Evolution of the standard genetic code

Michael Yarus

Department of Molecular, Cellular and Developmental Biology

University of Colorado Boulder

Boulder, CO 80309-0347

Tel: 303 492-8376
Mobile: 303 817-6018
Email: yarus@colorado.edu

Keywords: coding table, codon, triplet, evolution, distribution fitness

Abstract

A near-universal Standard Genetic Code (SGC) implies a single origin for Earthly life. To study this unique event, I compute paths to the SGC, comparing different histories. SGC-like coding tables emerge, using traditional evolutionary mechanisms - a superior evolutionary route can be identified.

To objectively measure evolution, progress values from 0 (random coding) to 1 (SGC-like) are defined, measuring 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 was based on selected RNAs performing aminoacyl-RNA synthetase reactions. Acceptor RNAs exhibit SGC-like wobble; alternatively, non-wobbling triplets uniquely encode 20 amino acids/start/stop. Triplets acquire their 22 functions by stereochemistry, selection, coevolution, or randomly. Assignments also propagate to an assigned triplet's neighborhood via single mutations, but can decay.

SGC order is especially sensitive to disorder from random assignments. Futile evolutionary paths are plentiful due to the vast code universe. Evolution inevitably slows near coding completion. Coding likely avoided these difficulties, and two suitable pathways are compared in detail. In "late wobble", a majority of non-wobble assignments are made before wobble is adopted. In "continuous wobble", a uniquely advantageous early intermediate supplies the gateway to an ordered SGC. Revised coding table evolution (limited randomness, late wobble, concentration on amino acid encoding, chemically conservative coevolution with a simple elite) produces varied full codes with excellent joint progress values. A population of only 600 independent coding tables includes SGC-like members, and a Bayesian path to further refinement is available.

Introduction

The object of the investigation. Fig. 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 (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 values for polar requirement are not Woese's original chromatographic values, but these quantities corrected (Mathew and Luthey-Schulten 2008) by molecular dynamics distribution studies, which can circumvent chromatographic 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, 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. The SGC's division into a few coherent regions is 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 a protein (but see below).

To illustrate 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 same polar requirement scheme as Fig. 1A. Differences; in particular, the distinctive, pervasive chemical order of the SGC, are strikingly evident in the dissimilarity between Fig. 1A and 1B.

A simplified model. To investigate SGC appearance, we desire the fewest, least specific assumptions, in order to maximally respect limited knowledge of the early code. These are: 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 fully assigned coding table that resembles the SGC (**S**tandard **G**enetic **C**ode). Meaningful average rates of coding assignment, which includes both enabling mutation and ensuing events that fix the new meaning, are assumed to exist. Having said only this much, a great deal about code descent is implied.

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 similar 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 triplet positions; third position nucleotides presumably mutate as do other nucleotides. Instead similarity is attributable to wobble (Crick 1966), which assigns versatile base-pairing to third codon positions, allowing them to be read ambiguously by pairing with the same molecule. This allows the code to easily

expand to accommodate third position mutational variation, thus immortalizing many such easy SGC expansions (Fig. 1A). Thus the SGC itself indicates that its defining acceptors wobbled (Fig. 1A) because it so frequently uses wobble (Crick 1966) assignments.

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 broadly for chemically varied amino acids: for 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. Similar relations between chemically similar amino acids via second position change are quite frequent, as for Ser-Tyr (Fig. 1A).

The formative influence of mutational neighborhoods. These varied observations are consolidated by supposing that code evolution was guided by likely mutational pathways. A triplet with a given function might mutate to any other triplet related to it by a single mutation. Thus there are three possible triplets that might be captured for the same function 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). Such differences are observed for examples of RNA evolutionary change in vitro (Lehman and Joyce 1993), for most common transitions in rRNA (Vawter and Brown 1993), and for mitochondrial (Kumar 1996) and nuclear DNAs (Collins and Jukes 1994) in vivo.

Plausible primordial RNA acceptors. 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 this tiny 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 for sidechain-specific amino acid binding RNAs show that these exist, and require a minimum of 18-20 ribonucleotides (Yarus et al. 2005a; Yarus 2017b). Thus, regiospecific aminoacyl transfer requires a surprisingly simple center with only three conserved ribonucleotides. Ribonucleotides therefore are unexpectedly proficient at 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 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 its oligonucleotide acceptor, but would transfer multiple amino acids. The small aminoacyl-ribonucleotide product could base pair relatively specifically with a subset of codons (Illangasekare and Yarus 2012), paralleling its base pairs with the aminoacyl transfer center, and would thereby associate its triplet codon(s) with a varied 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 mutant aminoacyl-RNAs associate their amino acids with neighboring triplet(s), the event here termed mutational capture.

Further, early code expansion could add ribonucleotides 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.

Aminoacyl-RNAs. Thus, 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), and can be elaborated to add amino acid selectivity. Acquisition of new triplet specificities permits such an aminoacyl transfer center to readily explore its triplet neighborhood, capturing the nine codons related to it by single mutations.

Simple wobble. Early wobble coding must be minimal, independent of complex nucleotide modifications which can only arise later (Grosjean and Westhof 2016). To model wobble, I use a 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 I lists these and therefore allows visualization of mutational transitions, and therefore the evolutionary routes that simplified wobble coding most likely can follow. Thus, for example: one cannot assign XYA or XYC specifically; such functional triplets exist only as members of wobble pairs. When a wobble or non-wobble choice is made, as for codon XYU, wobble occurs with probability Pwob.

Table I – The simple 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**

Table I legend: **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**. Coding triplets are permitted only one acceptor.

Quantitative detection of evolutionary progress. To compare evolving coding tables, objective capture of differences like those between Fig. 1A and 1B 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 SGC grouping of identical functions because such 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; 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 mutational distances for the number of pairs, to calculate the mean distance/triplet pair. This makes spacing resilient when tables with varying unassigned triplets are compared. In 1000 random complete coding tables, identical functions are an average of 2.284 (mean) ± 0.002 (sem) mutations apart. The SGC (Standard Genetic Code) 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 thousand independent completely random tables average 2.286 ± 0.002 mutations from the SGC (per pair; Fig. 1A) by this measure. Random spacing and distance are further clarified in Methods.

Chemical order (dPR). The SGC shows exquisite, virtually complete 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 (Woese et al. 1966). Only neighborhood pairs that differ are counted and normalized for the number of comparisons. Thus, dPR does not overlap with mutational 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 of SGC proximity, they are used in a form which does not require comparison to other measured 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.

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

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, showing whether greater or lesser mean SGC likeness was attained. Thousands of such discriminations are the crux of the present inquiry.

Progress values respond to random assignments. To clarify progress values, Fig. 2 plots mean spacing, distance and dPR for groups of 250 full coding tables which have been constructed with known numbers of SGC assignments, then unassigned triplets filled with random assignments with no relation to the SGC. These coding table populations therefore are otherwise random, but have a specified fraction of SGC assignments - the latter fraction has been plotted across the Fig. 2 abscissa. 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 proximity to SGC order.

A useful model. A biologist is only slightly interested in average behavior of 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, the subset of tables that evolve to a finished code is of great interest. Finished tables assign all 64 triplets (becoming “full” tables) or encode all 22 functions (becoming “complete” tables).

Coding history is done via computation (described in Methods) that follows one coding table at a time to its particular fate. One triplet of 64 is randomly chosen. Subsequent events occur at random on the basis of probabilities for initial triplet assignment (Pinit), mutational capture of a nearby triplet by an assigned triplet (Pmut) or assignment decay (Pdecay). Using randomized numbers to choose chance events with specified probabilities, an unassigned chosen triplet can be allocated to one of the 22 essential functions (Pinit). If the randomly chosen triplet has already been assigned, it can capture new triplets for its function via mutation in its neighborhood (Pmut). Alternatively, its function can decay, with the triplet losing its previously assigned meaning (Pdecay). Probabilities are chosen to limit outcomes to a total probability of ≤ 1.0. Repeating these chance events can ultimately build a full (64-assignment) table or make complete (22 function) tables. With repetition, such computations can compare outcomes for different coding histories.

Kinetics using triplets chosen randomly. Because it is not usual to compare kinetics by performing a succession of random events, we first demonstrate that this procedure yields normal dynamic behavior. Fig. 3 shows velocities as initial triplet assignments, mutational captures and assignment decays as a function of the number of assigned triplets (randomly chosen from the SGC in each repetition). First order reactions should be proportional to reactant availability, second order reactions to the product of two reactant availabilities. Fig. 3 shows that these expectations are accurately met.

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 it extrapolates to the complete table's Pinit = 0.6), decreases linearly as triplets become assigned, and extrapolates near zero when 64 triplets are occupied, so that no opportunity for initiation exists. 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 seen.

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 similar assignments. Expansion of the code by mutational capture therefore should be second order, varying with the product: assigned*unassigned triplets. The data shows that the expected second order maximum capture rate is observed when half of triplets (32) are assigned. Moreover, the fitted rate extrapolates to zero both when assigned codons = 0 (at left) and also when unassigned codons = 0 (at right). Thus: mutational triplet capture behaves as a second order reaction. There is further quantitative support for this rate analysis in Methods.

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. To respect this difference, durations are expressed in "passages" (single 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 cited in legend), through their initial 4096 passages. There is an initial period of rapid change (≈ 0 - 500 passages), then a later near steady-state in which numbers of assigned and unassigned triplets change little. In that later era, total decays, initial assignments and mutational captures increase at almost constant rates, but no change in mean assigned/unassigned triplets. 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 we return to unassigned triplets that persist, and to stably incomplete coding, exemplified for Fig. 4A conditions as a mean of 60 steadily assigned and 4 unassigned triplets.

Fig. 4B, however, shows new, finished code behavior. Both transient and near-steady-state behavior appear for the full and complete coding tables of particular 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, complete codes, shape code evolution and accordingly, figure again below.

Steady state order. Now consider evolution of progress values. Fig. 4B shows spacing, distance, and dPR progress with increasing duration for Fig. 4A. 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, 4B). Finally, Fig. 4 evolution uses random initiations and random later mutational captures. Fig. 4B extensively documents the persistent random fate of such a table (progress values ≈ 0) throughout 4096 passages.

Thus, to evolve an SGC we must add source(s) of order to Fig. 4. This plan is initially implemented by assigning early triplets matching the SGC (as for stereochemical origins). Though initial stereochemistry ultimately proves insufficient, its failure suggests a successful path.

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

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

Coding tables initiated with 16 SGC assignments: wobble coding. Fig. 5A shows mean durations (passages) and the number of random assignments (inits) needed to reach particular numbers of encoded functions. Notably, both time and assignments needed to reach a specified wobble code complexity increase dramatically after 20 coding functions. In fact, starting at an initial mean of 12.35 functions (from 16 chance SGC triplets), encoding the last two functions costs 136-fold as much in time and assignments as do the first 20 encodings. This implies a history of great complexity - a complete wobble coding table has assigned triplets an average of ≈ 25000 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 (enumerated as inits 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. Finally, decay of assignments will be maximal near completion, also slowing completion.

Moreover, there are an 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 dilution, persisting indefinitely at ≈ 40% of SGC levels. However, wobble's spacing order occurs without parallel effects on SGC distance or on dPR, which descend to indistinguishability from random coding (line labeled 'random').

But 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 (compare Fig. 5B, 5D). Sixteen such wobble initiations immediately make spacing, distance, and dPR equivalent to the SGC, as we might expect.

Coding tables initiated with 16 SGC assignments: non-wobble coding. Fig. 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 duration is required. In addition, even if non-wobbling completion with 22 functions is mandated, only ≈ 1 additional assignment per triplet is required.

However, code completion again devastates progress values and coding order. Fig. 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 5B). 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 randomly-formed coding tables (Fig. 5D).

Wobble summary. Non-wobble completion is quicker and simpler than for a wobbling code, but code order supplied by initial SGC assignments 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 wobble assignments (Fig. 5B), and also mitigate related effects of a thousand-fold delay (Fig. 5A) during progress from a near-complete to a complete wobble code.

A key to completion complexity. Because dramatic delays are confined to the era between 20 encoded functions and 22 function completion (Fig. 5A), 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 translation initiation evolved late, after divergence of the major domains of life.

Moreover, 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 (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, 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 long-standing molecular support.

Rapid routes to wobbling codes. Fig. 6A shows that average coding behavior (as in Fig. 5) conceals a possible resolution of completion complexity. Fig. 6A plots the distribution of times to acquire 20 coded functions, for wobbling and non-wobbling codes, in successive 50-passage time 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) for 22 encoded functions. Modes, most probable completion times, do not actually differ 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 are taken instead of average ones (peak at left in Fig. 6A), we can evolve codes that wobble as does the SGC, but also appear quickly. Fig. 6B reinforces this discussion by showing that complete coding tables do not possess an early peak of completions. Adoption of 22-function coding as a goal makes rapidly completed coding tables rare (Fig. 6B), instead of frequent (Fig. 6A).

Quickly evolved wobble codes. There appear two short paths to a wobble code. By the first route, 20 functions are encoded without wobble, exploiting the easy access non-wobble coding has to nearly complete tables (Fig. 5C). Then, translation advances and wobble becomes possible. Wobble innovation is quickly adopted - pre-existing near-complete 20-function codes quickly add wobble wherever possible. We term this path "late wobble" for conciseness.

A second rapid route to SGC-like wobble coding, called "continuous wobble", allows wobble assignments (Table I) from the initiation of coding and throughout. This path seeks SGC access specifically from the early peak of 20-function wobble codes (Fig. 6A). An SGC via this minority of codes is nevertheless readily accessible, available in about ¼ of all evolutions (Fig. 6A).

A second barrier to SGC-like codes: coding table order. Consider progress values in Fig. 5B and 5D; their declines imply that evolution of the exquisitely ordered SGC (Fig. 1A) will require specific, persistent organizing influences. Therefore, we now compare ordering processes often cited for the SGC. Calculations below compare 6 ordering mechanisms utilizing coevolution and paralogous selection, adaptation and neutral mechanisms. These 6 mechanisms (termed Coevo, Coevo_PR, 0±1 PR, 0±2 PR, 0±3 PR, 0±4 PR, defined below) shape capture specificities for new triplets to get existing assignments.

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. An example is that RNA binding sites selected for amino acids contain cognate coding triplets with unusual frequency (Yarus 2017b).

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

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 assignment of related triplets by

minimizing the structural effects of substitution errors on protein structure (Freeland and Hurst 1998a). The SGC is, in fact, very unusual in its minimization of the cost of such errors (Freeland and Hurst 1998b).

Recently, a third **neutral** mechanism has been proposed (Massey 2008, 2016, 2019). Because successor RNA-amino acid interactions would likely be paralogous to 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 amino acids in the SGC. Descent of related RNA sequences for related amino acids also occurs within co-evolution and adaptation – this produces code order by means paralleling the neutral mechanism. Because co-evolution, 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. Such unification of the effects of co-evolution, selection and relatedness implements Crick's prescient comment that "similar amino acids would tend to have similar codons" (Crick 1968).

Encoding order. The above considerations, which determine functions assigned to neighborhood triplets captured for an existing assignment by mutation, have been embodied in code.

Coevolution. Related triplets are assigned to amino acids linked by synthetic pathways, as suggested by Wong (Wong 1975), but using later thermodynamic corrections (Ronneberg et al. 2000). Such assignments for the purpose of testing co-evolution theory 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 abbreviated Coevo.

Coevolution respecting PR chemical similarity. Here biosynthetically related triplet/amino acid assignments are made as for coevolution, but the synthetically related amino acid assignment that best conserves polar requirement is chosen with higher probability, rising as PR difference decreases. This mechanism is abbreviated 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 ordered 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. These chemically conservative, paralogical mechanisms are termed 0±1 PR, 0±2 PR, 0±3 PR and 0±4 PR.

Revised code evolution. I now react to Fig. 5A-5D by making several mechanistic alterations, as well as targeting 20-function codes to minimize completion complications.

