids and cognate coding triplets are related by chemical interaction (Woese 1967; Crick 1968). Thus, stereochemical hypotheses predict that contemporary experiments can reveal code origins by studying interactions, for example, in RNA-binding sites selected for amino acids containing cognate coding triplets with unusual frequency (Yarus 2017b).

Coevolution

In another logic, historical explanations of coding order take one of three somewhat parallel forms. The first is **coevolution**, the idea that ancient encoded amino acids ceded their codons successively to related amino acids produced via extension of biosynthetic pathways (Wong 1975). Coevolution of the code and biosynthesis can be examined by testing the SGC to see if SGC triplet assignments are frequently connected in the way predicted by synthetic pathways (Amirnovin 1997; Ronneberg et al. 2000). Moreover, a possible molecular remnant of coevolution exists (Di Giulio 2002).

Adaptation

The second category of historical ideas is that there is a selective **adaptation** behind the code's order. For example, minimizing polar requirement change might guide capture assignments by minimizing the structural effects of substitution errors on protein structure (Freeland and Hurst 1998a). The SGC is very effective in minimizing the cost of such errors (Freeland and Hurst 1998b).

Neutral Change: Paralogy

A third **neutral** mechanism has been tested (Massey 2008, 2016, 2019). Because successor RNA–amino acid interactions would likely be molecular derivatives of prior RNA–amino acid interactions, they would employ related sequences. As a result, there could be sufficient order in a descendant coding table to explain the relatedness of triplets and SGC amino acids. Descent of related RNA sequences for related amino acids also occurs within adaptation. Selection therefore produces code order by means paralleling the neutral mechanism. Because adaptation and neutral paralogy plausibly exist together, producing overlapping, similar code order via a shared mechanism, I suggest their unification as paralogical sources of related triplet-amino acid assignments. Unification of selection and relatedness implements Crick's prescient comment that "similar amino acids would tend to have similar codons" (Crick 1968).

Encoding Order: Coevolution

The above considerations, determining functions assigned to neighborhood triplets captured by an existing assignment, have been programed. For coevolution, related triplets are assigned to amino acids linked by synthetic pathways, as suggested by Wong (1975), but using later thermodynamic corrections (Ronneberg et al. 2000). Such assignments for the purpose of testing coevolution usually are restricted to unique biosynthetic pathways, and common amino acid interconversions are ignored. However, in present evolutions, common amino acid interconversions are included and used to guide assignment of triplets to related amino acids. Coevolutionary amino acid conversions used are listed in Ronneberg et al. (2000). This assignment mechanism is called Coevo.

Coevolution Respecting PR Chemical Similarity

Here, biosynthetically related triplet/amino acid assignments are made as for coevolution, but synthetically related amino acid assignments that best conserve polar requirement are chosen with higher probability, rising linearly as PR difference decreases. This mechanism is called Coevo_PR.

Selection and Paralogy

To represent paralogical sources of order, related triplets are assigned amino acids with related polar requirements.

Amino acids are ordered by their PRs (Mathew and Luthey-Schulten 2008), and related triplets are randomly assigned to the next amino acid, up or down, in the PR list (0 ± 1 PR). Alternatively, random assignments are made ± 1 or 2 places in the PR list (0 ± 2 PR), or randomly, ± 1, 2, or 3 places in the PR list (0 ± 3 PR). When such random changes fall outside the range of real amino acid PRs, unoccupied triplet assignment defaults to the same amino acid as for the already assigned triplet (0 PR changes). These chemically conservative paralogical mechanisms are called 0 ± 1 PR, 0 ± 2 PR, 0 ± 3 PR, and 0 ± 4 PR.

Revised Code Evolution: Interspersed SGC Assignments

I now react to Fig. 5a–d by making several mechanistic alterations, and by targeting 20 function codes to minimize completion complication. Ordered assignments implementing the SGC (as for stereochemical assignments) are not made solely at initiation of code history, but arbitrarily interspersed with random assignments, throughout code evolution. In this way, ordered code exposure to random dilution (Fig. 5b, d) is shortened. The probability of random initiation is P_{rand}. $(1 - P_{\text{rand}})$ is the probability of SGC-like initiation; both are constant throughout code history.

First Route: Evolution to 20 Functions, then Late Wobble

Time and assignments for late wobble are similar for all histories, requiring ≈ 50 initiations in ≈ 170 passages. Notably, quick, advantageous evolution is retained: coding tables with 20 functions appear 20–30 times faster than average for continuous wobble. Moreover, different assignment mechanisms under late wobble require few, and similar, initiations (≈ 0.78 assignments/triplet). Thus, all late-wobbling assignment histories yield coding tables rapidly, without multiple decays and assignments. In fact, the 20 to 30-fold shorter times to late-wobbling coding tables are accompanied by similar-fold decreases in other events, like assignments

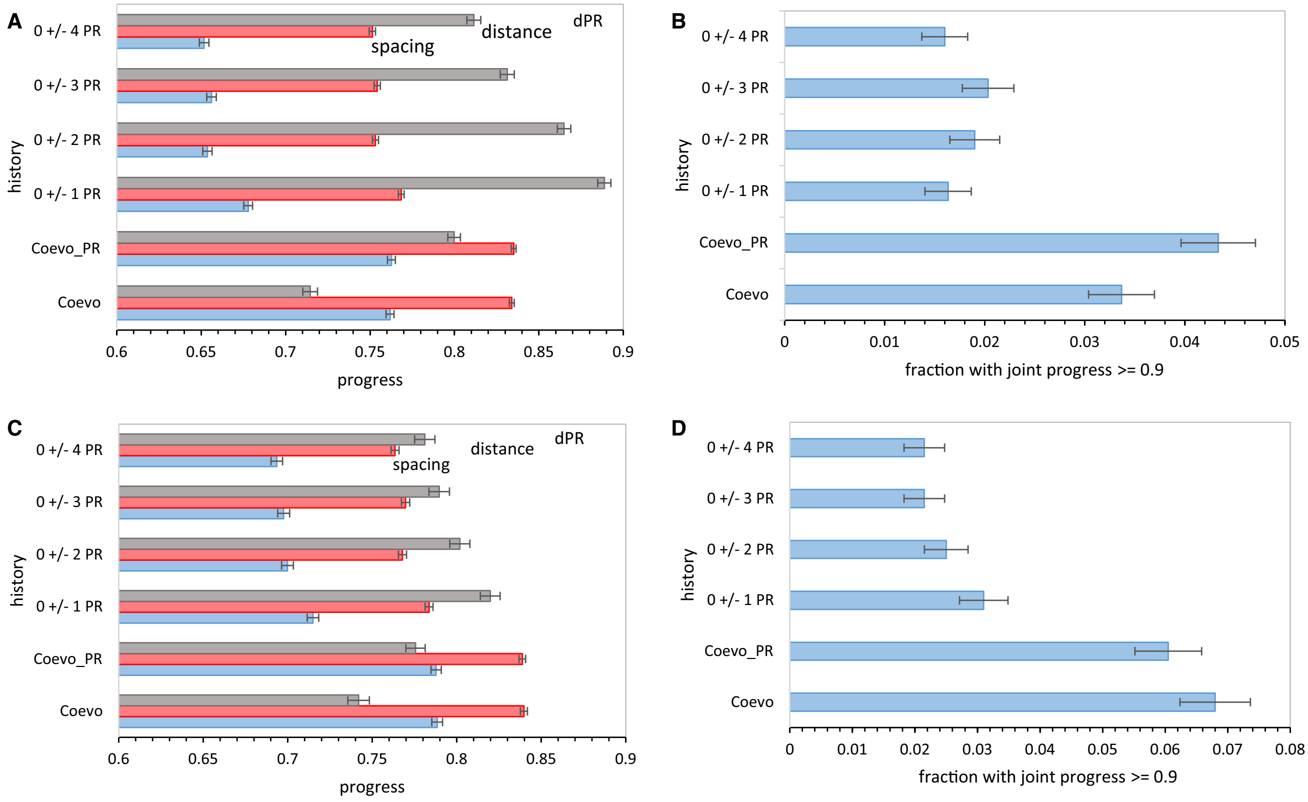


Fig. 7 **a** Progress for six mutational assignment histories using late wobble. 3000 evolutions were averaged for non-wobble encoding to 20 functions, then all possible late wobbles. Bars are sem's. $P_{\text{mut}} = 0.04$, $P_{\text{decay}} = 0.04$, $P_{\text{init}} = 0.6$, $P_{\text{rand}} = 0.1$. **b** Fraction of evolutions with joint progress for six mutational assignment histories using late wobble. Spacing, distance, and dPR progress values from 3000 late wobble evolutions were used, and joint progress is plotted with standard errors. Data are from **a**. **c** Progress for six mutational assignment histories using continuous wobble. Spacing, distance, and dPR progress values from 2000 continuous wobble evolutions were averaged, and plotted with standard errors. $P_{\text{mut}} = 0.04$, $P_{\text{decay}} = 0.04$, $P_{\text{init}} = 0.6$, $P_{\text{wob}} = 0.5$, $P_{\text{rand}} = 0.1$. **d** Fraction of evolutions with joint progress for six mutational assignment histories using continuous wobble. Fraction of 2000 continuous wobble evolutions with spacing, distance, and dPR progress simultaneously ≥ 0.9 were plotted with standard errors. Data are from **c**

transferred to new triplets. Equivalent time and inits among assignment mechanisms support evolutionary choice by criteria other than rate or complexity.

Assignment Mechanisms and Approach to the SGC

We now compare mean code order after varied assignment mechanisms during acquisition of 20 encoded functions. Figure 7a shows average progress for six assignment histories.

Overall outcomes from different assignment mechanisms faithfully reflect their individual rationales. Paralogical modes 0 ± 1 PR, 0 ± 2 PR, 0 ± 3 PR, 0 ± 4 PR are defined to conserve polar requirement, and their net evolutionary effects reflect this definition. They indeed conserve chemical order better than coevolutionary modes, Coevo and Coevo_PR. As chemical conservation relaxes, 0 ± 1 PR to 0 ± 4 PR, chemically ordered final codes become less frequent. Chemical order (dPR) is always more attainable than grouping (spacing progress), with resemblance to the SGC (distance) always the least frequent. But if chemical order were the sole coding goal, paralogous assignments produce it most effectively (Fig. 7a).

But, notably, what assignment mechanisms neglect also matters critically. Conserving chemical order (dPR) alone ignores and therefore sacrifices spacing and distance. Coevolution within Coevo and Coevo_PR always yields more compact spacing and closer approach to the SGC. The most balanced choice is Coevo_PR (Fig. 7b). Its dual emphasis on biosynthetically related assignments and related chemistry yields the most frequent mutual access to SGC-like spacing, distance, and dPR together, though chemical order is still most easily attained. To facilitate this balance, Coevo_PR is employed in further examples.