Interspersed ordered assignments. 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,

5D) is shortened. The probability of random initiation is Prand; (1 – Prand) is the probability of SGC-like initiation and both occur throughout the same era.

First route: evolution to 20 functions, then late wobble. Fig. 7A shows total passages and initial coding assignments for late wobble, appearing after 20 non-wobbling functions are encoded (see "**Third position**", above; Fig. 1A). Notably, quick non-wobble evolution is retained: coding tables with 20 functions appear ca. 20 to 30 times faster than average with continuous wobble. Moreover, different assignment mechanisms under late wobble require few, and similar, initiations (≈ 0.78 assignments/ triplet). Thus, all 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 transferred to new triplets. Equivalence of late wobble coding histories supports evolutionary optimization using criteria other than overall rate.

Assignment mechanisms and approach to the SGC. We now compare varied ways coding assignments gain triplets during acquisition of 20 encoded functions (Fig. 7B). We also take a step toward more realistic discussion, plotting a quantity more relevant to code evolution than averages – the fraction of tables with spacing, distance and dPR progress ≥ 1.0. That is, the fraction of codes with progress values as good as, or better than, the SGC.

Firstly, the assignment modes Coevo, Coevo_PR, 0 ± 1 PR… are roughly similar with continuous and late wobble. Different assignment mechanisms faithfully reflect their individual rationales. Paralogy modes 0 ± 1 PR, 0 ± 2 PR, 0 ± 3 PR, 0 ± 4 PR are defined to conserve polar requirement, and their evolutionary effects reflect this definition. They indeed conserve chemical order better than coevolutionary modes. 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. 7B).

It also matters what assignment mechanisms neglect. Thus the outcome differs for spacing and distance. Conserving chemical order (dPR) alone ignores and 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 both biosynthetically related assignments and related PR, yields the most frequent mutual access to SGC-like spacing, distance and dPR together, though dPR is still most easily attained. For this reason, Coevo_PR is employed in further examples.

Close approach to the SGC depends on several conditions. SGC-like codes require multiple kinds of order. As shown for late wobble in Fig. 5B, all three progress indices decline immediately on random substitution in an otherwise SGC-like coding table. Because close SGC resemblance, as defined here, requires maintenance of all indices, the effect of random substitution on true SGC resemblance is yet greater.

In Fig. 7C, the mean disruptive effect on late wobble spacing, distance and dPR order is shown as a function of the probability of random substitutions (Prand) during acquisition of 20 functions. Balanced progress from Coevo_PR assignments (Fig. 6B) is disrupted by a minority of random assignments. Not many 20-function coding tables reach SGC levels of order, particularly for spacing and distance, with > ≈ 15% random (thus < 85% SGC-like) assignments. Thus, 10% random assignment was chosen for illustration in calculations above. Some random evolutionary assignment is allowable, but random

assignment > 15% is incompatible with an SGC-like result, particularly for spacing and distance order. This disruptive effect of random assignments first appeared in Fig. 2.

Fig. 7D depicts the effect of random assignments on evolved coding tables with all three progress values at SGC levels or better. Declines with randomness are faster, and these SGC-like tables (discussed below) are rarer than singly excellent tables in Fig. 7C. Fig. 7D reinforces the previous limit: random substitution in late wobbling SGC-like tables must be small, certainly <= 15%, better <= 10%.

Distributions for progress values. In Fig. 8A is shown spacing, distance and dPR progress for 2000 coding tables evolved by late wobble. All distributions are roughly symmetrical single peaks and so well described by means and standard errors used here. Almost all evolved coding tables have progress index distributions highly shifted from random assignment (0.0), toward the SGC, indicating overall effectiveness for this evolutionary route.

These full distributions are consistent with the cross section shown above for 10% randomness in Fig. 7C: distance is the sharpest peak and the fewest examples at or exceeding SGC behavior. Spacing is broader and has an intermediate foot at or beyond the level of SGC grouping of identical functions. The broadest and least symmetrical distribution is for dPR, polar requirement resemblance among closely related triplets. As one result, grouping of related PRs is the most frequent kind of evolved order, as calculated previously in Fig.7C.

Moreover, Fig. 8A shows why coding tables with all progress ≥1 are very sensitive to code order. Highly ordered tables are in the upper tails of three progress distributions; fraction of evolved tables with three progress values ≥1 will therefore vary rapidly and non-linearly when change in history shifts or spreads underlying spacing, distance and dPR distributions (Fig. 8A), even slightly

Thus, the joint distribution of all progress values is a more comprehensive indicator of progress. This mathematical observation also implements the biological goal; a code with grouped assignments, SGC-related and chemically ordered. In Fig. 8B the fraction of all progress greater than the abscissa value is plotted. For example, 50% of evolved coding tables have spacing, distance and dPR simultaneously >= 0.7. To illustrate how these statistics represent the SGC, coding tables within the rightward blue bar are shown in Fig. 9. These examples were picked from 600 successively evolved random tables: the best available progress is shown, and then tables with indices around 1.0, 0.95 and 0.9 to illustrate the meaning of differing joint distributions.

Evolved coding examples. Because evolutionary outcomes take a stochastic range (Fig. 8B), we must now grapple more fully with varied coding outcomes. Here are coding tables selected from an initial 600 examples, in descending order of joint progress values: SGC-like initiations, random assignment 10%, late wobble, and Coevo_PR controlling related triplet assignments. The most ordered code cannot be accurately placed because there are not comparable tables to estimate its real frequency; real frequencies for 1.0, 0.95 and 0.9 examples can be computed from positions in the observed joint distribution (Fig. 9).

These tables exemplify the use of progress indices to characterize less-than-SGC order. For example, comparison to the SGC shows that tables 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 8B. 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 is equivalent to or better than Fig. 9D , which is ≈ 90% the mutational and chemical distance from random coding to the SGC.

Other non-trivial implications appear from Fig. 9. The frequency of coding tables with spacing ≅ distance ≅ dPR ≥ 1 is low (Fig. 7D). Thus, orderly coding tables are not a subset having uniformly favorable properties; instead, progress values vary individualistically. Nevertheless, the observed joint distribution is promising: coding tables ranging up to the most SGC-like of Fig. 9 are not exceedingly rare.

The second route to an SGC: evolution to 20 functions with continuous wobble. 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, for reasons explained shortly. 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 similar to late wobble. As Fig. 10A shows, distributed joint progress for continuous wobble early is very similar to joint progress for late wobble in Fig. 8B. There is little to choose between the two wobble pathways on this basis.

Assignment effects for continuous wobble. Order due to various kinds of mutational capture (Fig. 10B) also varies similarly to that for late wobble (Fig. 7B). 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.

Sensitivity to random assignment. The sensitivity of continuous wobbling to random (rather than SGC) assignments is again pronounced (Fig. 10C). All three progress values decline rapidly, with spacing and distance approaching random assignment at Prand >≈ 0.15, resembling the late wobble evolutionary response (Fig. 7C). Moreover, joint overall order for continuous wobble is particularly sensitive to random assignments (Fig. 10 D), as it was for late wobble (Fig. 7D), because of the similar requirement for simultaneous upper-tail behavior in three distributions. Thus, predominant SGC-like initiations (Fig. 2, Fig. 7D) are not unique to late wobble, but are required for continuous wobble also (Fig. 7A).

A highly significant difference between continuous and late wobble. But late and continuous wobble coding differ. This can be seen by comparing example codes from 600 (Fig. 11) to parallel examples of late wobble (Fig. 9). Fig. 11 illustrates the best order observed in a continuous wobbling population of 600 (three progress values >= 1), and also joint progress ≈ 1, 0.95 and 0.9.

More frequent unassigned triplets (black with white dashes) among continuous wobbling tables (Fig. 11) are apparent, compared to late wobbling (Fig. 9). These example tables were chosen to illustrate joint progress, but greater prevalence of unassigned triplets is not produced by human choice. Average unassigned triplets in 1000 continuous-wobbling examples was 19.9 ± 0.1 (sem), whereas 6.5 ± 0.1 triplets were unassigned in a parallel sample of late-wobbling coding tables. Thus, evolution of almost-complete 20 function codes via continuous wobble leaves about a third of amino acid triplets

unassigned. Via late wobble, unassigned triplets are many fewer, more proportionate to coding needed for the small number of yet-to-be-encoded functions.

Continuous wobble assignments have been favored. In evaluating Fig. 10 and 11, recognize that continuous wobble evolution has been modified to *reduce* unassigned triplets. Assignment decay (Fig. 10, 11) has been reduced ten-fold and elective wobble has been increased: Pwob = 0.9. Finally, data are taken at 200 passages. Near-complete coding peaks at 171 passages: 200 passages augment triplet assignment. These measures do increase codon assignment: together they add a mean of 4.3 assigned triplets at 200 passages. However, Fig. 11's 200-passage coding tables are near steady state (compare unassigned triplets in Fig. 4A) - unassigned codons will never decrease substantially from levels shown. With increased assignment, continuous wobble steadily yields 2.6-fold more unassigned triplets than late wobble. With usual probabilities, the steady difference is ≈ 20 unassigned (continuous wobble) to ≈ 6.5 unassigned (late wobble), a 3-fold excess of unassigned codons for continuous wobble.

Discussion

The major implication. For this work, 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 coding table itself, 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 evolves easily, with no requirement for exotic events.

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

Wobble is inevitable in code descent. Wobble's capture of third position variation 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 characteristic closely-spaced identical assignments (Fig. 5B).

Coding with unique assignments, 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 shortened random-assignment era (Fig. 5D).

Two wobble solutions. Non-wobble's advantageous evolutionary rate and wobble's ordering effects can be combined supposing that wobble was delayed, but immediately adopted by preexisting codes when

it was made possible by translational advances (late wobble; Fig. 7A). Moreover, because coevolution has a milder disruptive effect on spacing and distance order (Fig. 7B, 10B), and dPR can be enhanced by favoring conservation of polar requirement in biosynthetically related amino acids (Fig. 7B, 10B), coevolution with polar requirement matching (Coevo_PR) during mutational capture balances the progress values of a late-wobbling coding table. Twenty encoded functions are targeted to reduce completion complexity and because initiation and termination have distinct, unconserved mechanisms. Such late wobble yields coding that attains order close to SGC levels (Fig. 9, 11).

The second route to prompt wobble coding exploits an atypical fraction of wobble codes completed very early (continuous wobble; Fig. 6A). These reproduce SGC order as well as do late wobbling coding tables (Fig. 10A), and also share similar sensitivity to random assignments (Fig. 10C, 10D). But while continuous wobble easily completes coding, it does not fill coding tables (Fig. 11A, 11B, 11C, 11D).

Sources of coding distribution. To further resolve the late wobble pathway, differences can be calculated between the average 20-function late-wobbling coding table and an exceptional subset with joint progress ≥ 0.9 (Fig. 8B). Such differences should objectively, quantitatively characterize more favorable routes toward the SGC.

Table II – Average 20 function late wobble coding versus an excellent subset

population	passages	decays	init	mut capt	wob init	assigned	spacing	distance	dPR
Mean late wobble	179	3.7	51.5	11.1	12.2	57.5	0.768	0.837	0.783
joint progress ≥ 0.9	113	1.9	42.5	6.1	15.5	55.2	0.98	0.96	1.00

Table II legend: Means for 2000 coding tables are shown down to the significant figure the order of its sem. Evolution employed 10% random assignment, mutational capture of neighborhood triplets with coevo_PR, and late wobble at 20 encoded functions. There were 82/2000 tables with joint progress ≥ 0.9 (Fig. 7). Tot init = initial assignments; mut capt = assignments conferred on neighborhood triplets, wob init = wobbles added after 20 fn are encoded; assigned = total assigned codons; spacing = fraction SGC distance between assignments; distance = fraction mutational distance to SGC assignments; dPR = mean difference in polar requirement over neighborhood triplets as a fraction of random-SGC distance.

Simplicity. Selection of superior joint progress (Table II) dramatically increases resemblance to the SGC, as expected: from a mean of 0.7 – 0.8 to near SGC 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 associated with half the mutational captures of an average code.

Initiations are also minimized, used more efficiently, in coding tables which become more SGC-like. Only 83% of average initial assignments occur in the excellent subset, and excellent codes reach 20 encoded functions in 63 % of the average duration.

Excellent codes arise by chance simple routes, which are faster to complete. The favorable shortening of evolution in Table II is a smaller version of the ten thousand-fold superiority of 20-function late wobble compared to complete 22-function continuous wobble coding (Fig. 7A, 5A). 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 to 0.8 assignments/triplet to reach a near complete, late-wobbling code seems wholly credible (Fig. 7A, Table II).

Unassigned triplets, late assignments, late wobble. Late unassigned triplets support late assignments, deep into coding table evolution (Table II). This in turn is consistent with the division of code history into two eras, with late-arising assignments of unique character for translation initiation and termination (Fig. 5A, 5C). 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, event in SGC-like coding tables (Table II). Because excellent SGC resemblance arises with fewer initiations of all kinds, more late wobble is used to fill in superior coding tables. In Table II, an average of 15.5 triplets are newly assigned when wobble is introduced to a superior 20-function code.

Extended conclusions. Table II further defines the choice of ≈10% random, Coevo_PR and late-wobbling coding histories. Given distributed results due to stochastic evolution (Fig. 8A), SGC-like coding will be selectively observed among quickly-appearing coding tables with the simplest possible histories: the fewest assignments, mutational captures, and decays, accompanied by the greatest additions during wobble implementation.

Code sensitivity to random assignments. SGC-like order requires that randomly assigned triplets be stringently limited in number (Fig. 2). This imposes a limit, <= 15% random assignment if one requires SGC-like regularity in either continuous wobbling (Fig. 10C) or late wobbling codes (Fig. 7C). Such a limit is essential because good spacing, distance and dPR must occur simultaneously (Fig. 1, Fig. 8A) to approach overall SGC order. It is therefore noteworthy that, all findings together, 1 of 24 late-wobbling coding tables, or 1 of 21 continuously-wobbling coding tables approach SGC properties and appearance (joint progress >= 0.9; Fig. 8B, 9, 10A, 11). Limited randomness is required for a combinatorial reason, considered next.

Finding the SGC: the combinatorial abyss. Required simplicity above 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 SGC (Standard Genetic Code) 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 probability that shrinks toward order 10^{-83} (non-wobbling) or 10^{-41} (wobbling), because pervasively ordered SGC-like tables (Fig. 1A) are a minute selection of total code configurations (Fig. 1B, Fig. 2). Alternatively, it is 1.43×10^{17} seconds 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 multiple codes/second, can the 24 to 66 order-of-magnitude disparity between a random code search and time available for searching be spanned, and an SGC found?

Accordingly, it 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 (Fig. 2, 7D, 10D). 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 assignments**", 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 (Fig. 9, 11).

Nevertheless, the combinatorial abyss awaits. Completion complications (Fig. 5A, 5C) 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 6B are surely lost to the hiss of randomness. Code sensitivity to random substitution (Fig. 7D, 10D) is a whiff of the combinatorial abyss.

Independent evidence for non-random initial assignments. Coding tables must emphasize SGC-like assignments (**Finding the SGC**, just above). A large amount of independent evidence supports such specific triplet association with cognate amino acids.

RNA binding sites. The most recent account (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 every case (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 central conclusion. Simplest sites with their triplets are so prevalent they become 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 forced. 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 combined, strongly support stereochemical interactions as a basis of primordial coding.

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.

Bioinformatics. Moreover, there are data suggesting that relations between amino acids and their cognate coding triplets are yet more general. Coding triplets in 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 for accessible surfaces of completely folded proteins (Beier et al. 2014).

Thus: five independent arguments and data of varied types, point to stereochemical SGC assignments. Such assignments are required to order the SGC (Fig. 7C, 10C), 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 relation consistent with such interactions.

Unfamiliar mechanisms in coding history. Present models include events not usually discussed. Wobble itself can strongly stimulate code order (Fig. 5B, 5D). Decays, reassignments and mutational captures are infrequently accorded key roles in coding history. But here, they are routine. Because their inclusion leads to the SGC (Fig. 9, 11), and they are chemically plausible, all can have had a role in SGC history. Moreover, each has a potential function: for example, reassignments allow recovery if non-specific initiations are inconsistent with SGC order (Fig. 1A, 1B). Of triplets assigned during average code history (Table II), 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. The most successful present coding history emphasizes specific initial assignments; that is, stereochemical interactions. But it also calls on order from wobble, coevolution and paralogy. Such a mixed basis presently seems required, 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, 5D). Each assignment capture mechanism makes a distinctive contribution to code order (Fig. 7B, 10B), thus mixed contributions are needed to reach the 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 to act on, implying pre-existing stereochemistry and/or minor randomness (Fig. 7C, 10C). The beautiful order of the SGC (Fig. 1A) does not rule out a heterogeneous origin.

A more definitive coding history. Late wobble with 85-90% SGC-like assignment, coevolution with polar requirement selection accurately locate the vicinity in which the SGC resides (Fig. 9, 11), 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 were analyzed here. Plausible but more complex possibilities have not been considered. For examples, the possibilities that amino acids were encoded in subsets (Grosjean and Westhof 2016). Such segmentation would be very consistent with **Plausible primordial acceptors** above, and should be tested. Perhaps transitions and transversions should be distinguished. 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, arguably 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, a more likely mechanism explains more aspects of the current SGC (Yarus et al. 2005b). Importantly, a hypothesis does not necessarily slowly become more plausible. Instead, it multiplies its probability when it explains independent aspects of the code. Such a "Bayesian convergence" rapidly reinforces a correct explanation. This criterion is particularly appropriate for events remote in time and scale, like the origin of the genetic code (Yarus et al. 2005b).

Convergence points to late wobble, which quickly yields excellent coding tables, almost complete and almost full (Fig. 9). By comparison, continuous wobble quickly creates similarly excellent, almost complete coding tables (Fig. 10), but not almost full ones (**A highly significant difference…** above). Late wobble at 20 encoded functions is almost sufficient to the SGC - it leaves 6.5 triplets unassigned, a bit more than meeting the requirement for later initiation and termination (Fig. 9). In contrast, continuous wobble requires yet-unspecified ways to assign 20 triplets (Fig. 11).

Distribution fitness exploits primordial fluctuations. SGC evolution could be said to exploit 'distribution fitness'; that is, fitness entirely dependent on the distribution. A rigorous requirement is met by a heterogeneous group with excellent upper-tail members (Fig. 10A). Thus, undirected primordial variation is not a barrier; instead it 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 just beginning a reaction. Early starters have uniquely disperse product amounts – they are exceptionally well suited to simultaneous selection of their product and its inheritance, all in a prebiotic, gene-free chemical system. In fact, a new inherited chemical capability can emerge after only one selection, possibly requiring only a few days after partially activated ribonucleotides accidently encounter each other (Yarus 2017a, 2018).

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 can be

routine in a fluctuating milieu, that a desirable reactant is selected despite a 100-fold excess of a destructive competitor (Yarus 2016).

Thus: chance utility, starting bloc selection and distribution fitness solve notable evolutionary problems because primitive systems offer not specificity, but 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.

Computer 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, Ctable18b.pas, capable of all probability calculations presented here with slight adjustments, is available on request. Because of the speed of integer operations, coding tables were represented as arrays of integers. These were translated into ordinary coding tables (as in Fig. 1) using an alphabetically related dictionary, after evolutionary calculations. Runs with varied numbers of passages suggest that the ≈ 900-line program run as above requires about 4 µsec for one passage through a coding table (dependent on passage complexity).

Rate constants and probabilities. The kinetic method used in this work can be justified by showing that probabilities of reaction per computer transit are equivalent to normal rate constant formalism.

Initiations. The relation between P_{init} and the related first order rate constant, k_{init} in passages^{-1}, can be calculated by equating kinetic and probability equations for the overall rate of initiations/passage:

$$k_{init}(u) = P_{selection} * P_{reaction} = \frac{u}{64} * P_{init}$$

Where ***u*** is the number of ***u***nassigned triplets, and time is in passages. So

$$k_{init} = P_{init}/64$$

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

$$k_{decay} = P_{decay}/64$$

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:

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

$$k_{mut} = P_{mut}/448$$

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 significantly alter relevant results. Transition probability variations alter rates, but usually have smaller effects on order compared to effects discussed; thus such effects have usually not been included.

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:

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

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

$$spacing = 2.286 \pm 0.002(sem)$$

$$distance = 2.284 \pm 0.002\ (sem)$$

thereby validating randomization and calculation of mean mutational distances.

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

Figure Legends

Figure 1A, top. Color coding for polar requirement in Fig. 1. Each number indicates the midpoint PR for that color. That is: the 10.5 box spans 10.01 to 11.0.

Figure 1A, 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

Figure 1B. A randomized genetic coding table, in which each triplet was assigned to one of 22 functions by randomized number. Colors visually represent polar requirement, as in Fig. 1A. The table shown is representative of randomized tables: its progress values - Spacing = 0.009, Distance = -0.017, dPR = -0.195.

Figure 2. Progress values in randomly assembled coding tables. Computed mean spacing, distance and dPR progress values for 250 full coding tables assembled with the abscissa's fraction of random assignments, complemented with the remaining fraction of SGC-like assignments.

Figure 3. Rates of initiation, decay and mutational capture in coding tables with varied numbers of random SGC triplets. Measured mean rates (events/passage ± sem) are shown for 1000 randomly composed tables. Fitted least squares dotted lines portray expected kinetic behavior: decreasing linearly for initiation, increasing linearly for decay, and peaked quadratic for mutational capture. Decays and captures have been enlarged 10-fold to visualize them on the same scale as initiations. Pmut = 0.04, Pdecay = 0.04, Pinit = 0.6, Prand = 1.0.

Figure 4A. 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.

Figure 4B. Progress and finished non-wobbling coding tables. Means of 1000 spacing, distance and dPR progress values during the 4096 passages of Fig. 4A are plotted on the left ordinate, and the fraction of full and complete coding tables are on the right ordinate.

Figure 5A. Mean passages and initiations required for random wobble encoding of varied functions, after 16 random SGC wobble initiations. 1000 evolutions using random wobble coding and Coevo_PR mutational capture were stopped at the numbers of encoded functions shown on the abscissa. Lines labeled "init" indicate positions just after initial 16 random SGC triplets are chosen, but no evolution. Lines labeled "complete" indicate acquisition of all 22 encoded functions. Pmut = 0.04, Pdecay = 0.04, Pinit = 0.6, Pwob = 0.5, Prand = 1.0.

Figure 5B. Mean spacing, distance and dPR progress values for random wobble encoding after 16 random SGC wobble initiations. Data from calculations in Fig. 5A. **"**Init" and "complete" lines as in Fig. 5A. "SGC" line indicates SGC-like progress values, "random" line indicates progress for randomized assignments.

Figure 5C. Mean passages and initiations required for random non-wobble encoding, after 16 random SGC non-wobble initiations. 1000 evolutions using random wobble coding and Coevo_PR mutational capture were stopped at the numbers of encoded functions shown on the abscissa. Lines labeled "init" indicate positions just after initial 16 random SGC triplets are chosen. Lines labeled "complete" indicate acquisition of all 22 encoded functions. Transition probabilities as in Fig. 5A.

Figure 5D. Mean spacing, distance and dPR progress values for random non-wobble encoding after 16 random SGC non-wobble initiations. Data from calculations in Fig. 5C. Labeling as in Fig. 5B.

Figure 6A. Fraction of near-complete 20-function codes arising at varied durations. 1000 evolutions with Coevo_PR captures and 10% random assignment were applied to wobble and non-wobble coding. Probabilities except for Prand as in Fig. 5A. Signpost symbols are distribution means (171 and 4867 passages). All non-wobble evolutions are plotted, but 0.216 of wobble evolutions required longer durations and are off-scale to the right.

Figure 6B. Fraction of complete 22-function wobble codes arising at varied durations. 1000 evolutions with Coevo_PR assignments and Prand = 0.1 were applied to wobble coding. Other probabilities as for Fig. 6A. Signpost symbol is the distribution mean (630,000 passages). 0.142 of complete wobble evolutions required longer durations and are off-scale to the right.

Figure 7A. Duration and required initial assignments for 6 mutational capture histories. 2500 evolutions were averaged using late wobble coding to 20 functions, then all possible wobbles and Prand = 0.1 . Other probabilities as in Fig. 6A, bars are sems.

Figure 7B. Exceptional spacing, distance and dPR progress with 6 mutational capture histories. The fraction of individual progress values >= 1 (equal SGC or better) is shown for different mutational captures. Data is the same as in Fig. 7A. Bars are sems.

Figure 7C. Sensitivity of progress to random assignment. Spacing, distance and dPR progress values from 1000 late wobble evolutions with Coevo_PR assignments were averaged to get the fraction that progress to equal or better the SGC, using 0% to 30% random assignments (Prand from 0.0 to 0.3).

Figure 7D. Sensitivity of joint progress to random assignment. Fraction of 1000 late wobble evolutions with spacing, distance and dPR progress values simultaneously >= 1. Data is same as Figure 7C.

Figure 8A. Distributions of progress values with late wobble. Histograms of 2000 late wobble evolutions to 20 encoded functions with Prand = 0.1 and Coevo_PR assignments are shown. Lines marking randomized ("random") behavior and Standard Genetic Code ("SGC") behavior are supplied. Pinit = 0.6, Pdecay = 0.04, Pmut = 0.04, Pwob = 0.5.

Figure 8B. Joint distribution of spacing, distance and dPR progress values under late wobble. Data is that of Fig. 8A. "SGC" marks point where all progress values are simultaneously >= 1. Blue bar shows range of sample evolved coding tables in Fig. 9.

Figure 9. Sample coding tables from 600 late wobble, 20-function evolutions. The best, and sample codes having joint progress ≈1, ≈ 0.95 and ≈ 0.90 from late wobbling evolutions as in Fig. 8 are shown.

Figure 9A. 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 fully reproduced (compare Fig. 1A).

Figure 9B. Example of 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.

Figure 9C. Example of 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.

Figure 9D. Example of 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.

Figure 10A. Comparison of joint distribution of spacing, distance and dPR progress values under continuous and late wobble. Continuous wobble joint progress values (solid line) are compared to the joint distribution for late wobble (dashed line; from Fig. 8B). 1000 evolutions at 200 passages with Coevo_PR assignments and Prand = 0.1, Pinit = 0.6, Pdecay = 0.004, Pmut = 0.04, Pwob = 0.9.

Figure 10B. Exceptional spacing, distance and dPR progress, comparing 6 mutational capture histories. The fraction of individual progress values >= 1 (equal SGC or better) is shown for different mutational capture mechanisms, in 1000 continuous wobble evolutions at 200 passages. Other conditions and probabilities are as for Fig. 10A. Bars are sems.

Figure 10C. Sensitivity of continuous wobble progress values to random assignments. Mean progress values in 1000 evolutions for continuous wobble at 200 passages. Other conditions and probabilities are as for Fig. 10A. Bars are sems.

Figure 10D. Sensitivity of joint progress to random assignments, for continuous wobble. Conditions and probabilities are as for Fig. 10A. Bars are sems.

Figure 11. Sample coding tables from 600 continuous wobble evolutions. The best, and sample tables having joint progress ≈1, ≈ 0.95 and ≈ 0.90 from continuous wobbling at 200 passages, with probabilities as in Fig. 10, are shown.

Figure 11A. 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.

Figure 11B. Example of 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.

Figure 11C. Example of 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.

Figure 11D. Example of 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.

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**.

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**: 5257–63.

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, Sanchez G, Nickles T, Yarus M. 1995. Aminoacyl-RNA synthesis catalyzed by an RNA. *Science* **267**: 643–7.

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

Illangasekare M, Yarus M. 1999. Specific, rapid synthesis of Phe-RNA by RNA. *Proc Natl Acad Sci U A* **96**: 5470–5.

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

Johnson DB, Wang L. 2010. Imprints of the genetic code in the ribosome. *Proc Natl Acad Sci U A* **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**: 241–7.

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.

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**: 5482–93.

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

Massey SE. 2019. Genetic Code Error Minimization as a Non-Adaptive But Beneficial Trait. *J Mol Evol* **87**: 4–6.

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

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

Patterson, Clair. 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 U A* **97**: 13690–5.

Turk RM, Illangasekare M, Yarus M. 2011. Catalyzed and spontaneous reactions on ribozyme ribose. *J Am Chem Soc* **133**: 6044–50.

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**: 217–25.

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**: 1375–82.

Vetsigian K, Woese C, Goldenfeld N. 2006. Collective evolution and the genetic code. *Proc Natl Acad Sci U A* **103**: 10696–701.

Woese CR. 1965. Order in the genetic code. *Proc Natl Acad Sci U A* **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 U A* **55**: 966–974.

Wong JT-F. 1975. A Co-Evolution Theory of the Genetic Code. *Proc Natl Acad Sci U A* **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**: 1200–9.

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. 2018. Eighty routes to a ribonucleotide world; dispersion and stringency in the decisive selection. *RNA N Y N*.

Yarus M. 2017b. The Genetic Code and RNA-Amino Acid Affinities. *Life* **7**: 13.

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

Yarus M, Caporaso JG, Knight R. 2005a. Origins of the genetic code: the escaped triplet theory. *Annu Rev Biochem* **74**: 179–98.

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

Yarus M, Christian EL. 1989. Genetic Code Origins. *Nature* **342**: 349–350.

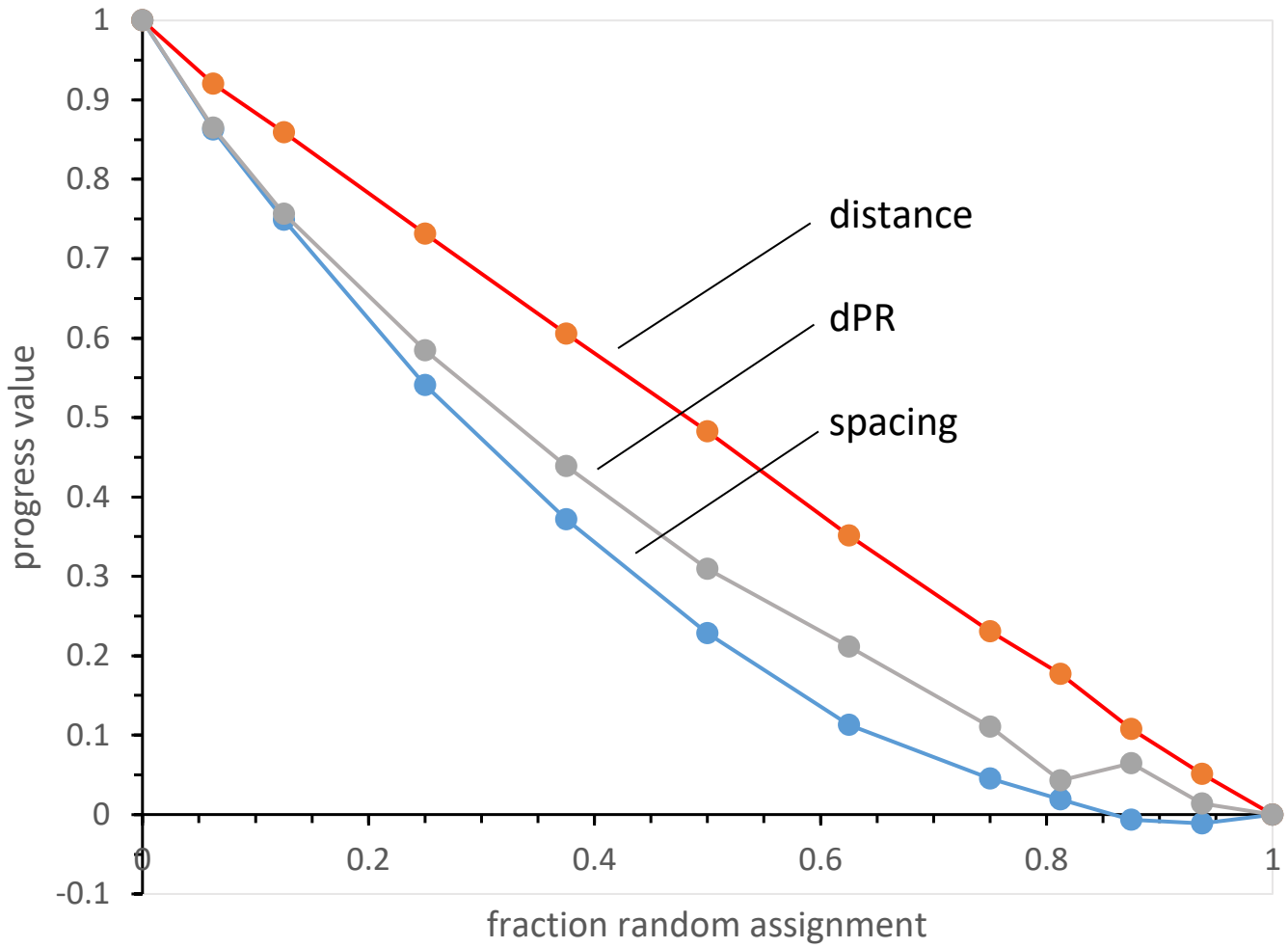
Yarus M, Widmann JJ, Knight R. 2009. RNA-amino acid binding: a stereochemical era for the genetic code. *J Mol Evol* **69**: 406–429.