We confirm this choice and also take a step toward realism, plotting a quantity more relevant to code evolution than average progress—the fraction of evolved tables with joint progress ≥ 0.9 (Fig. 2c). In Fig. 7b, abundance of coding tables in the SGC vicinity is similar for all paralogical modes, the differences in Fig. 7a are compensated by changes in distributions. However, coevolutionary assignments produce ≈ twofold more SGC-proximal codes, with Coevo_PR again the best.

Second Route: Evolution to 20 Functions with Continuous Wobble

Assignment modes Coevo, Coevo_PR, 0 ± 1 PR… act similarly in continuous and late wobble. Again, paralogical modes promote chemical order (Fig. 7c), with tight conservation (0 ± 1 PR) more effective than less constrained assignments (0 ± 4 PR). But spacing and distance order are again enhanced in coevolutionary modes, with Coevo_PR again benefitting from its built-in preference for chemical order.

And again (Fig. 7d), using joint progress as a more comprehensive indicator, paralogical modes are similar and more than twofold less productive of codes with SGC-like order than coevolutionary assignments. In "Discussion" section below, we reconsider the ≈ 50% superiority of coevolution with continuous wobble (Fig. 7d) over coevolution with late wobble (Fig. 7b).

Near-Complete Codes from Late Wobble: Random Assignments

Disruption by random assignments first appeared in the specially constructed coding tables of Fig. 2. In Fig. 8a, mean disruptive effects of random assignment on 20-function late wobble code evolution are plotted. All progress values are decreased by random assignment (upper curves, Fig. 8a). But as expected (Fig. 2b, c), disruption is greater when SGC likeness is assayed using joint progress (Fig. 8a, lower curves, rightward ordinate). In particular, balanced progress gained from Coevo_PR assignments (Fig. 7a, b) is disrupted by a minority of random assignments. Good joint progress at $P_{\text{rand}} = 0$ (no random assignment) is lost if more than 15% of assignments are random rather than identical to the SGC. While some random assignment is allowable, > 15% is incompatible with an SGC-like result, particularly for spacing and distance order. Thus, 10% random assignment was chosen for illustrative calculations above.

Near-Complete Codes from Continuous Wobble: Random Assignments

Figure 8b plots the effect of random assignment on continuously wobbling codes. In particular, it reinforces the previous limit, random substitution with continuous wobble must be restricted; certainly $\leq 15\%$, better $\leq 10\%$.

But Fig. 8b differs from the late-wobbling case in Fig. 8a, as seen in its lower joint progress curves. Code order is similarly sensitive to random codon assignments in late and continuous wobble (joint progress ≥ 0.9; Fig. 8a, b). However, if coding capacity for ≥ 20 encoded functions is also required (joint progress ≥ 0.9 & ≥ 20 functions), then late wobble supplies more candidates than continuous wobble. This result traces to Fig. 6a because random assignment's effect on order is similar for late and continuous wobble, it is limitation to a minority (Fig. 6a) of continuous wobble coding tables that diminishes the frequency of codes in the vicinity of the SGC. By comparison, codes encoding 20 functions before adopting wobble are already near-complete, thus joint progress, and joint progress with completeness, superpose for late wobble (Fig. 8a).

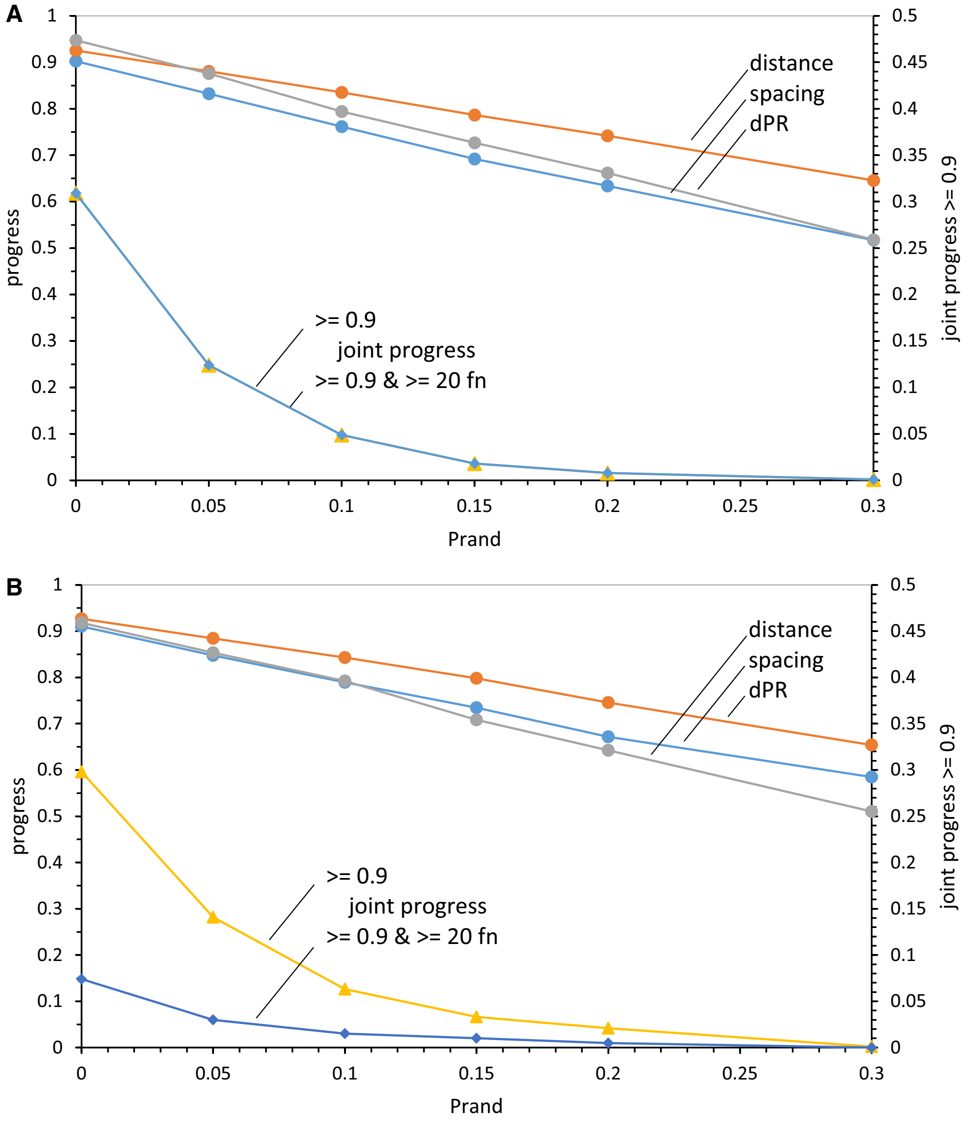


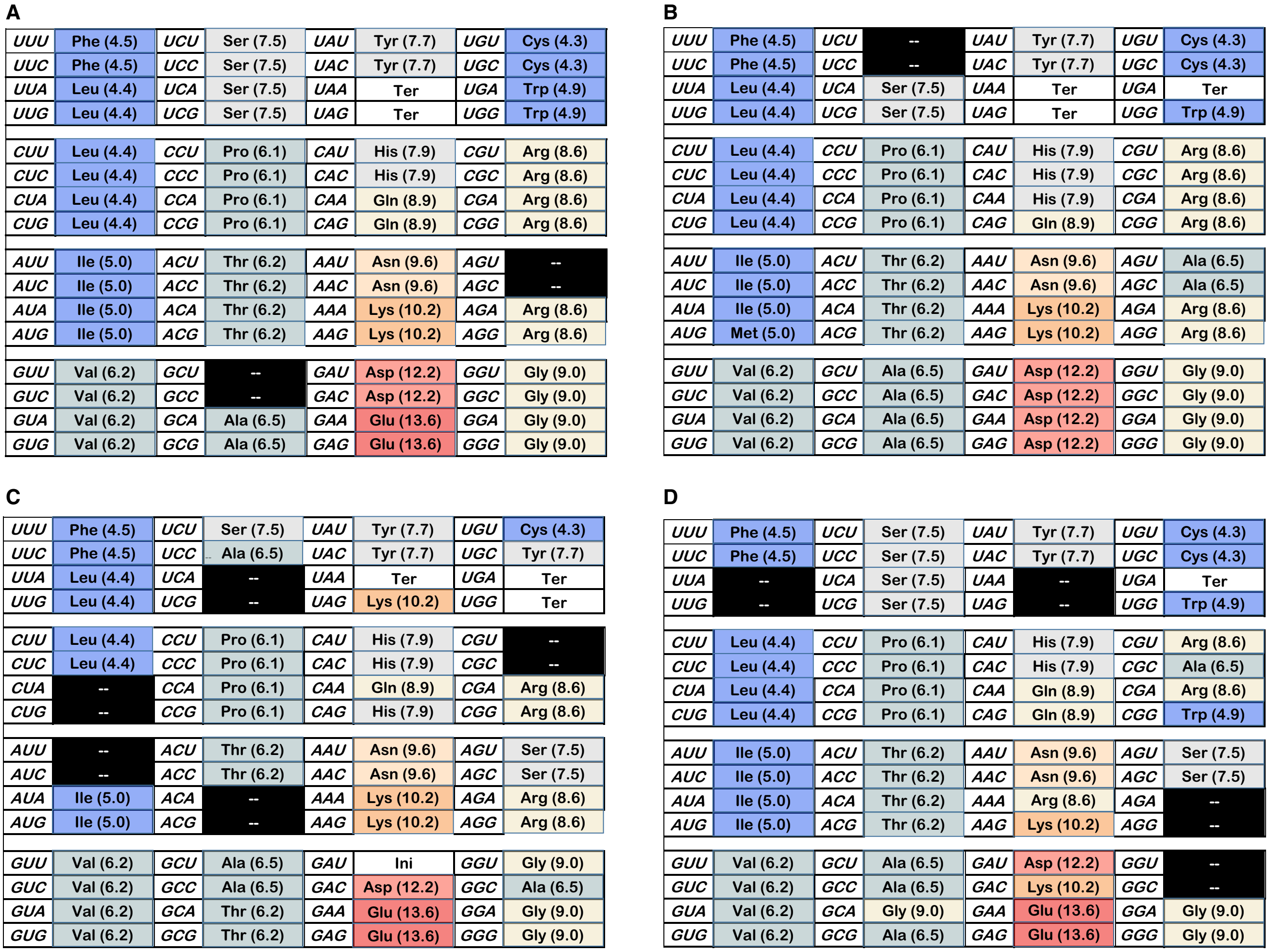
Fig. 8 **a** Effects of random assignments on late-wobbling code order. 1000 evolutions to 20 encoded functions with $P_{mut} = 0.04$, $P_{decay} = 0.04$, $P_{init} = 0.60$, and varied P_{rand} were used. Progress values plotted on left ordinate, fraction of evolutions with joint progress on right. **b** Effects of random assignments on continuous wobbling code order. 1000 evolutions to 175 passages with $P_{wob} = 0.5$ and otherwise, the same probabilities and plotting as in **a**

Evolved Examples Distributed Coding Outcomes

Because evolutionary outcomes span a large stochastic range (Fig. 2c), we must now grapple more fully with variation. To illustrate how progress statistics represent the SGC, coding tables under the rightward gray bar of Fig. 2c are displayed in Fig. 9. These examples were picked from 600 successively evolved random tables. The best available is shown, and also progress examples around 1.0, 0.95, and 0.9 to illustrate differing joint distributions.