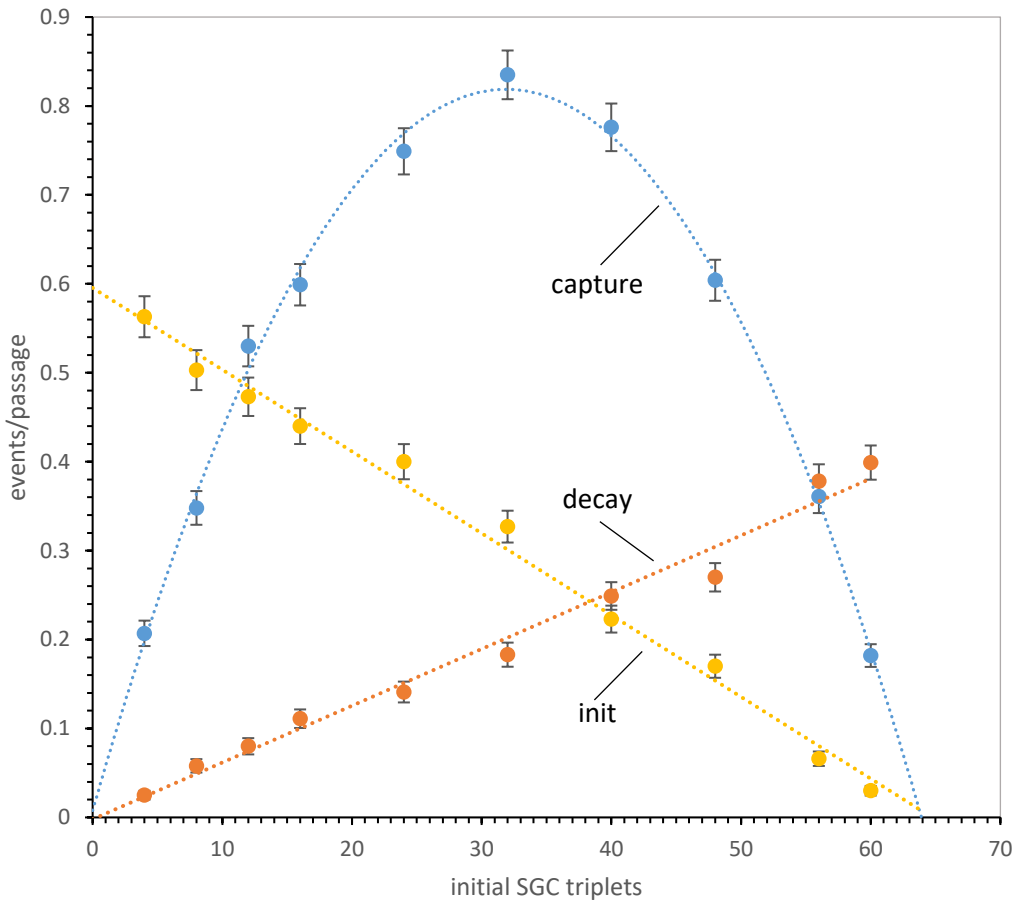
Zavialov AV, Hauryliuk VV, Ehrenberg M. 2005. Splitting of the posttermination ribosome into subunits by the concerted action of RRF and EF-G. *Mol Cell* **18**: 675–686.

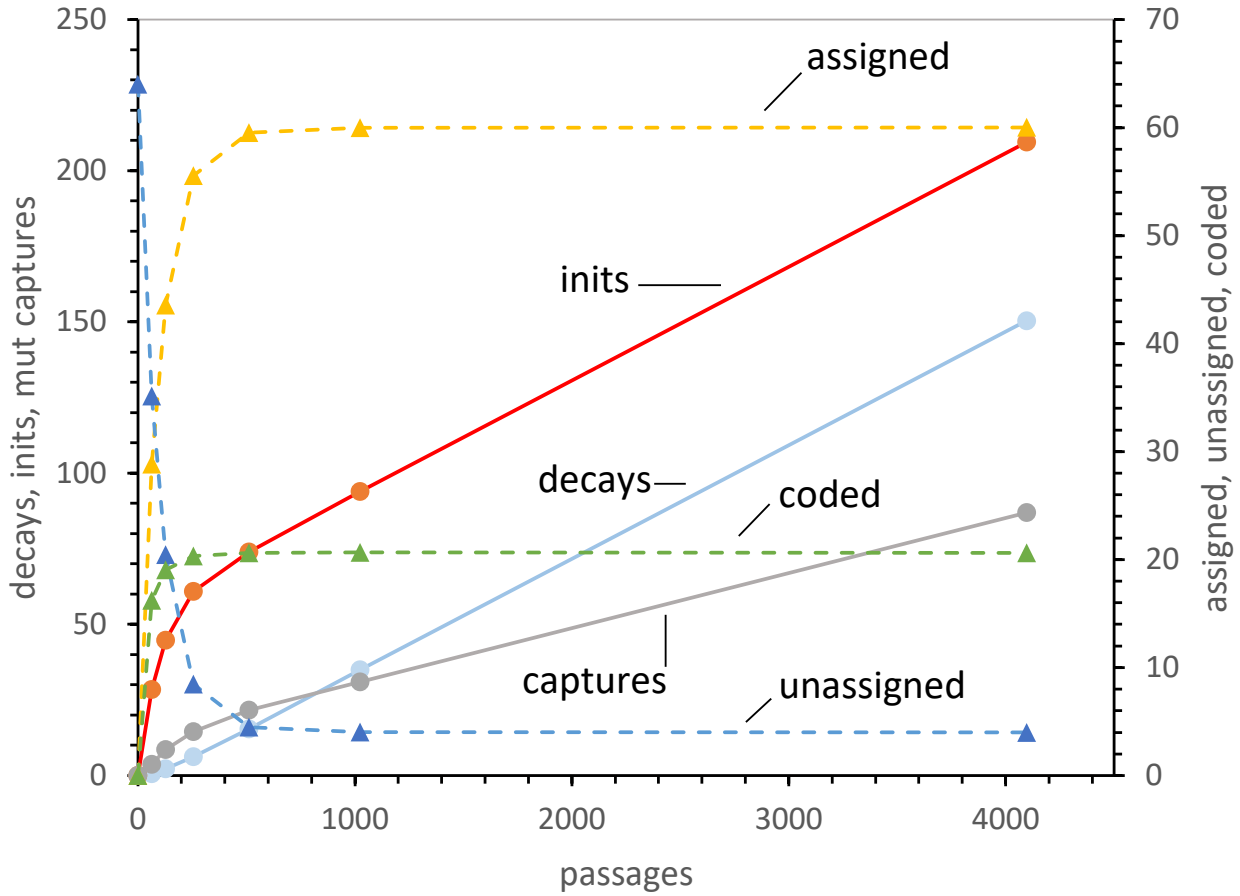
UUU	Phe (4.5)	*UCU*	Ser (7.5)	*UAU*	Tyr (7.7)	*UGU*	Cys (4.3)
UUC	Phe (4.5)	*UCC*	Ser (7.5)	*UAC*	Tyr (7.7)	*UGC*	Cys (4.3)
UUA	Leu (4.4)	*UCA*	Ser (7.5)	*UAA*	Ter	*UGA*	Ter
UUG	Leu (4.4)	*UCG*	Ser (7.5)	*UAG*	Ter	*UGG*	Trp (4.9)
CUU	Leu (4.4)	*CCU*	Pro (6.1)	*CAU*	His (7.9)	*CGU*	Arg (8.6)
CUC	Leu (4.4)	*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)

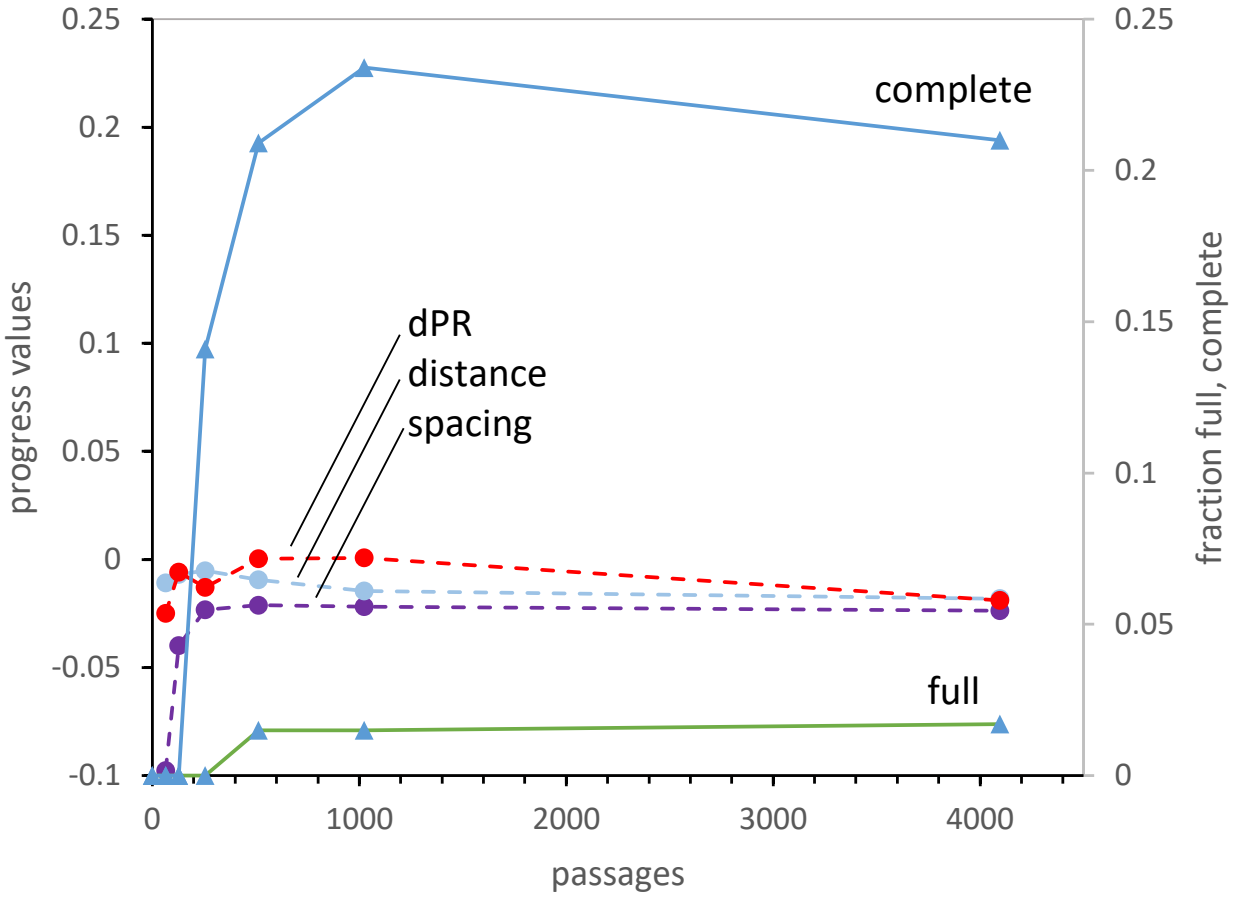
4.5	5.5	6.5	7.5	8.5	9.5	10.5	11.5	12.5	13.5

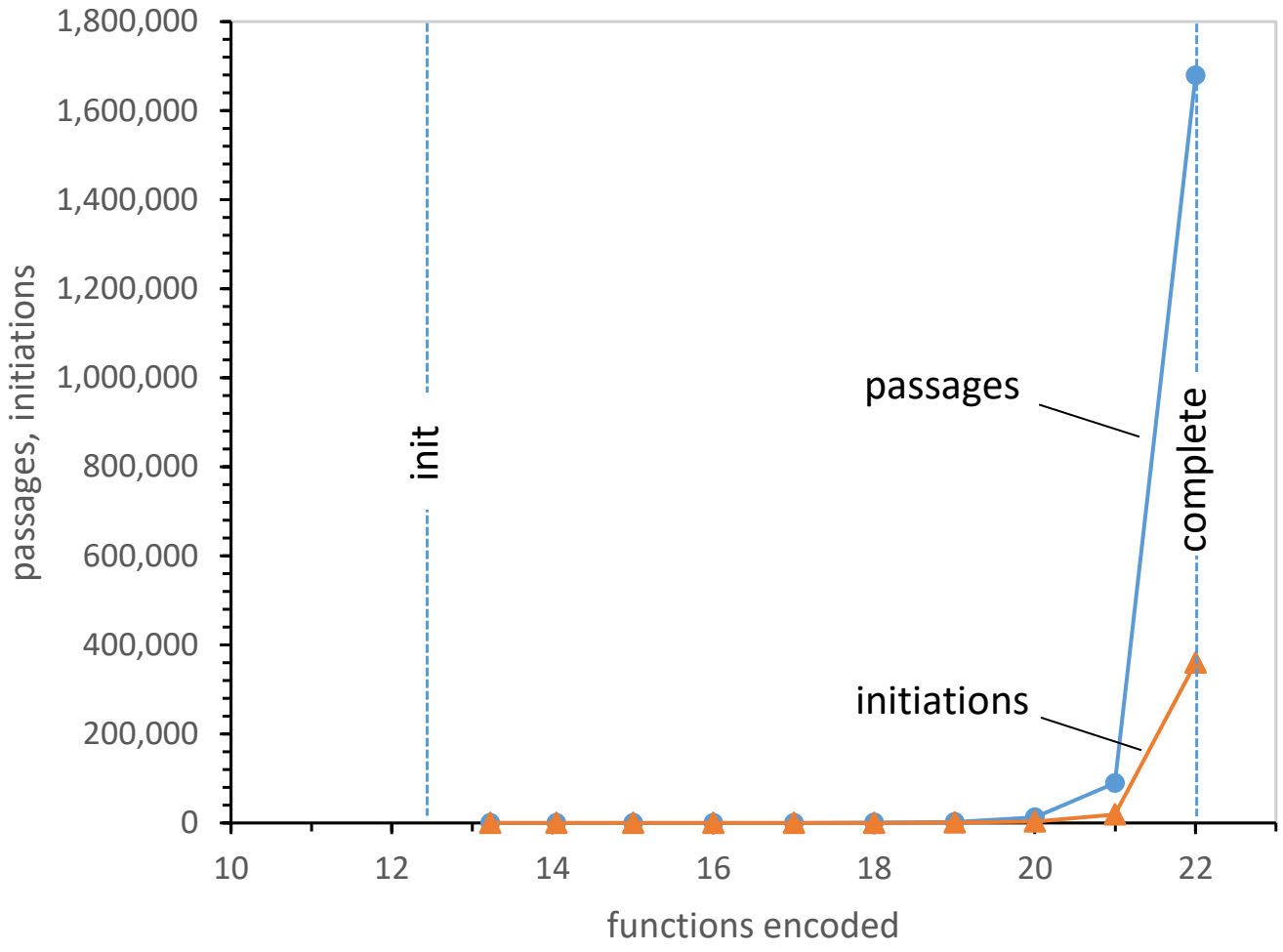
UUU	Met (5.0)	*UCU*	Arg (8.6)	*UAU*	Ter	*UGU*	Glu (13.6)
UUC	Asn (9.6)	*UCC*	Met (5.9)	*UAC*	Asn (9.6)	*UGC*	Arg (8.6)
UUA	Ile (5.0)	*UCA*	His (7.9)	*UAA*	Leu (4.4)	*UGA*	Val (6.2)
UUG	Asp (12.2)	*UCG*	Met (5.9)	*UAG*	Ter	*UGG*	Cys (4.3)
CUU	Gln (8.9)	*CCU*	Ile (5.0)	*CAU*	Val (6.2)	*CGU*	Leu (4.4)
CUC	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)