Figure 9 shows coding tables in descending order of joint progress, using SGC-like initiations, random assignment 10%, late wobble, and Coevo_PR controlling-related triplet assignments. The most ordered table cannot be accurately placed because there are no comparable tables to define its real frequency; but frequencies for 1.0, 0.95, and 0.9 examples are computed from positions in the observed joint distribution (Fig. 9).

These tables exemplify the use of progress indices to characterize SGC-like order. For example, comparison to the SGC shows that Fig. 9a through and including 9D resemble the highly ordered SGC (Fig. 1a) much more than they do a random coding table (Fig. 1b), thereby substantiating progress index shifts plotted in Fig. 8a and b. A detailed examination of these examples also indicates that a code with high resemblance to the SGC would be accessible from a small population of hundreds of codes evolved by these means. Call this outcome ‘distribution fitness,’ to indicate that the better members of a distribution contribute disproportionately to evolutionary potential. For example, about 1 in 24 late wobbling, 10% random, Coevo_PR coding tables are equivalent or better than Fig. 9d, which



A

UUU	Phe (4.5)	UCU	Ser (7.5)	UAU	Tyr (7.7)	UGU	Cys (4.3)
UUC	Phe (4.5)	UCC	Ser (7.5)	UAC	Tyr (7.7)	UGC	Cys (4.3)
UUA	Leu (4.4)	UCA	Ser (7.5)	UAA	Ter	UGA	Trp (4.9)
UUG	Leu (4.4)	UCG	Ser (7.5)	UAG	Ter	UGG	Trp (4.9)
CUU	Leu (4.4)	CCU	Pro (6.1)	CAU	His (7.9)	CGU	Arg (8.6)
CUC	Leu (4.4)	CCC	Pro (6.1)	CAC	His (7.9)	CGC	Arg (8.6)
CUA	Leu (4.4)	CCA	Pro (6.1)	CAA	Gln (8.9)	CGA	Arg (8.6)
CUG	Leu (4.4)	CCG	Pro (6.1)	CAG	Gln (8.9)	CGG	Arg (8.6)
AUU	Ile (5.0)	ACU	Thr (6.2)	AAU	Asn (9.6)	AGU	--
AUC	Ile (5.0)	ACC	Thr (6.2)	AAC	Asn (9.6)	AGC	--
AUA	Ile (5.0)	ACA	Thr (6.2)	AAA	Lys (10.2)	AGA	Arg (8.6)
AUG	Ile (5.0)	ACG	Thr (6.2)	AAG	Lys (10.2)	AGG	Arg (8.6)
GUU	Val (6.2)	GCU	--	GAU	Asp (12.2)	GGU	Gly (9.0)
GUC	Val (6.2)	GCC	--	GAC	Asp (12.2)	GGC	Gly (9.0)
GUA	Val (6.2)	GCA	Ala (6.5)	GAA	Glu (13.6)	GGA	Gly (9.0)
GUG	Val (6.2)	GCG	Ala (6.5)	GAG	Glu (13.6)	GGG	Gly (9.0)

B

UUU	Phe (4.5)	UCU	--	UAU	Tyr (7.7)	UGU	Cys (4.3)
UUC	Phe (4.5)	UCC	--	UAC	Tyr (7.7)	UGC	Cys (4.3)
UUA	Leu (4.4)	UCA	Ser (7.5)	UAA	Ter	UGA	Ter
UUG	Leu (4.4)	UCG	Ser (7.5)	UAG	Ter	UGG	Trp (4.9)
CUU	Leu (4.4)	CCU	Pro (6.1)	CAU	His (7.9)	CGU	Arg (8.6)
CUC	Leu (4.4)	CCC	Pro (6.1)	CAC	His (7.9)	CGC	Arg (8.6)
CUA	Leu (4.4)	CCA	Pro (6.1)	CAA	His (7.9)	CGA	Arg (8.6)
CUG	Leu (4.4)	CCG	Pro (6.1)	CAG	Gln (8.9)	CGG	Arg (8.6)
AUU	Ile (5.0)	ACU	Thr (6.2)	AAU	Asn (9.6)	AGU	Ala (6.5)
AUC	Ile (5.0)	ACC	Thr (6.2)	AAC	Asn (9.6)	AGC	Ala (6.5)
AUA	Ile (5.0)	ACA	Thr (6.2)	AAA	Lys (10.2)	AGA	Arg (8.6)
AUG	Met (5.0)	ACG	Thr (6.2)	AAG	Lys (10.2)	AGG	Arg (8.6)
GUU	Val (6.2)	GCU	Ala (6.5)	GAU	Asp (12.2)	GGU	Gly (9.0)
GUC	Val (6.2)	GCC	Ala (6.5)	GAC	Asp (12.2)	GGC	Gly (9.0)
GUA	Val (6.2)	GCA	Ala (6.5)	GAA	Asp (12.2)	GGA	Gly (9.0)
GUG	Val (6.2)	GCG	Ala (6.5)	GAG	Asp (12.2)	GGG	Gly (9.0)

C

UUU	Phe (4.5)	UCU	Ser (7.5)	UAU	Tyr (7.7)	UGU	Cys (4.3)
UUC	Phe (4.5)	UCC	Ala (6.5)	UAC	Tyr (7.7)	UGC	Tyr (7.7)
UUA	Leu (4.4)	UCA	--	UAA	Ter	UGA	Ter
UUG	Leu (4.4)	UCG	--	UAG	Lys (10.2)	UGG	Ter
CUU	Leu (4.4)	CCU	Pro (6.1)	CAU	His (7.9)	CGU	--
CUC	Leu (4.4)	CCC	Pro (6.1)	CAC	His (7.9)	CGC	--
CUA	--	CCA	Pro (6.1)	CAA	Gln (8.9)	CGA	Arg (8.6)
CUG	--	CCG	Pro (6.1)	CAG	His (7.9)	CGG	Arg (8.6)
AUU	--	ACU	Thr (6.2)	AAU	Asn (9.6)	AGU	Ser (7.5)
AUC	--	ACC	Thr (6.2)	AAC	Asn (9.6)	AGC	Ser (7.5)
AUA	Ile (5.0)	ACA	--	AAA	Lys (10.2)	AGA	Arg (8.6)
AUG	Ile (5.0)	ACG	--	AAG	Lys (10.2)	AGG	Arg (8.6)
GUU	Val (6.2)	GCU	Ala (6.5)	GAU	Ini	GGU	Gly (9.0)
GUC	Val (6.2)	GCC	Ala (6.5)	GAC	Asp (12.2)	GGC	Ala (6.5)
GUA	Val (6.2)	GCA	Thr (6.2)	GAA	Glu (13.6)	GGA	Gly (9.0)
GUG	Val (6.2)	GCG	Thr (6.2)	GAG	Glu (13.6)	GGG	Gly (9.0)

D

UUU	Phe (4.5)	UCU	Ser (7.5)	UAU	Tyr (7.7)	UGU	Cys (4.3)
UUC	Phe (4.5)	UCC	Ser (7.5)	UAC	Tyr (7.7)	UGC	Cys (4.3)
UUA	--	UCA	Ser (7.5)	UAA	--	UGA	Ter
UUG	--	UCG	Ser (7.5)	UAG	--	UGG	Trp (4.9)
CUU	Leu (4.4)	CCU	Pro (6.1)	CAU	His (7.9)	CGU	Arg (8.6)
CUC	Leu (4.4)	CCC	Pro (6.1)	CAC	His (7.9)	CGC	Ala (6.5)
CUA	Leu (4.4)	CCA	Pro (6.1)	CAA	Gln (8.9)	CGA	Arg (8.6)
CUG	Leu (4.4)	CCG	Pro (6.1)	CAG	Gln (8.9)	CGG	Trp (4.9)
AUU	Ile (5.0)	ACU	Thr (6.2)	AAU	Asn (9.6)	AGU	Ser (7.5)
AUC	Ile (5.0)	ACC	Thr (6.2)	AAC	Asn (9.6)	AGC	Ser (7.5)
AUA	Ile (5.0)	ACA	Thr (6.2)	AAA	Arg (8.6)	AGA	--
AUG	Ile (5.0)	ACG	Thr (6.2)	AAG	Lys (10.2)	AGG	--
GUU	Val (6.2)	GCU	Ala (6.5)	GAU	Asp (12.2)	GGU	--
GUC	Val (6.2)	GCC	Ala (6.5)	GAC	Lys (10.2)	GGC	--
GUA	Val (6.2)	GCA	Gly (9.0)	GAA	Glu (13.6)	GGA	Gly (9.0)
GUG	Val (6.2)	GCG	Ala (6.5)	GAG	Glu (13.6)	GGG	Gly (9.0)

Fig. 9 Sample coding tables from 600 late wobble, 20-function evolutions. The best observed, and example codes having joint progress ≈ 1, ≈ 0.95, and ≈ 0.90 from late-wobbling evolution are shown. **a** Arguably the best late wobble coding table observed in 600. Spacing progress, 1.129; distance, 1.048; dPR, 0.880. Frequency approximately 1 of 600: number 294/600: relation to SGC—2 altered assignments (AUG Ile and UGA Trp), 4 unassigned, but SGC chemical ordering (Fig. 1a) fully reproduced. **b** Example, late wobble joint progress ca. 1.0. Spacing progress, 0.983; distance, 1.002; dPR, 0.998; frequency of equivalent or better indices = 0.0015: number 23/600: relation to SGC—3 altered assignments (AGU/C Ala, GAA/G Asp and CAA His), 2 unassigned, but preserves chemical order except for AGY Ser. **c** Example, late wobble joint progress ca. 0.95. Spacing progress, 0.946; distance, 0.937; dPR, 1.056; frequency of equivalent or better indices, 0.0045: number 6/600: relation to SGC—10 altered assignments, 10 unassigned, intermediate polar requirement bloc seriously disrupted. **d** Example, late wobble joint progress ca. 0.9. Spacing progress, 0.912; distance, 0.905; dPR, 0.920; frequency of equivalent or better indices, 0.0325: number 286/600: relation to SGC—6 altered assignments, 8 unassigned, small perturbations in intermediate and very polar blocs

shows progress values ≈ 90% the distance from random to SGC coding.