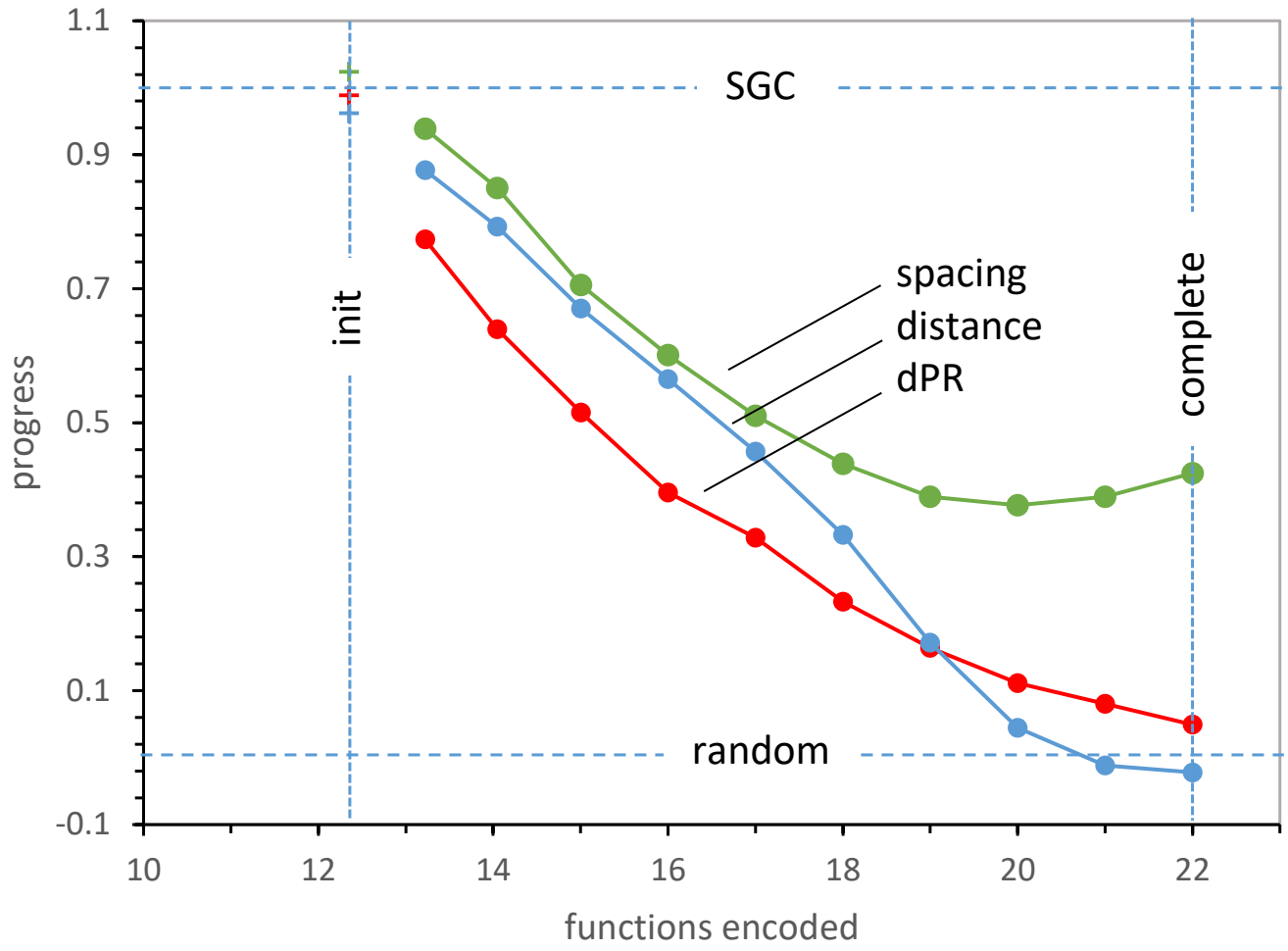

1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
progress value
0
0.2
0.4
0.6
0.8
1
fraction random assignment
distance
dPR
spacing

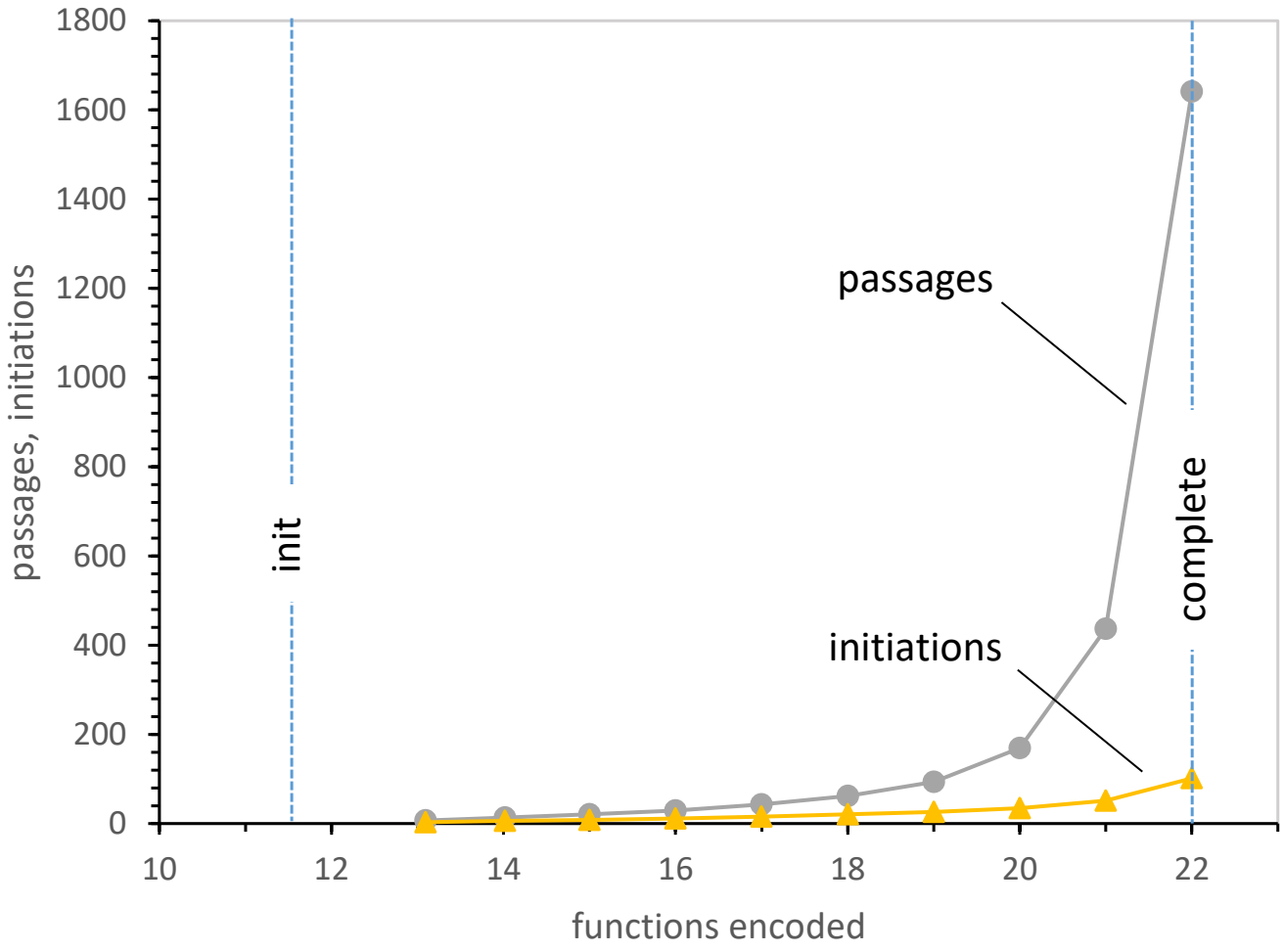

0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
events/passage
0
10
20
30
40
50
60
70
initial SGC triplets
capture
decay
init

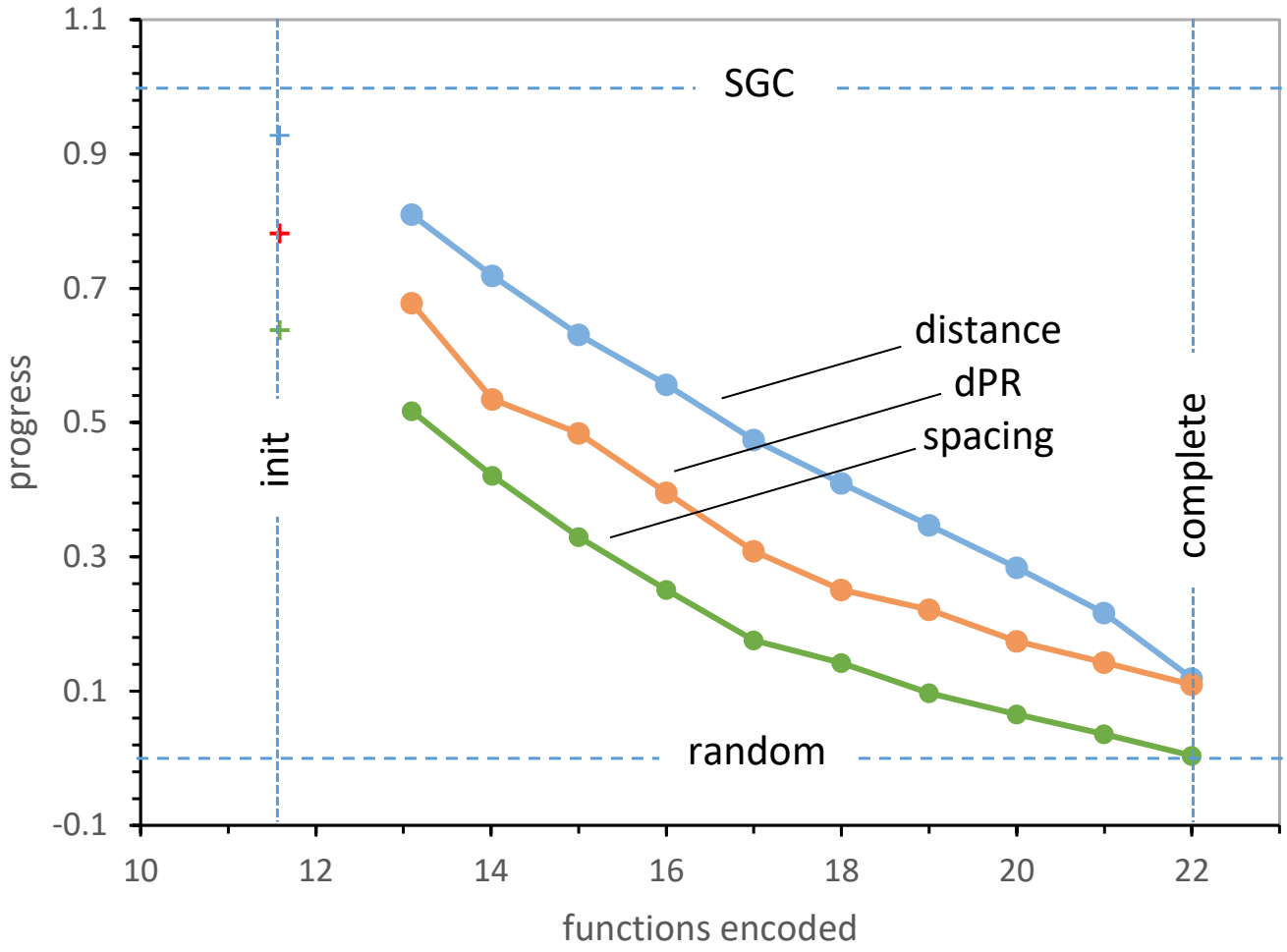

250
200
150
100
50
0
70
60
50
40
30
20
10
0
0
1000
2000
3000
4000
decays, inits, mut captures
assigned, unassigned, coded
passages
assigned
inits
decays
coded
captures
unassigned

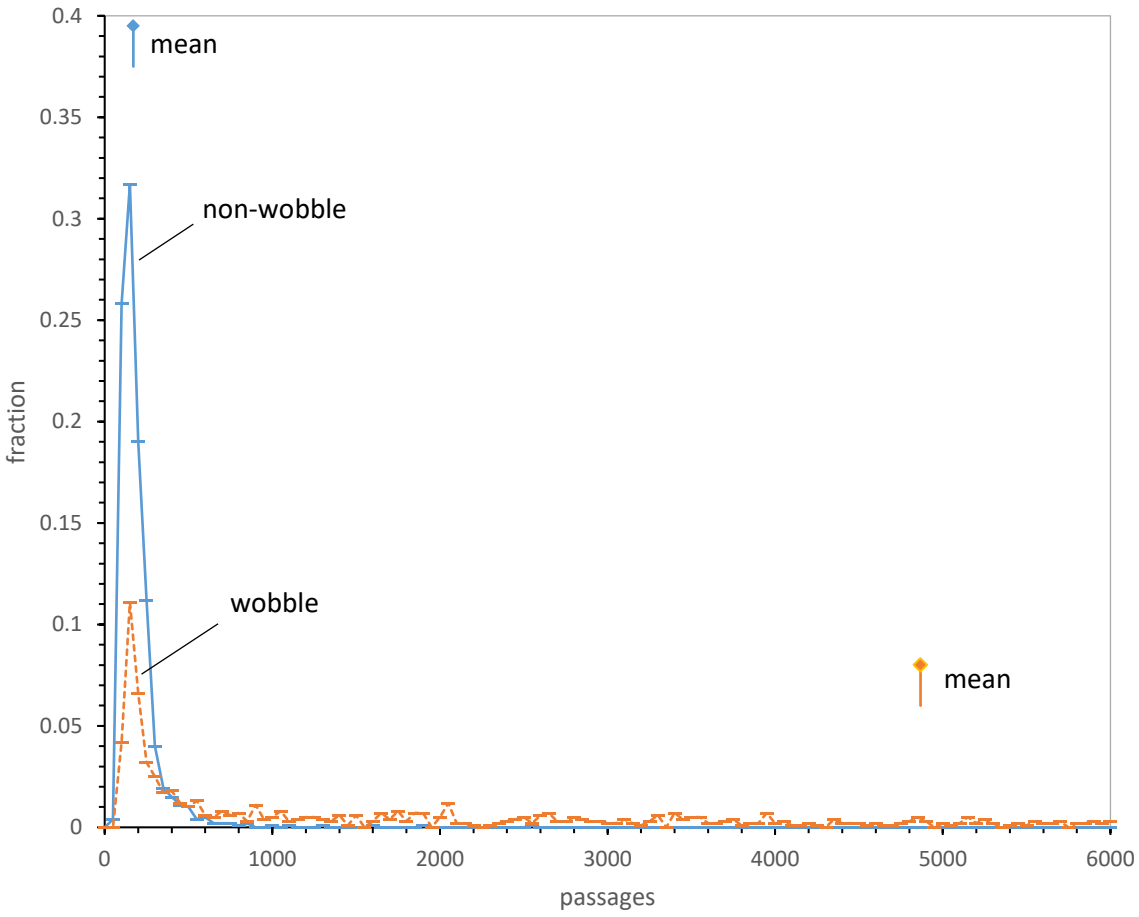

0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
progress values
0.25
0.2
0.15
0.1
0.05
0
fraction full, complete
0
1000
2000
3000
4000
passages
complete
dPR
distance
spacing
full