Other non-trivial implications appear from Fig. 9. The frequency of coding tables with spacing ≅ distance ≅ dPR ≥ 1 is low (Figs. 2c, 7d). Thus, orderly coding tables are not a subset having uniformly favorable properties; instead, progress values vary individualistically.

The Second Route to an SGC: Continuous Wobble to 20 Functions

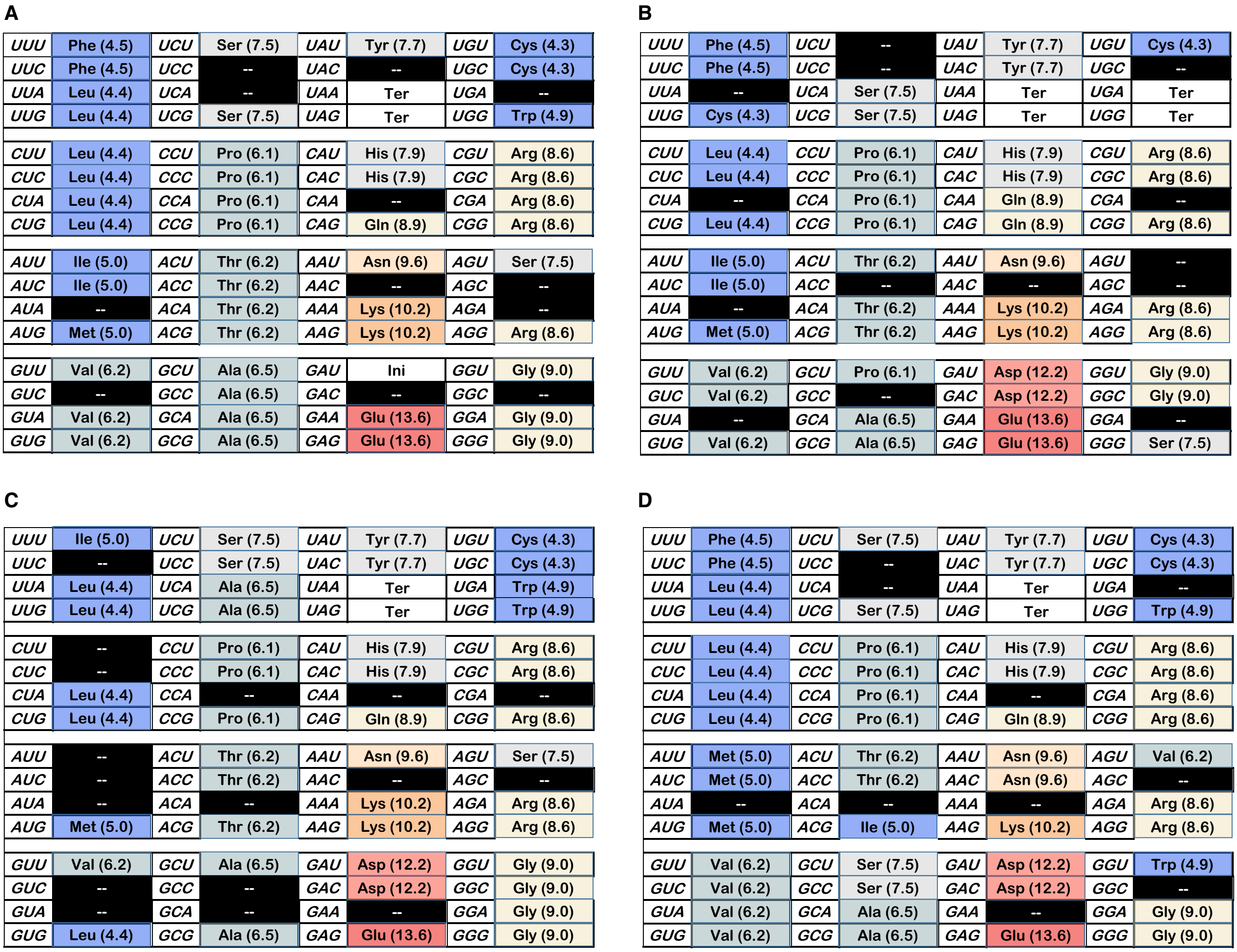
The second route to an ordered wobbling SGC is code completion during the early 20 function peak (Fig. 6a). In order to present explicit quantitation, we concentrate on a population of coding tables at 200 passages. Because all such coding tables have existed for exactly 200 passages, all experience similar mean development, with close to 51 initiations, 0.45 decays, and 12 mutational captures.

Overall Order is Roughly Similar

As Fig. 2c showed, distributed joint progress for continuous wobble early is very similar to joint progress for late wobble, but with a slight advantage to continuous wobble. This continuous wobble advantage is re-evaluated in "Discussion" section.

Assignment Effects for Continuous Wobble

Order due to various kinds of mutational capture (Fig. 7a, b) also varies similarly to that for late wobble (Fig. 7c, d). Paralogous mechanisms conserve chemical order best, with tighter paralogous constraints (e.g., 0 ± 1 PR, 0 ± 2 PR) more effective. Again, coevolutionary mechanisms, Coevo and Coevo_PR, are better balanced, with better distance and spacing, and good, but usually less effective, chemical ordering. Thus, we continue using Coevo_PR for specific continuous wobble calculations.



A

UUU	Phe (4.5)	UCU	Ser (7.5)	UAU	Tyr (7.7)	UGU	Cys (4.3)
UUC	Phe (4.5)	UCC	--	UAC	--	UGC	Cys (4.3)
UUA	Leu (4.4)	UCA	--	UAA	Ter	UGA	--
UUG	Leu (4.4)	UCG	Ser (7.5)	UAG	Ter	UGG	Trp (4.9)
CUU	Leu (4.4)	CCU	Pro (6.1)	CAU	His (7.9)	CGU	Arg (8.6)
CUC	Leu (4.4)	CCC	Pro (6.1)	CAC	His (7.9)	CGC	Arg (8.6)
CUA	Leu (4.4)	CCA	Pro (6.1)	CAA	--	CGA	Arg (8.6)
CUG	Leu (4.4)	CCG	Pro (6.1)	CAG	Gln (8.9)	CGG	Arg (8.6)
AUU	Ile (5.0)	ACU	Thr (6.2)	AAU	Asn (9.6)	AGU	Ser (7.5)
AUC	Ile (5.0)	ACC	Thr (6.2)	AAC	--	AGC	--
AUA	--	ACA	Thr (6.2)	AAA	Lys (10.2)	AGA	--
AUG	Met (5.0)	ACG	Thr (6.2)	AAG	Lys (10.2)	AGG	Arg (8.6)
GUU	Val (6.2)	GCU	Ala (6.5)	GAU	Ini	GGU	Gly (9.0)
GUC	--	GCC	Ala (6.5)	GAC	--	GGC	--
GUA	Val (6.2)	GCA	Ala (6.5)	GAA	Glu (13.6)	GGA	Gly (9.0)
GUG	Val (6.2)	GCG	Ala (6.5)	GAG	Glu (13.6)	GGG	Gly (9.0)

B

UUU	Phe (4.5)	UCU	--	UAU	Tyr (7.7)	UGU	Cys (4.3)
UUC	Phe (4.5)	UCC	--	UAC	Tyr (7.7)	UGC	--
UUA	--	UCA	Ser (7.5)	UAA	Ter	UGA	Ter
UUG	Cys (4.3)	UCG	Ser (7.5)	UAG	Ter	UGG	Ter
CUU	Leu (4.4)	CCU	Pro (6.1)	CAU	His (7.9)	CGU	Arg (8.6)
CUC	Leu (4.4)	CCC	Pro (6.1)	CAC	His (7.9)	CGC	Arg (8.6)
CUA	--	CCA	Pro (6.1)	CAA	Gln (8.9)	CGA	--
CUG	Leu (4.4)	CCG	Pro (6.1)	CAG	Gln (8.9)	CGG	Arg (8.6)
AUU	Ile (5.0)	ACU	Thr (6.2)	AAU	Asn (9.6)	AGU	--
AUC	Ile (5.0)	ACC	--	AAC	--	AGC	--
AUA	--	ACA	Thr (6.2)	AAA	Lys (10.2)	AGA	Arg (8.6)
AUG	Met (5.0)	ACG	Thr (6.2)	AAG	Lys (10.2)	AGG	Arg (8.6)
GUU	Val (6.2)	GCU	Pro (6.1)	GAU	Asp (12.2)	GGU	Gly (9.0)
GUC	Val (6.2)	GCC	--	GAC	Asp (12.2)	GGC	Gly (9.0)
GUA	--	GCA	Ala (6.5)	GAA	Glu (13.6)	GGA	--
GUG	Val (6.2)	GCG	Ala (6.5)	GAG	Glu (13.6)	GGG	Ser (7.5)

C

UUU	Ile (5.0)	UCU	Ser (7.5)	UAU	Tyr (7.7)	UGU	Cys (4.3)
UUC	--	UCC	Ser (7.5)	UAC	Tyr (7.7)	UGC	Cys (4.3)
UUA	Leu (4.4)	UCA	Ala (6.5)	UAA	Ter	UGA	Trp (4.9)
UUG	Leu (4.4)	UCG	Ala (6.5)	UAG	Ter	UGG	Trp (4.9)
CUU	--	CCU	Pro (6.1)	CAU	His (7.9)	CGU	Arg (8.6)
CUC	--	CCC	Pro (6.1)	CAC	His (7.9)	CGC	Arg (8.6)
CUA	Leu (4.4)	CCA	--	CAA	--	CGA	--
CUG	Leu (4.4)	CCG	Pro (6.1)	CAG	Gln (8.9)	CGG	Arg (8.6)
AUU	--	ACU	Thr (6.2)	AAU	Asn (9.6)	AGU	Ser (7.5)
AUC	--	ACC	Thr (6.2)	AAC	--	AGC	--
AUA	--	ACA	--	AAA	Lys (10.2)	AGA	Arg (8.6)
AUG	Met (5.0)	ACG	Thr (6.2)	AAG	Lys (10.2)	AGG	Arg (8.6)
GUU	Val (6.2)	GCU	Ala (6.5)	GAU	Asp (12.2)	GGU	Gly (9.0)
GUC	--	GCC	--	GAC	Asp (12.2)	GGC	Gly (9.0)
GUA	--	GCA	--	GAA	--	GGA	Gly (9.0)
GUG	Leu (4.4)	GCG	Ala (6.5)	GAG	Glu (13.6)	GGG	Gly (9.0)

D

UUU	Phe (4.5)	UCU	Ser (7.5)	UAU	Tyr (7.7)	UGU	Cys (4.3)
UUC	Phe (4.5)	UCC	--	UAC	Tyr (7.7)	UGC	Cys (4.3)
UUA	Leu (4.4)	UCA	--	UAA	Ter	UGA	--
UUG	Leu (4.4)	UCG	Ser (7.5)	UAG	Ter	UGG	Trp (4.9)
CUU	Leu (4.4)	CCU	Pro (6.1)	CAU	His (7.9)	CGU	Arg (8.6)
CUC	Leu (4.4)	CCC	Pro (6.1)	CAC	His (7.9)	CGC	Arg (8.6)
CUA	Leu (4.4)	CCA	Pro (6.1)	CAA	--	CGA	Arg (8.6)
CUG	Leu (4.4)	CCG	Pro (6.1)	CAG	Gln (8.9)	CGG	Arg (8.6)
AUU	Met (5.0)	ACU	Thr (6.2)	AAU	Asn (9.6)	AGU	Val (6.2)
AUC	Met (5.0)	ACC	Thr (6.2)	AAC	Asn (9.6)	AGC	--
AUA	--	ACA	--	AAA	--	AGA	Arg (8.6)
AUG	Met (5.0)	ACG	Ile (5.0)	AAG	Lys (10.2)	AGG	Arg (8.6)
GUU	Val (6.2)	GCU	Ser (7.5)	GAU	Asp (12.2)	GGU	Trp (4.9)
GUC	Val (6.2)	GCC	Ser (7.5)	GAC	Asp (12.2)	GGC	--
GUA	Val (6.2)	GCA	Ala (6.5)	GAA	--	GGA	Gly (9.0)
GUG	Val (6.2)	GCG	Ala (6.5)	GAG	Glu (13.6)	GGG	Gly (9.0)

Fig. 10 Sample coding tables from 600 continuous wobble evolutions. The best, and example tables having joint progress ≈ 1, ≈ 0.95, and ≈ 0.90 at 200 passages are shown. **a** Arguably the best continuous wobble coding table of 600. Spacing progress, 1.072; distance, 1.031; dPR, 1.114. Frequency approximately 1 of 600: number 549/600: relation to SGC—1 altered assignment (GAU Ini), 12 unassigned, but reproduces SGC chemical ordering fully (compare Fig. 1a). The only 21-function coding table among examples. **b** Example, continuous wobble joint progress ca. 1.0. Spacing progress, 0.983; distance, 0.969; dPR, 0.967; frequency of equivalent or better values = 0.005: number 387/600: relation to SGC—4 altered assignments, 14 unassigned, but preserves chemical order. 20 encoded functions. **c** Example, continuous wobble joint progress ca. 0.95. Spacing progress, 0.956; distance, 0.941; dPR, 0.965; frequency of equivalent or better values, 0.012: number 372/600: relation to SGC—5 altered assignments, 17 unassigned, moderate perturbation of chemical order. 20 encoded functions. **d** Example, continuous wobble joint progress ca. 0.9. Spacing progress, 0.928; distance, 0.920; dPR, 0.909; frequency of equivalent or better values, 0.029: number 478/600: relation to SGC—7 altered assignments, 10 unassigned, substantial perturbation of intermediate polar requirement bloc. 20 encoded functions

Sensitivity to Random Assignment

The sensitivity of continuous wobbling to random (rather than SGC) assignments is again pronounced (Fig. 8b). All three progress values decline, with joint progress approaching random codes at $P_{\text{rand}} > \approx 0.15$, thus resembling late wobble (Fig. 8a). Moreover, joint overall order for continuous wobble is particularly sensitive to random assignments (Fig. 8b), as it was for late wobble (Fig. 8a), because of a similar requirement for simultaneous upper-tail behavior in the three distributions. Thus, majority SGC-like initiations (Figs. 2, 8a) are not unique to late wobble, but similarly required for continuous wobble (Fig. 8b).

A Highly Significant Difference Between Continuous and Late Wobble

Late and continuous wobble coding do differ. This is apparent in examples from 600 continuously wobbling (Fig. 10) versus parallel late-wobbling (Fig. 9) coding tables. Figure 10 illustrates the best order observed in a continuous wobbling population of 600 (joint progress values ≥ 1), and also codes with joint progress ≈ 1, 0.95, and 0.9.

More frequent unassigned triplets (black with white dashes) among continuous wobbling tables (Fig. 10) are apparent, compared to late wobbling (Fig. 9). Example tables were chosen to illustrate joint progress, but excess unassigned codons are not due to human choice. Figure 11 shows that a ≈ 20 triplet assignment superiority for late wobble is not idiosyncratic and will not disappear; it is characteristic of the near steady state. Evolution of almost-complete 20-function wobble codes will leave about a third of amino acid triplets unassigned. In contrast, unassigned late wobble triplets are fewer, more comparable to the small number of yet-to-be-encoded functions.

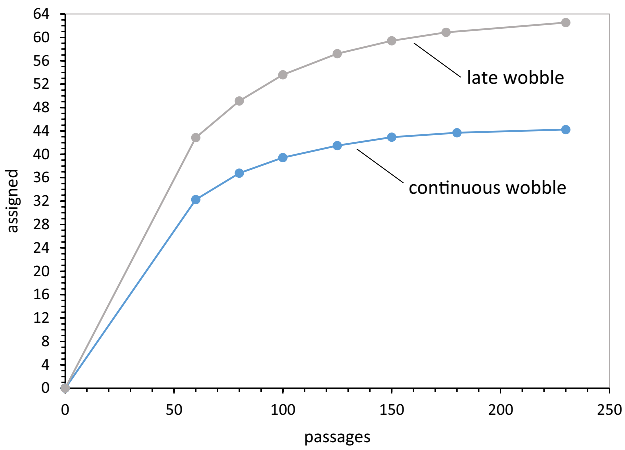


Fig. 11 Number of codons assigned early during late and continuous wobble code evolution. 1000 evolutions with Coevo_PR assignments, $P_{\text{mut}} = 0.04$, $P_{\text{decay}} = 0.04$, $P_{\text{init}} = 0.60$, $P_{\text{wob}} = 0.5$, $P_{\text{rand}} = 0.1$

Discussion

The Major Conclusion

A computation is introduced to evolve finished coding tables. Evolutionary qualities are varied to evaluate coding pathways. Computation was guided by simultaneous progress toward three objectives: SGC grouping of identical functions ("spacing"), minimal mutation to reach the SGC ("distance"), and SGC's minimal PR differences between codons related by single mutation ("dPR"). Thus, definitive origin information, the structure of the SGC, is combined with a coherent goal: a correct pathway for code emergence must yield the SGC. The major result is that an SGC-like coding table can be selected easily, from small independent groups of codes (Figs. 9a, 10a), with no requirement for exotic events.

The Effective Mechanism

The most rapid and accurate SGC evolution consigns translation initiation and termination to distinct, later evolutionary events; implements wobble after early non-wobbling code assignments; uses predominantly SGC-like, possibly stereochemical, assignment of sense codons, and exploits coevolutionary mutational capture with assignments that conserve polar requirement.

Wobble is Inevitable in Code Descent

Wobble's capture of third-position mutation is required to emulate the SGC, but it is a double-edged sword. By extending initial triplet assignments to related wobbles, it decisively increases order. Such order is visible in initial spacing, distance, and dPR progress arising from SGC-like triplets (Fig. 5b versus Fig. 5d, initial points, upper left). Nevertheless, subsequent evolution of a complete wobble code (22 encoded functions) is surprisingly prolonged (Fig. 5a); this is completion complexity. Continuous wobble's slow evolution also allows destructive effects on pre-existing spacing and dPR order (Fig. 5c). Spacing progress is the exception, sustained at a moderate level by wobble's persistent closely spaced identical assignments (Fig. 5b).

But Non-wobble Evolves to Completion Faster

Non-wobble code evolution contrasts strikingly with wobble, initial non-wobble allows quick code completion (Fig. 5c). However, because initial SGC triplet assignments are less effective, and wobble's intrinsic enhancement of spacing (Fig. 5b) does not exist, spacing, distance, and dPR still decline to near-random levels even during a non-wobbling code's greatly shortened random-assignment era (Fig. 5d).

Two Wobble Solutions

Non-wobble's advantageous evolutionary rate and wobble's ordering effects can combine if wobble was delayed, but immediately adopted into pre-existing codes when translational advances due to adaptor and ribosomal evolution made specific wobble possible. Moreover, because coevolution has a milder disruptive effect on spacing and distance order (Fig. 7a, c), and dPR can be enhanced by favoring conservation of polar requirement during biosynthetically related amino acid assignments (Fig. 7b), coevolution with intrinsic polar requirement matching (Coevo_PR) best balances the progress of a late-wobbling coding table (Fig. 7b). Twenty encoded functions are targeted to reduce completion complexity and because initiation and termination have distinct, unconserved mechanisms in life's domains. Such late wobble yields coding that attains order close to SGC levels (Fig. 9a).

The second route to prompt wobble coding exploits a minority of wobble codes completed very early (continuous wobble; Fig. 6a). These reproduce SGC order well (Fig. 2c), and also exhibit similar sensitivity to random assignments (Fig. 8a, b). But while continuous wobble easily completes coding, it does not fill coding tables (Figs. 10, 11).

Characteristics of Code Variation

To further resolve late wobble, the average 20-function late-wobbling coding table differs reproducibly from its exceptional subset in the SGC's vicinity; with joint progress ≥ 0.9. Such differences objectively, quantitatively characterize favorable routes toward the SGC.

Extended Conclusion: SGC-Like Coding Exploits Simplicity

Selection of superior joint progress (Table 2) dramatically increases resemblance to the SGC, as expected: from a mean of 0.7–0.8 to SGC-like levels of spacing, distance, and dPR.

Excellent coding appears in tables that have approximately half the average number of assignment decays. Superior coding also is achieved with about 60% the mutational captures occurring for an average code.

Initiations are also used more efficiently in coding tables which become more SGC-like. Only 85% of average initial assignments occur in the excellent code subset, and excellent codes reach 20 encoded functions in 65% of the average evolutionary duration.

That is, codes that resemble the SGC arise by chance simple routes, faster to complete. The favorable shortening of evolution in Table 2 is a smaller version of the ten thousandfold superiority of 20-function late wobble compared to complete 22-function continuous wobble coding (Figs. 5a, 7a). Put another way, it seems unlikely that the SGC arose initially assigning codons an average of 300 times over, as implied for complete continuous wobble (Fig. 5a), or even 40 times/triplet on average, as for average 20-function continuous wobble (Fig. 5a). By comparison, 0.66–0.8 assignments/triplet to reach a near complete, late-wobbling code seems wholly credible (Table 2).

Routine Unassigned Triplets, Late Assignments, Late Wobble

Late unassigned triplets (Figs. 4a, 9, 10) support late assignments, deep into coding table evolution. This in turn is consistent with late-arising assignments of unique character for translation initiation and termination (Fig. 5a, c). Late unassigned triplets may also be advantageous if they provide for late advent of complex amino acids like tryptophan and methionine before encoding (Koonin and Novozhilov 2017).