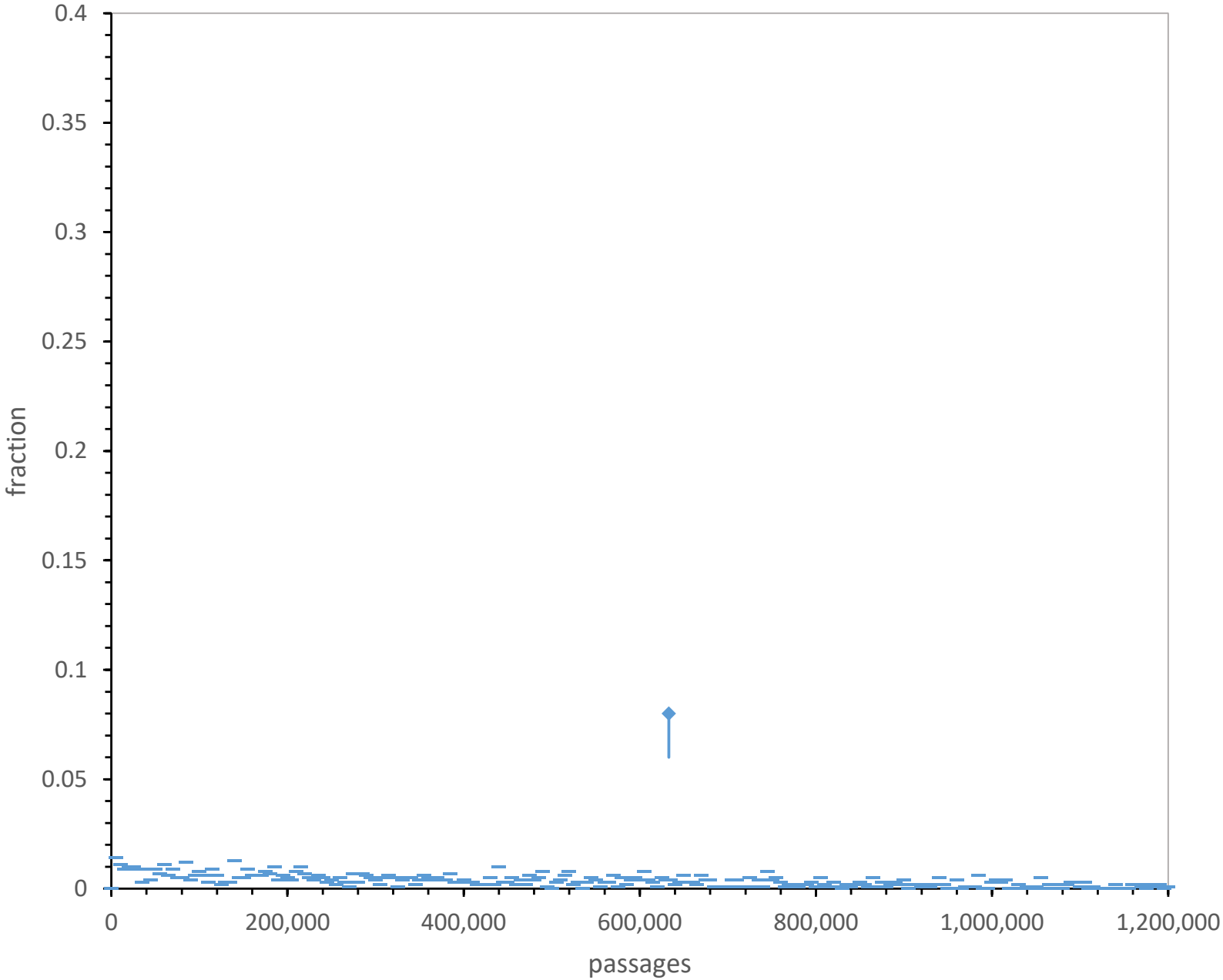

1,800,000
1,600,000
1,400,000
1,200,000
1,000,000
800,000
600,000
400,000
200,000
0
passages, initiations
10
12
14
16
18
20
22
functions encoded
init
complete
passages
initiations

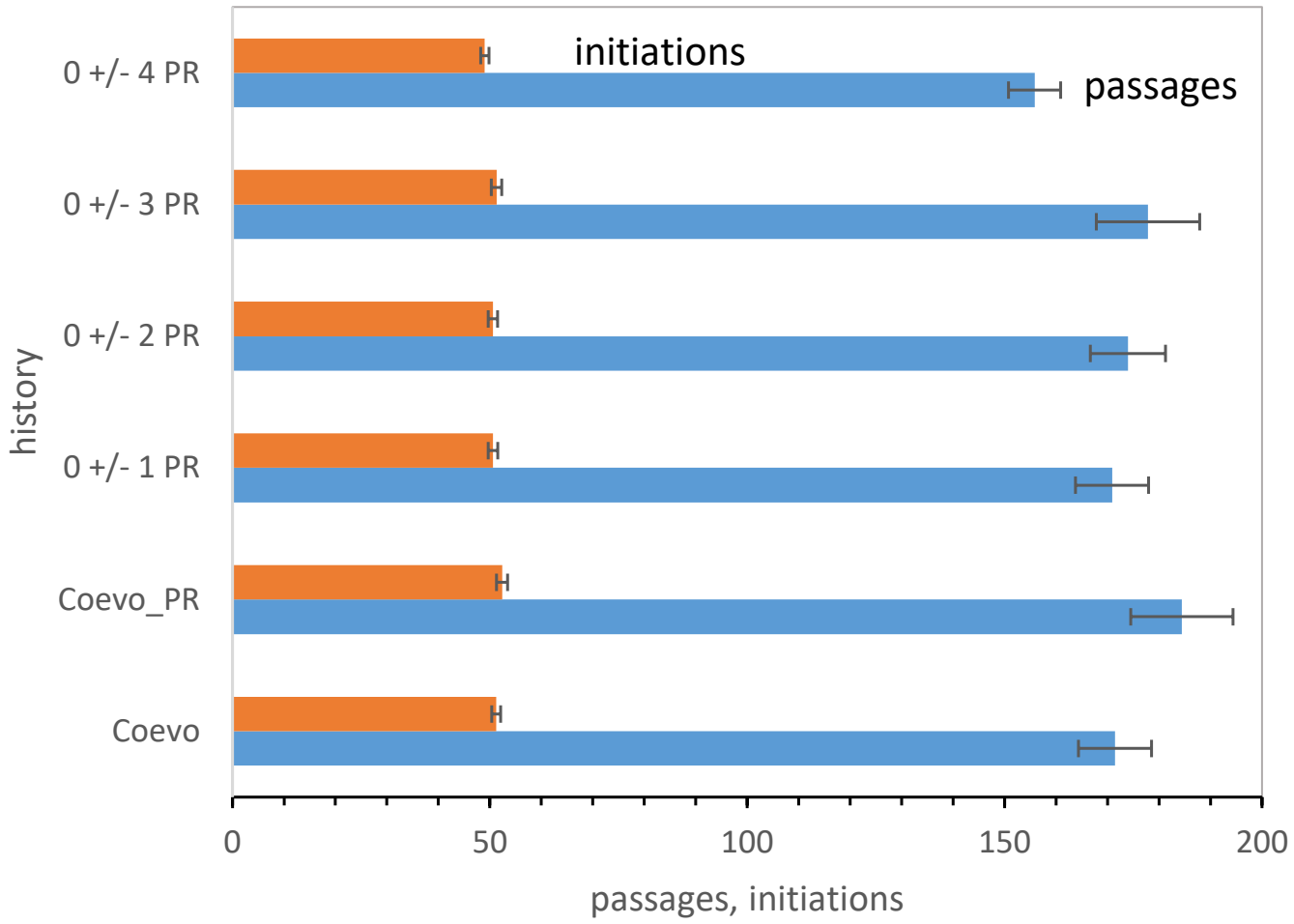

1.1
0.9
0.7
0.5
0.3
0.1
-0.1
progress
10
12
14
16
18
20
22
functions encoded
SGC
random
init
complete
spacing
distance
dPR

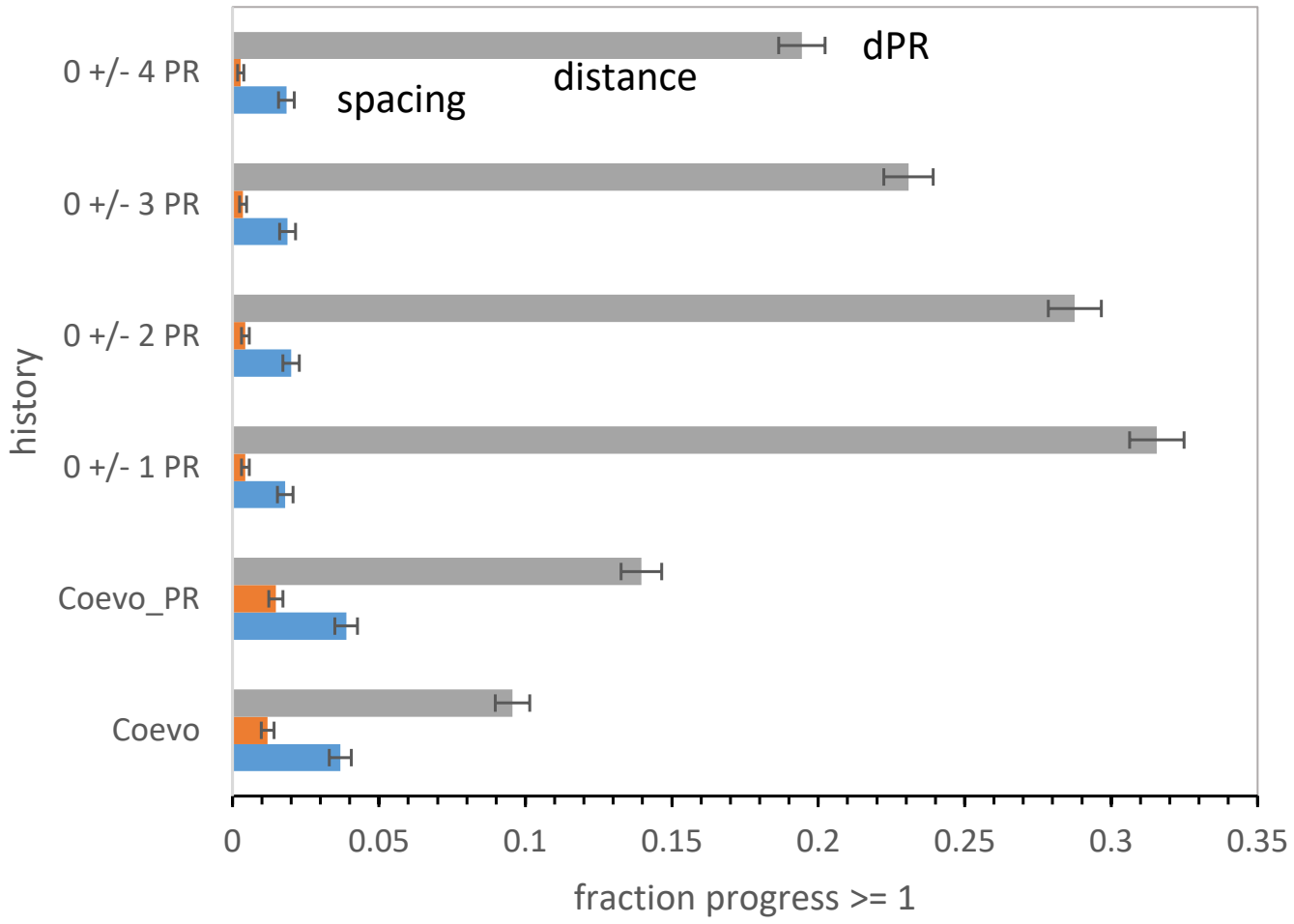

1800
1600
1400
1200
1000
800
600
400
200
0
passages, initiations
10
12
14
16
18
20
22
functions encoded
init
complete
passages
initiations

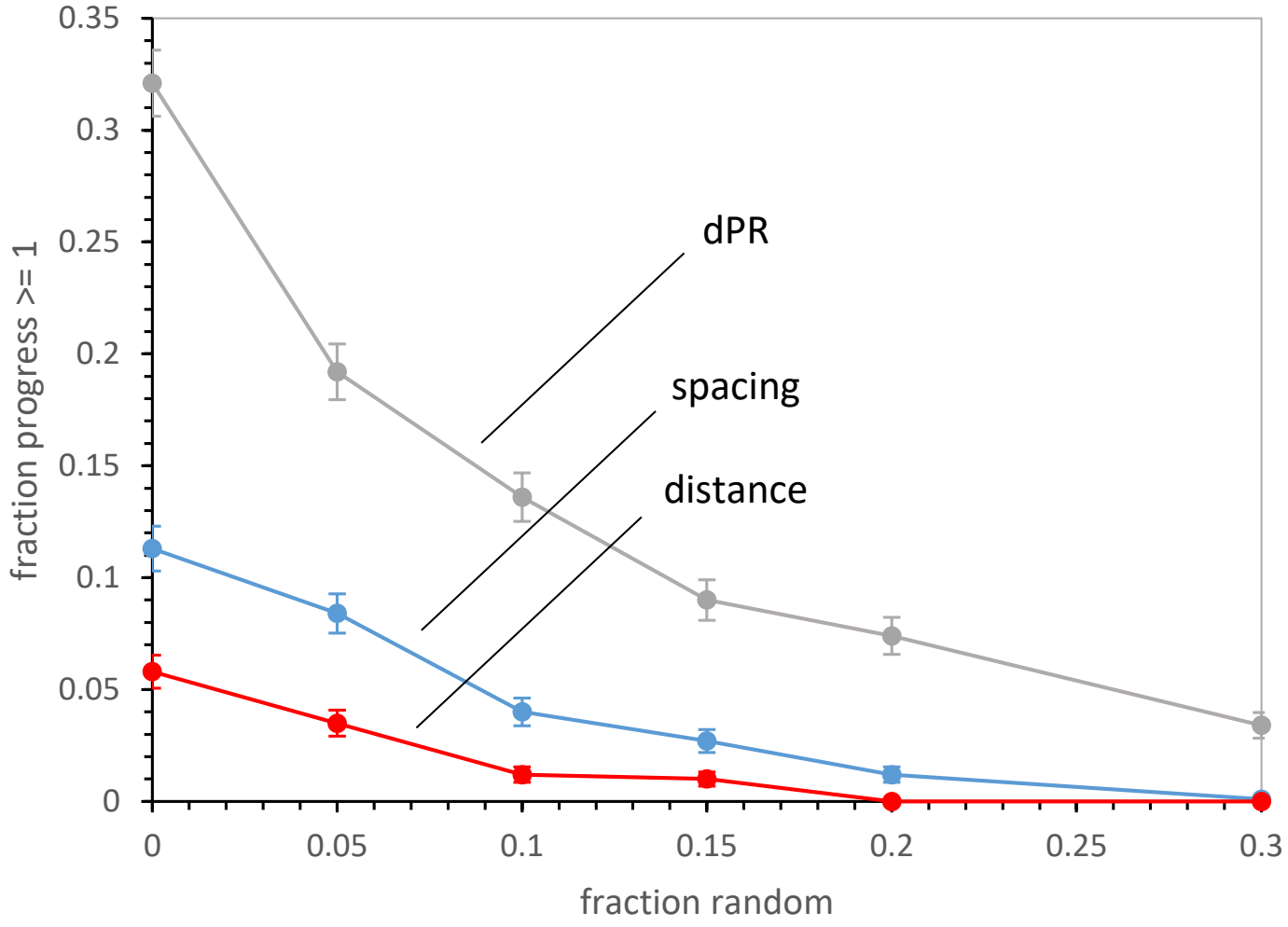

1.1
0.9
0.7
0.5
0.3
0.1
-0.1
progress
10
12
14
16
18
20
22
functions encoded
SGC
random
init
complete
distance
dPR
spacing