Late unassigned triplets are even more pertinent for late-wobble advent. In fact, late wobbles are the exceptional more frequent evolutionary event among SGC-like coding tables (Table 2). Because excellent SGC resemblance arises with fewer initiations of all kinds, more late wobble is used to

Table 2 Average 20-function late wobble codes versus an excellent subset

Population	Passages	Decays	Init	Mut capt	Wob init	Spacing	Distance	dPR
Mean late wobble	177	3.7	49.7	10.5	12.6	0.761	0.834	0.793
Joint progress ≥ 0.9	115	1.8	42.3	6.3	15.1	0.98	0.96	1.02

Means for 10,000 late-wobbling coding tables are shown down to the significant figure the same order as its sem. Evolution employed 10% random assignment, mutational capture of neighborhood triplets with coevo_PR, and late wobble at 20 encoded functions. There were 407/10,000 tables with joint progress ≥ 0.9 (Fig. 7). Captions are defined in the supplementary lexicon fill in superior coding tables. In Table 2, an average of 15.1 triplets are newly assigned after wobble is introduced to a superior 20 function code.

Code Sensitivity to Random Assignments

SGC-like order requires that randomly assigned codons be stringently limited in number (Fig. 2a). This imposes a limit, ≤ 15% random assignment if one requires accessible SGC regularity in either late-wobbling (Fig. 8a) or continuous wobbling codes (Fig. 8b). Such a limit is essential because good spacing, distance, and dPR occur together (Figs. 1a, 2c) in the SGC. It is therefore noteworthy that, all findings together, 1 of 24 late-wobbling coding tables, or 1 of 16 continuously wobbling coding tables approach SGC order (joint progress ≥ 0.9 in Fig. 7b, d). Limited randomness is required for a combinatorial reason, considered next.

Finding the SGC: The Combinatorial Abyss

Required simplicity hints at a much greater hindrance. Finding the ‘universal’ SGC demands exquisite discrimination. For slightly idealized coding tables like these, with 64 triplets and 20 encoded functions, there are $20^{64} = 1.8 \times 10^{83}$ ways to assign triplets to functions if unassigned functions are allowed. Thus, there are astronomical numbers of possible non-wobbling genetic codes.

The situation is “improved” somewhat by wobble ordering; there are 32 two-codon wobble triplet groups, as assigned here, and $20^{32} = 4.3 \times 10^{41}$ ways of assigning wobble groups to 20 functions, again with unassigned functions allowed. This is a minimum for wobbling genetic codes, because non-wobbling assignments are not counted, and will add to complexity.

The Standard Genetic Code (SGC) is an exceptionally ordered entity (Fig. 1a). Starting from unthinkably diverse sets like these, the SGC cannot plausibly be reached by starting at an arbitrary place, and/or taking an arbitrary path. Such an event has a small probability, because even arbitrary, pervasively ordered SGC-like tables (Fig. 1a) are a minute selection of total code configurations (Fig. 2c). Alternatively, it is 1.43×10^{17} s since the Earth aggregated from the early solar disc (Patterson 1956). It is rational to ask, even if a quick-starting, planetary-scale selection exists to reject whole orders of possible codes/second, can the 24–66 order-of-magnitude disparity between a random code search and time available for searching be spanned, and an SGC found?

It therefore seems very improbable that the genetic code arose by exhaustive comparison of alternatives. Instead, the combinatorial abyss must have been virtually circumvented. That is also the finding here, on independent grounds (Figs. 7, 9, 10). The present 10% solution, mandating that 90% of initiations, more or less, correspond to SGC assignments, confines a coding table to a negotiable vicinity in code space near the SGC (see “Sensitivity to Random Assignment” section, above). The result is wholly dramatic. Evolution need not distinguish 10^{83} or 10^{41} options; instead, close SGC relatives appear in populations as small as hundreds of independent codes (Figs. 7b, d, 9a, 10a).

But the abyss abides. Completion complications (Fig. 5a, c) are a portent, partly due to late evolutionary rates (Fig. 3), but also to the distance between almost complete and particular complete codes. Off-scale evolutions in in Fig. 6a and b are surely lost to the hiss of randomness. Code sensitivity to random substitution (Fig. 8a, b) is a hint of the combinatorial abyss.

Independent Evidence for Non-random Assignments

Coding tables must emphasize SGC-like assignments (“Finding the SGC” section, just above). A large amount of independent evidence supports such specific triplet association with cognate amino acids.

Experimental RNA-Binding Sites for Amino Acids

The most recent account of binding selection (Yarus 2017b) reviews data for 464 amino acid-binding sites, all of independent molecular origin, selected from random sequence RNAs in vitro for specific binding of 8 amino acids of varied chemical classes. These include sites for disparate amino acid side chains, for example, as for polar Arg (Janas et al. 2010) and hydrophobic Ile (Lozupone et al. 2003). When the smallest RNA-binding sites (perhaps more accessible in a primitive milieu) are specifically selected, the cognate triplet/amino acid association is observed in four of four cases (Yarus et al. 2009). Initially randomized nucleotide tracts in the same RNAs that are *not* required for amino acid binding are used as controls, and randomization and statistical tests show that triplet concentration in binding regions is specific and exceedingly non-random (Yarus 2017b). Statistical analysis requires assumptions, so it is notable that statistical tests are not essential to the crucial conclusion. Simplest sites and their triplets are so prevalent they are apparent when selected RNA sequences are simply aligned to reveal conserved sequences (for example, see L-Trp sites in: Majerfeld and Yarus 2005).

In total, comparisons of 7137 sequenced ribonucleotides within binding sites and 14,801 accompany control nucleotide sequences find that cognate triplets, whose nucleotides are essential to binding function, appear exceptionally often in amino acid-binding sites. Thus, selection reveals seven cognate anticodon triplets and two cognate codons *within* newly selected binding sites for six of eight tested amino acids. The two negative cases (L-Leu and L-Gln) are also among the least well explored; that is, those that yielded few sites for examination.

Further, a related tendency has been found in selected, specific RNA-binding sites for peptide, like His-Phe (Turk-Macleod et al. 2012) when affinity for both side chains is demanded. When these experimental stereochemical interactions are added to chemical models, consistency with the genetic code has been shown to be improved (Buhrman et al. 2013). These selection experiments, especially when combined, strongly support stereochemical interactions as one basis of primordial coding.

Binding to Natural RNAs

Natural examples of coding triplet/cognate amino acid also exist, such as the *Tetrahymena* active center (Yarus and Christian 1989), and the *Sulfobacillus* guanidinium riboswitch (Breaker et al. 2017; Yarus 2017b). Both bind arginine congeners to structures containing arginine codons.

Bioinformatic Analysis of RNA Structures

Moreover, there are data suggesting that relations between amino acids and their cognate coding triplets are yet more general. Coding triplets in the present RNA biostructures appear significantly related to their amino acids. Within crystallographically defined ribosomes, shortened distances appear between protein amino acids and cognate rRNA triplets (Johnson and Wang 2010). Most remarkably, when mRNA sequences are examined across complete genomes (Polyansky et al. 2013), their cognate peptide sequences show significant correlations with mRNA sequences, consistent with amino acid/RNA chemical interrelations. Such interrelations and their potential peptide/mRNA interactions persist even in the accessible surfaces of folded proteins (Beier et al. 2014).

Thus, five arguments using data of varied types, point to stereochemical SGC assignments. Such assignments are required to order the SGC (Fig. 8a, b), they are required to find the SGC in the combinatorial abyss (**Coding history**, above), chemical interaction between RNA-binding sites with essential triplets and their cognate amino acids has been selected, measured, and characterized, parallel interactions are observed in natural RNAs, and bioinformatic analyses find a wide-ranging amino acid–codon relations consistent with such interactions.

Unfamiliar Mechanisms in Coding History

Present models include events not usually discussed. Wobble captures a part of the underlying mutational pattern, and can strongly stimulate code order (Fig. 5b, d). In contrast, decays and reassignments are infrequently included in coding history. But here, they are routine. Because their inclusion allows evolution of the SGC (Figs. 9, 10), and they are chemically plausible, they can have occurred in SGC history. Moreover, each has a potential coding function, for example, reassignments allow recovery if non-specific initiations are inconsistent with SGC order (Fig. 8a, b). Of triplets assigned during average code history (Table 2), 69% are initial assignments (perhaps 90% of these stereochemical), 15% are mutational captures by an assigned triplet, and 16% are late appearing wobbles. Moreover, assignments decay, on average, once for 16 assigned triplets. The beautiful order of the SGC (Fig. 1a) does not rule out a heterogeneous origin.

Mixed Mechanisms in Coding History

Here, successful coding history emphasizes specific initial assignments, for example, stereochemical interactions. But it also calls on order from wobble, coevolution, and paralogy. Such a mixed basis presently seems inevitable, because different mechanisms emphasize order of distinct kinds. For example, wobble supports all progress, but particularly spacing and dPR (initial points, upper left: Fig. 5b, d). Each assignment capture mechanism makes a distinctive contribution to code order (Fig. 7), thus mixed contributions are needed to reach a broadly ordered SGC (Fig. 1a). In addition, a mixed history is independently plausible (e.g., Yarus 2017b) because coevolution and paralogy both require pre-existing assignments, implying pre-existing stereochemistry and/or minor randomness. The beautiful order of the SGC (Fig. 1a) does not rule out a heterogeneous origin.

More Definitive Coding History

Late wobble with 85–90% SGC-like assignment, coevolution with polar requirement selection accurately locate the SGC vicinity (Figs. 9a, 10a), but can this accuracy be improved? Yes, likely.

Only a restricted inventory of effects has been considered. For example, homogeneous, minimal models using constant rates for assignment, decay, and mutational capture are analyzed here. Plausible, more complex possibilities have not been examined. For examples, the possibilities that amino acids were encoded in subsets (Grosjean and Westhof 2016). Such segmentation would be very consistent with **A plausible primordial acceptor RNA** above, and should be tested. Perhaps transitions and transversions should be distinguished. Perhaps initial assignment took a different path (Wong 1975). Perhaps encoding was partially by RNAs, and subsequently by nucleoproteins (Koonin and Novozhilov 2017), or perhaps the SGC is a community's consensus (Vetsigian et al. 2006).

More generally, "…eventually one would reach a point where no new amino acid could be introduced without disrupting too many proteins. At this stage the code would be frozen" (Crick 1968). Given its universality, the SGC's origin lies in deep time, defining Crick's point. An accurate pathway that reproducibly attained the SGC (Fig. 1a), by linking to the Crick freezing point, would provide a credibly complete code history for discussion.

Bayesian Convergence

An objective criterion exists for such refinement. From Bayes' Theorem, the more likely mechanism explains more aspects of the current SGC (Yarus et al. 2005). Importantly, a hypothesis need not necessarily slowly become more plausible, given more evidence. Instead, it can multiply its probability if it explains independent aspects of the code. Such a "Bayesian Convergence" can rapidly reinforce a correct explanation. Convergence remains useful for events unimaginably remote in time and scale, like the origin of the genetic code (Yarus et al. 2005).