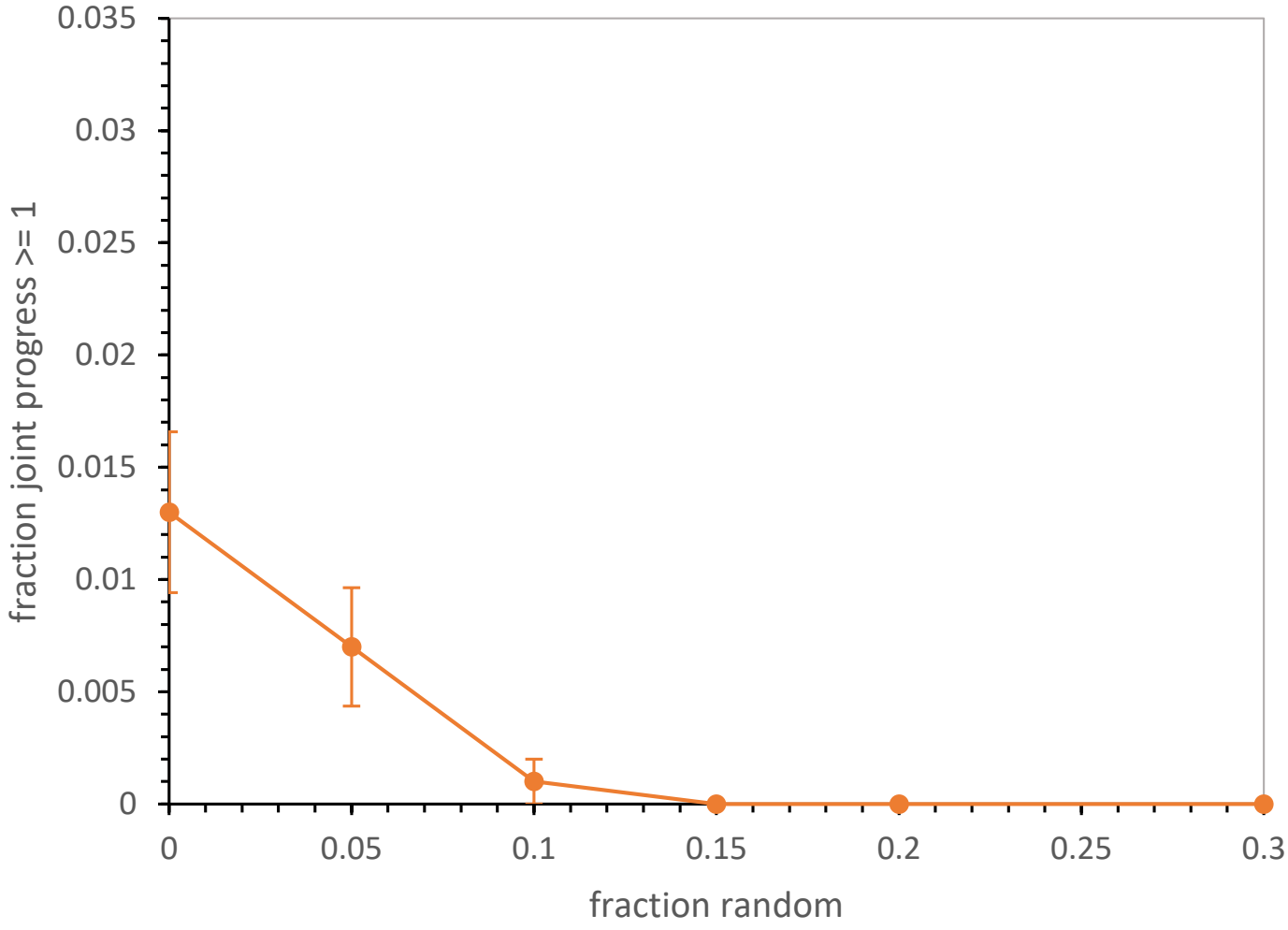

0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
fraction
mean
non-wobble
wobble
mean
0
1000
2000
3000
4000
5000
6000
passages

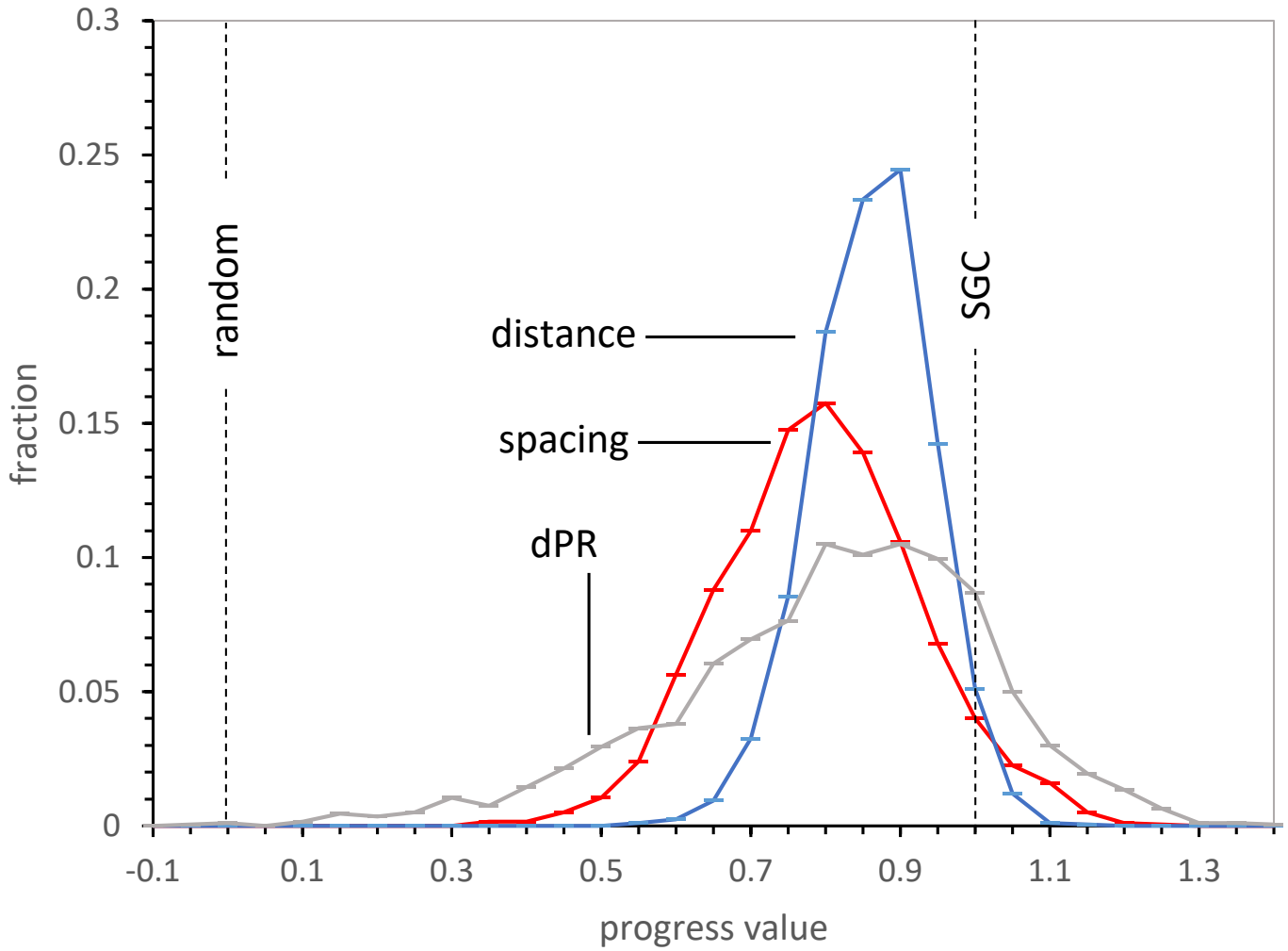

0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
fraction
0
200,000
400,000
600,000
800,000
1,000,000
1,200,000
passages

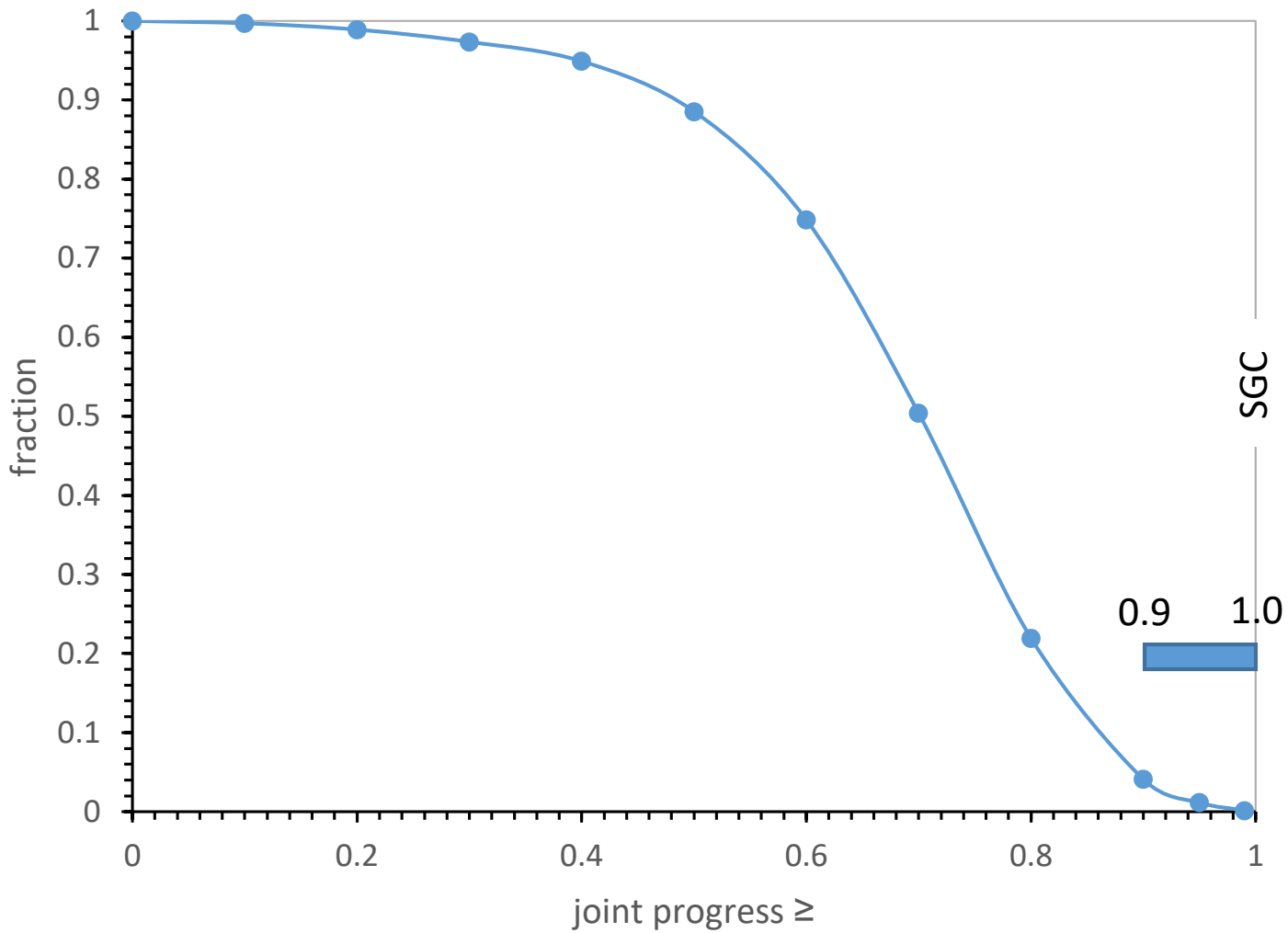

initiations
passages
0 +/- 4 PR
0 +/- 3 PR
0 +/- 2 PR
0 +/- 1 PR
Coevo_PR
Coevo
history
0
50
100
150
200
passages, initiations


dPR
distance
spacing
0 +/- 4 PR
0 +/- 3 PR
0 +/- 2 PR
0 +/- 1 PR
Coevo_PR
Coevo
history
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
fraction progress >= 1


0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
fraction progress >= 1
0
0.05
0.1
0.15
0.2
0.25
0.3
fraction random
dPR
spacing
distance


0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
fraction joint progress >= 1
0
0.05
0.1
0.15
0.2
0.25
0.3
fraction random


0.3
0.25
0.2
0.15
0.1
0.05
0
fraction
random
SGC
distance
spacing
dPR
-0.1
0.1
0.3
0.5
0.7
0.9
1.1
1.3
progress value

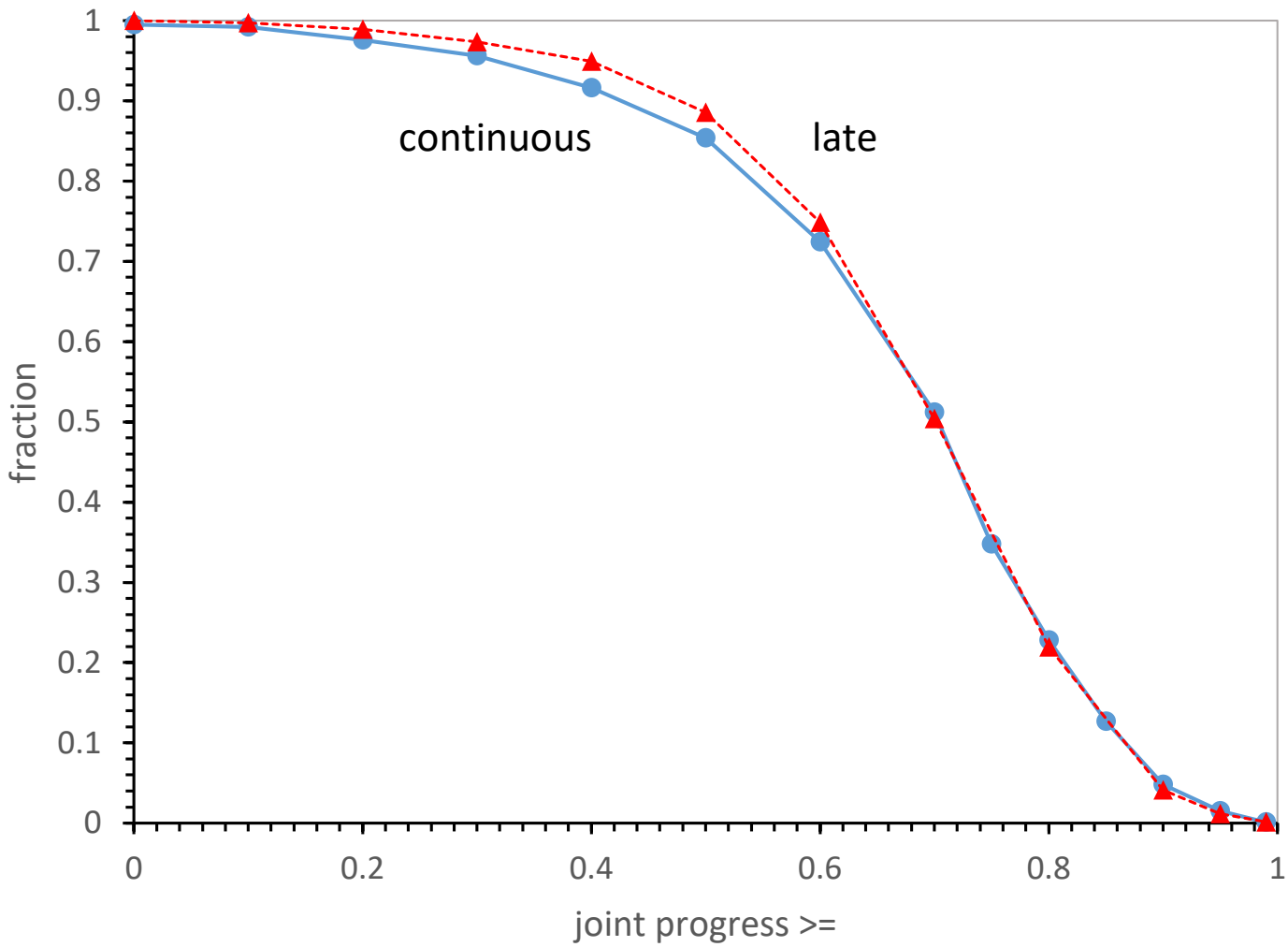

1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
fraction
0
0.2
0.4
0.6
0.8
1
joint progress ≥
SGC
0.9
1.0