Convergence points to late wobble. This conclusion rests on two findings; in convergence, the argument's force is determined by their combined weight.

Late wobble quickly yields excellent coding tables, almost complete and almost full (Fig. 9). By comparison, continuous wobble also quickly creates excellent, almost complete coding tables (Fig. 10), but not almost full ones ("A Highly Significant Difference Between Continuous and Late Wobble" above; Fig. 11). Late wobble around 20 pre-encoded functions is almost sufficient to the SGC—it has unassigned triplets that approximately meet a requirement for later initiation and termination. In contrast, continuous wobble requires a yet-unknown way to assign ≈ 20 triplets (Fig. 11).

Late Wobble Allows Better SGC Access

Continuous wobble creates a subtle, but reproducible, advantage in code order (cf. Fig. 1a). Continuous wobble's better joint progress appears in Fig. 2c's distributions, in better prevalence of joint progress in Fig. 7d versus 7b, and around 10% randomness in Fig. 8b versus 8a. However, this advantage is lost because continuous wobble reaches the SGC through a minority of an evolving code population (Fig. 6a). Coding must be both ordered and complete (Fig. 1a), and over the usable range for random substitution (Fig. 8a, b), such continuously wobbling codes are threefold to fourfold rarer. Accordingly, SGC access appears threefold to fourfold more probable via late wobble.

Distribution Fitness Exploits Primordial Fluctuation

SGC evolution exploits 'distribution fitness,' that is, a rigorous requirement met by a heterogeneous group with excellent upper-tail members (Figs. 2c, 7). Undirected primordial variability is not a barrier, but instead is the crux of SGC emergence. This idea bears elaboration because it parallels previous findings.

There is an efficacious route to inherited gene expression, which requires only already-known RNA reactions (Yarus 2017a). Evolution of chemical inheritance is facilitated by a highly disperse population, from which selection readily picks extremely functional members. The pivotal diverse event is 'starting bloc selection,' meaning selection of individuals initiating a reaction. Early reactions have uniquely disperse product amounts—they are exceptionally suited to simultaneous selection of their product and its inheritance in a prebiotic, gene-free chemical system. In fact, a new inherited chemical capability can emerge after only one selection, possibly only a few days after partially activated primordial ribonucleotides accidently encounter each other (Yarus 2017a, 2018).

The starting bloc reappears during descent of the genetic code. In Table 2, SGC-like codes are early-appearing, averaging 65% the duration of average code evolutions. Thus, if codes resembling the SGC were selected at an early time, SGC-like early starters would be prominent. Special capability is synonymous with early function, the typical linkage in starting bloc selection.

In the third example, prebiotic chemical systems must change to become biotic ones, so one may ask how did prebiota advance without genes? One answer is 'chance utility,' in which reactant variation permits persistent, unexpected evolutionary outcomes. For example, it is not only possible, but in a fluctuating milieu can be routine, that a desirable reactant is selected despite a 100-fold excess of a destructive competitor (Yarus 2016).

Prebiotic History Required Selection of Favorable Fluctuations

Thus, chance utility, starting bloc selection and distribution fitness solve notable evolutionary problems because primitive systems offer fluctuation and distribution. In this way, unregulated primordial chemistry is intrinsically suited to evolutionary change, toward non-Darwinian chemical progress, toward primordial inheritance and later, toward Darwinian appearance of the genetic code. A connection between distributions and productive change suggests that prebiotic evolution itself may be a tractable branch of statistical mechanics. Prebiotic history presents puzzles of the size and complexity of planets, but even such puzzles can yield quantitative, probable solutions.

Methods

Computation

All calculations were performed on a Dell XPC laptop with an Intel Core i9 64-bit processor @ 2.9 GHz and 32 GB of RAM, running Microsoft Windows 10, v. 1709. Usually computer data were imported into Microsoft Excel 2016 32-bit as tab-delimited files for further analysis and conversion to graphics.

Modeling

The probabilistic coding table model was developed and run in console mode of the Lazarus Integrated Development Environment v.1.8.4, with the Free Pascal Compiler v.3.0.4 supplying run-time modules. Pascal source code, Ctable18d1.pas, configured for late wobble but capable of all calculations presented with slight adjustments, and RandSens, a Microsoft Excel file that illustrates post-simulation calculation of sensitivity to random assignments, are available on request.

Because of the speed of integer operations, coding tables were represented as arrays of integers. One code array is followed to a specified evolutionary end, e.g., full coding or complete coding, using whatever evolutionary rules are being investigated. Such arrays were translated into ordinary coding tables (as in Figs. 1, 12) using an alphabetically related dictionary, after evolutionary calculations. Analysis of populations of finished arrays yields the probability of SGC-like results.

Runs with varied numbers of passages suggest that the ≈ 900-line program run as above requires about 4 µs for one passage through a coding table and about 20 ms for one evolution (dependent on passage complexity).

Flow During One Passage Through a Nascent Coding Table

Figure 12 sketches the flow of operations during one passage through an evolving coding table. Both continuous and

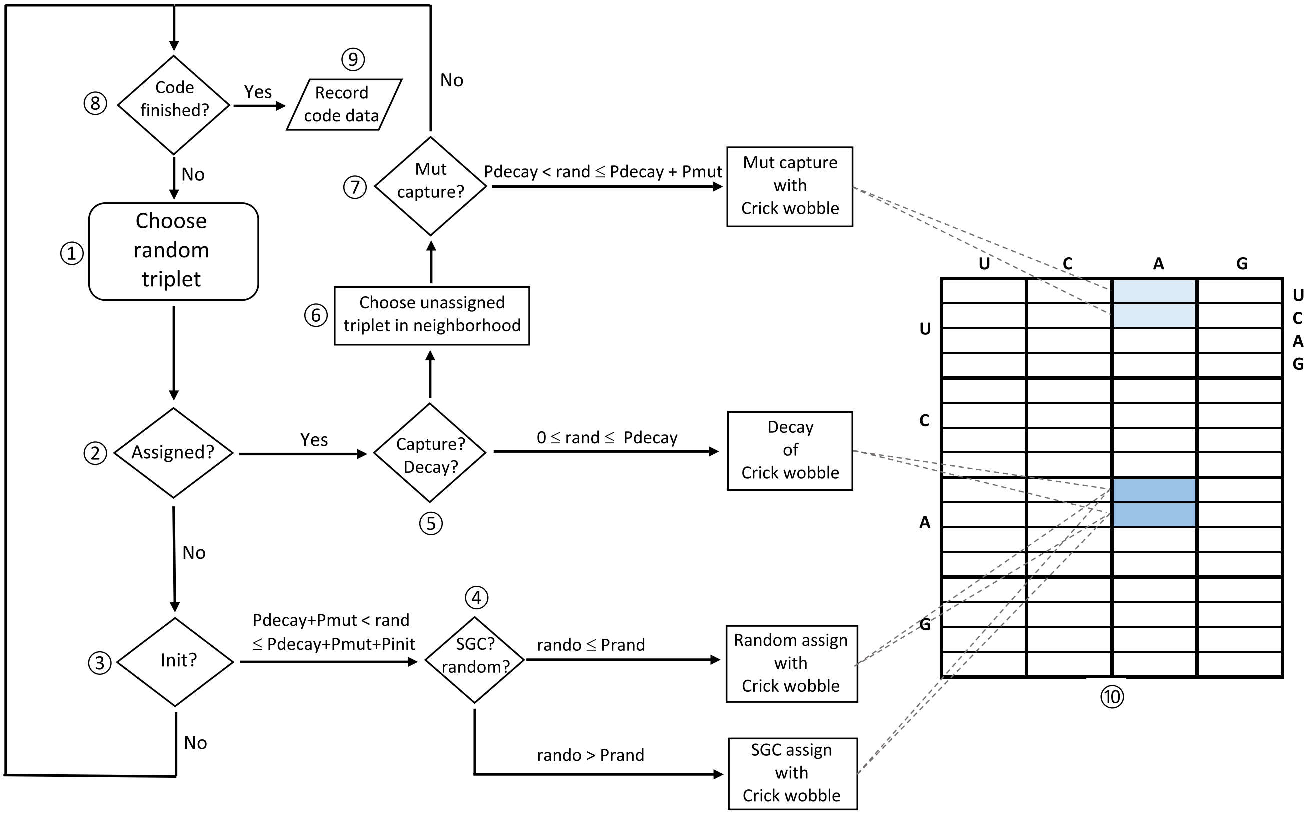


Fig. 12 Operations during one computed passage. The logic of coding evolution during one computer passage through a nascent code is shown on the left, and the effect of these operations is visualized on the right as a typical coding table (as in Figs. 1, 9, 10) evolving with Crick wobble assignments as an example (see "Flow During One Passage Through a Nascent Coding Table", "Methods" sections)

late wobble are described, but alterations for the latter are in notes (as for 4. Below). Notations rand and rando are uniformly distributed Mersenne Twister random numbers, $0 \leq \text{number} \leq 1$, chosen anew for each passage. *P* terms are probabilities, as defined in the next section of Methods.

1. A triplet is chosen with random integers 1–4 for all 3 array indices.
2. Is the chosen triplet already assigned to a function, or currently free?
3. If unassigned, random number rand determines whether an initial assignment will occur, as shown.
4. Random number rando determines whether assignments are to a random choice of the 22 possible assignments, or is drawn from an assignment pre-existing in the SGC. In Fig. 12, as drawn, assignments are not unique, but Crick wobbles are encoded whenever possible. Late wobble: For late Crick wobble evolution, assignments are unique, and Crick wobble is added at step 8, wherever possible.
5. The random floating point number rand determines whether an assigned codon will decay or will capture a neighborhood triplet (related by single mutation) for a function related to its current assignment, as exemplified in Fig. 12.
6. The 9 neighborhood assignments for a current triplet are searched to see if any are free for mutational capture. If > 1 is free, choose randomly; if none are free, go to 8.
7. If the step 6 search is successful, random number rand determines if capture of a randomly chosen free triplet will occur. Related assignment mechanisms utilize a variety of means, as in Fig. 7a, c.
8. This operation represents a *while* loop that determines whether a desired coding property has been attained, e.g., is the coding table full?
9. Selected properties, depending on the goal of the calculation, usually for 10^2 to 10^6 successive coding tables, are calculated and written to disk.
10. A 3-dimensional code array is displayed as a conventional coding table (Figs. 1, 9, 10), with operations on both codons in a Crick wobble group, as in this example, indicated. Coding tables with specified properties can also be detected, viewed onscreen, and saved during calculations. If adjacent unassigned codons are not available for Crick wobble, unique assignments are allowed. Late wobble: if late instead of continuous wobble is utilized, unique assignments are made to completion (at Step 8, Fig. 12), then wobble is added wherever possible.

Rate Constants and Probabilities

The kinetic method used here can be justified by showing that probabilities of reaction per passage are equivalent to use of normal chemical rate constants.