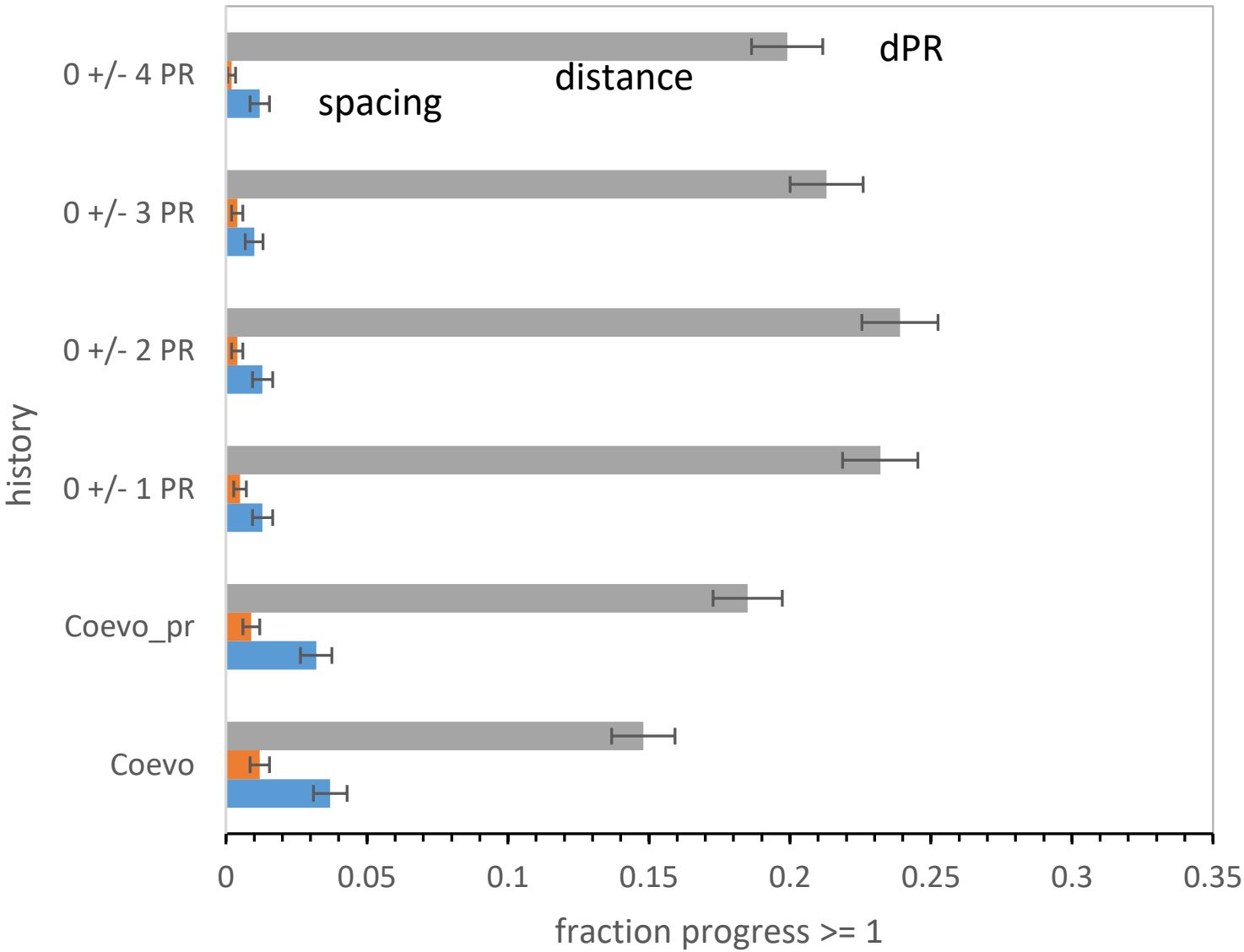
UUU	Phe (4.5)	*UCU*	Ser (7.5)	*UAU*	Tyr (7.7)	*UGU*	Cys (4.3)
UUC	Phe (4.5)	*UCC*	Ser (7.5)	*UAC*	Tyr (7.7)	*UGC*	Cys (4.3)
UUA	Leu (4.4)	*UCA*	Ser (7.5)	*UAA*	Ter	*UGA*	Trp (4.9)
UUG	Leu (4.4)	*UCG*	Ser (7.5)	*UAG*	Ter	*UGG*	Trp (4.9)
CUU	Leu (4.4)	*CCU*	Pro (6.1)	*CAU*	His (7.9)	*CGU*	Arg (8.6)
CUC	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)

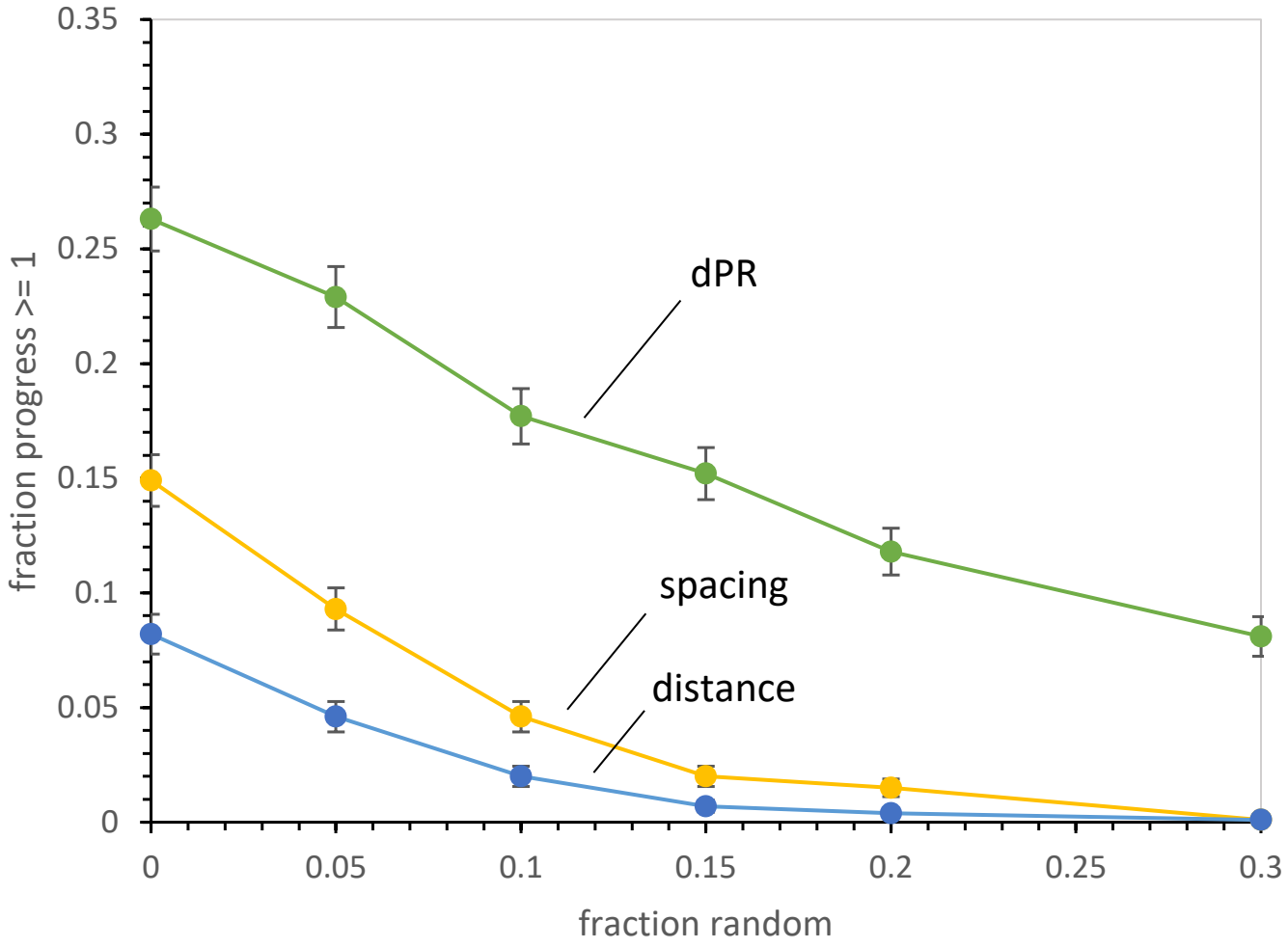
UUU	Phe (4.5)	*UCU*	--	*UAU*	Tyr (7.7)	*UGU*	Cys (4.3)
UUC	Phe (4.5)	*UCC*	--	*UAC*	Tyr (7.7)	*UGC*	Cys (4.3)
UUA	Leu (4.4)	*UCA*	Ser (7.5)	*UAA*	Ter	*UGA*	Ter
UUG	Leu (4.4)	*UCG*	Ser (7.5)	*UAG*	Ter	*UGG*	Trp (4.9)
CUU	Leu (4.4)	*CCU*	Pro (6.1)	*CAU*	His (7.9)	*CGU*	Arg (8.6)
CUC	Leu (4.4)	*CCC*	Pro (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)

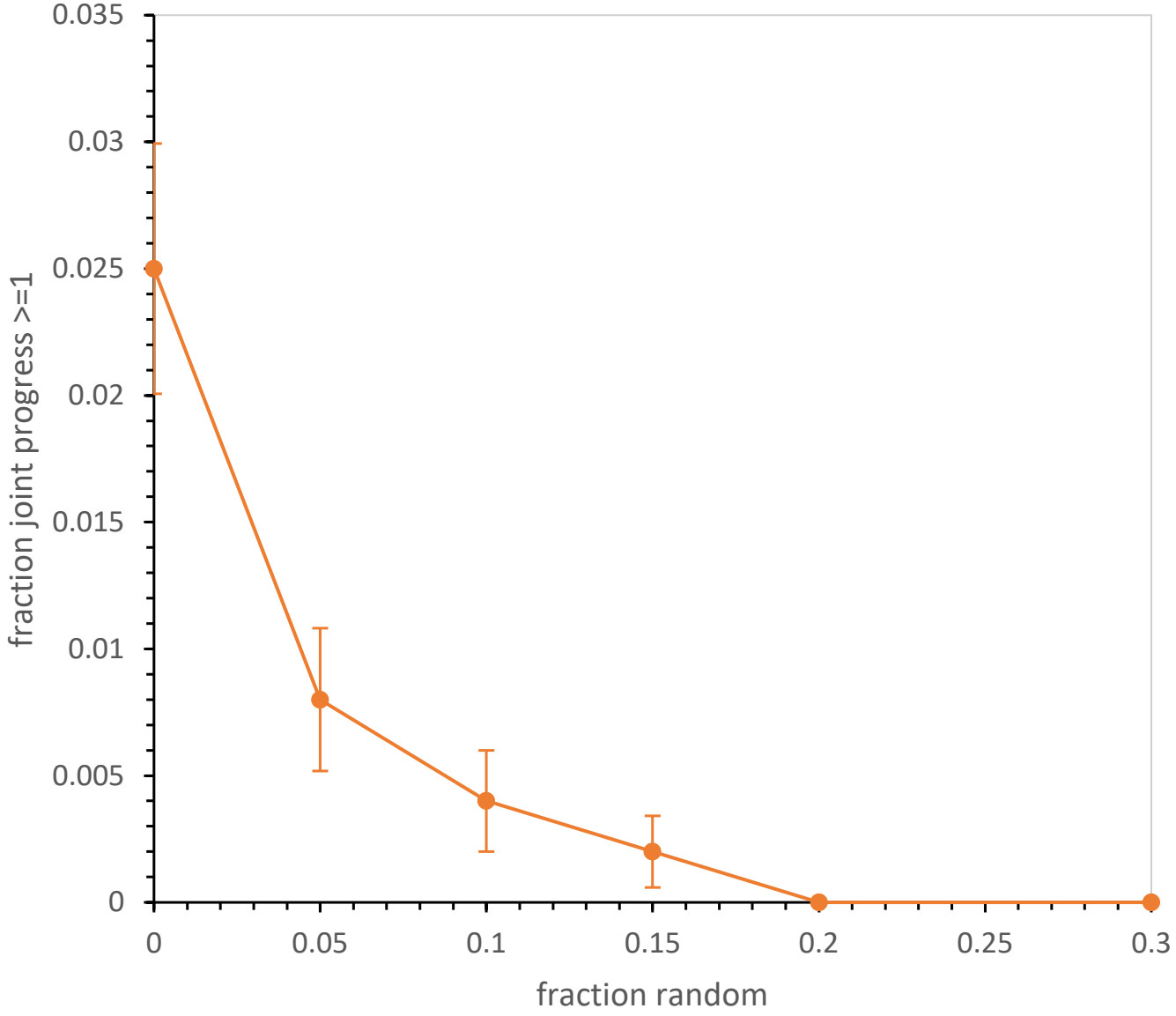
UUU	Phe (4.5)	*UCU*	Ser (7.5)	*UAU*	Tyr (7.7)	*UGU*	Cys (4.3)
UUC	Phe (4.5)	*UCC*	Ala (6.5)	*UAC*	Tyr (7.7)	*UGC*	Tyr (7.7)
UUA	Leu (4.4)	*UCA*	--	*UAA*	Ter	*UGA*	Ter
UUG	Leu (4.4)	*UCG*	--	*UAG*	Lys (10.2)	*UGG*	Ter
CUU	Leu (4.4)	*CCU*	Pro (6.1)	*CAU*	His (7.9)	*CGU*	--
CUC	Leu (4.4)	*CCC*	Pro (6.1)	*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)

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)


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
fraction
continuous
late
0
0.2
0.4
0.6
0.8
1
joint progress >=


dPR
distance
spacing
0 +/- 4 PR
0 +/- 3 PR
0 +/- 2 PR
0 +/- 1 PR
Coevo_pr
Coevo
history
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
fraction progress >= 1


0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
fraction progress >= 1
dPR
spacing
distance
0
0.05
0.1
0.15
0.2
0.25
0.3
fraction random


0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
fraction joint progress >=1
0
0.05
0.1
0.15
0.2
0.25
0.3
fraction random

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)

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)

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)

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)