Initiations

The relation between P_{init} and the related first-order rate constant, k_{init}, with time in passages^{-1}, can be calculated by equating kinetic and probability equations (probability for selection times probability for subsequent reaction) for the overall rate of initiations/passage:

$$\frac{d_{\text{init}}}{dt} = k_{\text{init}} * u = P_{\text{selection}} * P_{\text{reaction}} = \frac{u}{64} * P_{\text{init}},$$

where u is the number of unassigned triplets, and time is in passages. So

$$k_{\text{init}} = P_{\text{init}}/64.$$

Decays

A similar approach to a first-order rate constant for assignment decay in passages^{-1} yields

$$k_{\text{decay}} = P_{\text{decay}}/64.$$

Mutational Captures

A second-order rate constant, k_{mut}, for mutational capture with units triplets^{-1} passages^{-1} must account for the probability that triplets neighboring an assigned triplet are so far unassigned, and can therefore be captured:

$$k_{\text{mut}}(64 - u)(u) = P_{\text{selection}} * P_{\text{reaction}} = (64 - u)/64 * 9(u/63)P_{\text{mut}}$$

$$k_{\text{mut}} = P_{\text{mut}}/448,$$

where $9\,(u/63) = u/7$ is the expected number of unassigned triplets within the mutational neighborhood of a selected, assigned triplet.

Controls

Controls suggested that randomly generated small mean probabilities derived from the Mersenne Twister algorithm in Free Pascal were accurate, that substitution of original experimental polar requirements for corrected ones would not materially change conclusions and that inclusion or exclusion of initiation/termination triplets from calculations where they are relevant would not substantially alter cited results. Transition probability variations alter coding, and have effects on order because they create the substrate for subsequent adoption into codes. But such alterations usually had smaller effects on order, and so have not been discussed in this first ms.

Random Spacing and Distance Values

The value for mean random mutational spacing and distance from an arbitrary triplet to other triplets in a random coding table can be calculated exactly from the fact that there are 9 triplets 1 mutation away from any initial triplet (its mutational neighborhood), 27 triplets 2 mutations away, and 27 triplets 3 mutations away:

$$\text{random mean spacing, distance} = \frac{(9 * 1) + (27 * 2) + (27 * 3)}{63} = 2.28571.$$

This is in excellent agreement with simulated values for 1000 randomized tables:

$$\text{spacing} = 2.286 \pm 0.002 \text{ (sem)}$$

$$\text{distance} = 2.284 \pm 0.002 \text{ (sem)}$$

thereby validating programed randomization and calculation of mean mutational distances.

Acknowledgements Many thanks are due Robin Dowell, John Heumann, Leslie Leinwand, Bill McClain and Jacob Stanley for helpful comments on early drafts of this work.

References

Adio S, Sharma H, Senyushkina T, Karki P, Maracci C, Wohlgemuth I, Holtkamp W, Peske F, Rodnina MV (2018) Dynamics of ribosomes and release factors during translation termination in *E. coli*. eLife 7:e34252. https://doi.org/10.7554/eLife.34252

Amirnovin R (1997) An analysis of the metabolic theory of the origin of the genetic code. J Mol Evol 44:473–476

Beier A, Zagrovic B, Polyansky AA (2014) On the contribution of protein spatial organization to the physicochemical interconnection between proteins and their cognate mRNAs. Life Basel Switz 4:788–799

Breaker RR, Atilho RM, Malkowski SN, Nelson JW, Sherlock ME (2017) The biology of free guanidine as revealed by riboswitches. Biochemistry 56:345–347

Buhrman H, van der Gulik PTS, Klau GW, Schaffner C, Speijer D, Stougie L (2013) A realistic model under which the genetic code is optimal. J Mol Evol 77:170–184

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 20:1–24

Chumachenko NV, Novikov Y, Yarus M (2009) Rapid and simple ribozymic aminoacylation using three conserved nucleotides. J Am Chem Soc 131:5257–5263

Collins DW, Jukes TH (1994) Rates of transition and transversion in coding sequences since the human-rodent divergence. Genomics 20:386–396

Crick FH (1966) Codon–anticodon pairing: the wobble hypothesis. J Mol Biol 19:548–555

Crick FHC (1968) The origin of the genetic code. J Mol Biol 38:367–379

Di Giulio M (2002) Genetic code origin: are the pathways of type Glu-tRNA(Gln) –> Gln-tRNA(Gln) molecular fossils or not? J Mol Evol 55:616–622

Freeland SJ, Hurst LD (1998a) Load minimization of the code: history does not explain the pattern. Proc R Soc Lond B Biol Sci 265:1–9

Freeland SJ, Hurst LD (1998b) The genetic code is one in a million. J Mol Evol 47:238–248

Grosjean H, Westhof E (2016) An integrated, structure- and energy-based view of the genetic code. Nucleic Acids Res 44:8020–8040

Hinnebusch AG, Lorsch JR (2012) The mechanism of eukaryotic translation initiation: new insights and challenges. Cold Spring Harb Perspect Biol 4:1–25

Illangasekare M, Yarus M (1999) Specific, rapid synthesis of Phe-RNA by RNA. Proc Natl Acad Sci USA 96:5470–5475

Illangasekare M, Yarus M (2012) Small aminoacyl transfer centers at GU within a larger RNA. RNA Biol 9:59–66

Illangasekare M, Sanchez G, Nickles T, Yarus M (1995) Aminoacyl-RNA synthesis catalyzed by an RNA. Science 267:643–647

Janas T, Widmann JJ, Knight R, Yarus M (2010) Simple, recurring RNA binding sites for L-arginine. RNA 16:805–816

Johnson DB, Wang L (2010) Imprints of the genetic code in the ribosome. Proc Natl Acad Sci USA 107:8298–8303

Knight RD, Freeland SJ, Landweber LF (1999) Selection, history and chemistry: the three faces of the genetic code. Trends Biochem Sci 24:241–247

Koonin EV, Novozhilov AS (2017) Origin and evolution of the universal genetic code. Annu Rev Genet 51:45–62

Kozak M (1999) Initiation of translation in prokaryotes and eukaryotes. Gene 234:187–208

Kumar S (1996) Patterns of nucleotide substitution in mitochondrial protein coding genes of vertebrates. Genetics 143:537–548

Ledoux S, Olejniczak M, Uhlenbeck OC (2009) A sequence element that tunes Escherichia coli tRNA(Ala)(GGC) to ensure accurate decoding. Nat Struct Mol Biol 16:359–364

Lehman N, Joyce GF (1993) Evolution in vitro of an RNA enzyme with altered metal dependence. Nature 361:182–185

Lozupone C, Changayil S, Majerfeld I, Yarus M (2003) Selection of the simplest RNA that binds isoleucine. RNA 9:1315–1322

Majerfeld I, Yarus M (2005) A diminutive and specific RNA binding site for L-tryptophan. Nucleic Acids Res 33:5482–5493

Massey SE (2008) A neutral origin for error minimization in the genetic code. J Mol Evol 67:510–516

Massey SE (2016) The neutral emergence of error minimized genetic codes superior to the standard genetic code. J Theor Biol 408:237–242

Massey SE (2019) Genetic code error minimization as a non-adaptive but beneficial trait. J Mol Evol 87:4–6

Mathew DC, Luthey-Schulten Z (2008) On the physical basis of the amino acid polar requirement. J Mol Evol 66:519–528

Moazed D, Noller HF (1990) Binding of tRNA to the ribosomal A and P sites protects two distinct sets of nucleotides in 16 S rRNA. J Mol Biol 211:135–145

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

Patterson C (1956) Age of meteorites and the Earth. Geochim Cosmochim Acta 10:230–237

Polyansky AA, Hlevnjak M, Zagrovic B (2013) Proteome-wide analysis reveals clues of complementary interactions between mRNAs and their cognate proteins as the physicochemical foundation of the genetic code. RNA Biol 10:1248–1254

Ronneberg TA, Landweber LF, Freeland SJ (2000) Testing a biosynthetic theory of the genetic code: fact or artifact? Proc Natl Acad Sci USA 97:13690–13695

Turk RM, Illangasekare M, Yarus M (2011) Catalyzed and spontaneous reactions on ribozyme ribose. J Am Chem Soc 133:6044–6050

Turk-Macleod RM, Puthenvedu D, Majerfeld I, Yarus M (2012) The plausibility of RNA-templated peptides: simultaneous RNA affinity for adjacent peptide side chains. J Mol Evol 74:217–225

Vawter L, Brown WM (1993) Rates and patterns of base change in the small subunit ribosomal RNA gene. Genetics 134:597–608

Vestergaard B, Van LB, Andersen GR, Nyborg J, Buckingham RH, Kjeldgaard M (2001) Bacterial polypeptide release factor RF2 is structurally distinct from eukaryotic eRF1. Mol Cell 8:1375–1382

Vetsigian K, Woese C, Goldenfeld N (2006) Collective evolution and the genetic code. Proc Natl Acad Sci USA 103:10696–10701

Woese CR (1965) Order in the genetic code. Proc Natl Acad Sci USA 54:71–75

Woese CR (1967) The genetic code: the molecular basis for genetic expression. Harper & Row, New York

Woese CR, Dugre DH, Saxinger WC, Dugre SA (1966) The molecular basis for the genetic code. Proc Natl Acad Sci USA 55:966–974

Wong JT-F (1975) A co-evolution theory of the genetic code. Proc Natl Acad Sci USA 72:1909–1912

Xu J, Appel B, Balke D, Wichert C, Muller S (2014) RNA aminoacylation mediated by sequential action of two ribozymes and a nonactivated amino acid. ChemBioChem 15:1200–1209

Yarus M (1982) Translational efficiency of transfer RNA's: uses of an extended anticodon. Science 218:646–652

Yarus M (2011) The meaning of a minuscule ribozyme. Philos Trans R Soc Lond B Biol Sci 366:2902–2909

Yarus M (2016) Biochemical refinement before genetics: chance utility. J Mol Evol 83:89–92

Yarus M (2017a) Efficient heritable gene expression readily evolves in RNA pools. J Mol Evol 84:236–252

Yarus M (2017b) The genetic code and RNA–amino acid affinities. Life 7:13

Yarus M (2018) Eighty routes to a ribonucleotide world; dispersion and stringency in the decisive selection. RNA 24:1041–1055

Yarus M, Christian EL (1989) Genetic code origins. Nature 342:349–350

Yarus M, Caporaso JG, Knight R (2005) Origins of the genetic code: the escaped triplet theory. Annu Rev Biochem 74:179–198

Yarus M, Widmann JJ, Knight R (2009) RNA-amino acid binding: a stereochemical era for the genetic code. J Mol Evol 69:406–429

Zavialov AV, Hauryliuk VV, Ehrenberg M (2005) Splitting of the post-termination ribosome into subunits by the concerted action of RRF and EF-G. Mol Cell 18:675–686