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

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

Abstract

The Standard Genetic Code (SGC) can arise by fusion of partial codes evolved in different individuals, perhaps for differing prior tasks. Such code fragments can be unified into an SGC after late evolution of accurate third-position Crick wobble. Late wobble advent fills in the coding table, leaving only later development of final translational initiation and termination coding in separate domains of life. This code fusion mechanism is computationally implemented, and c3 fusion before late Crick wobble (c3-lCw) is tested for its ability to evolve the SGC in tens of thousands of trials. Compared with the previously-studied evolution of isolated coding tables, or with a population of increasing numbers of similar non-fusing codes, code fusions reach the SGC sooner, work in a smaller population, present more accurate and more complete codes more frequently, and expose SGC-like codes to selection for longer times than non-fusion mechanisms. Strikingly, c3-lCw also suppresses varied disordered assignments, helping to create a unified coding table. Thus, calculations confirm that c3-lCw code fusions provide a credible evolutionary route to the SGC.

Introduction

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

Here, such information is sought by quantitative modeling of SGC emergence, using, arguably, very general assumptions (Yarus 2021b). It is assumed only that codon assignment, capture and decay (adding here: new codes and their fusion) can be thought of as having characteristic rates. SGC existence is attributable to the joint effects of those rates on the 64 triplet space of a coding table.

In order to embody events whose complexity may be great, but unknown at the start, a computable model is used (Yarus 2021b; Fig. 7). This envisions the development of the SGC as a set of short intervals, called passages, during which one event only (a codon assignment, for example, or even no event) occurs. Encoding events have different probabilities during a passage. The virtue of this formulation is that it allows explicit programming of hypotheses about code evolution, even for histories of great complexity. Implied coding tables are readily computed. Mechanisms can be compared; the best-supported being that which most probably yields a known result, the SGC. Moreover, using different probabilities during an unchanging interval is equivalent to assignment of different rate constants (Yarus 2021b), and timing of events may therefore be compared. When (for example) the real-world time to make a codon assignment is known, such kinetics convert to times on an early Earth (Yarus 2021d).

These inquiries have taken their most recent form (Yarus 2021c) as the idea that the code was composed by fusing separately-developed partial codes, perhaps within fusing primitive cells that had developed differing coding competencies. Code fusion is a commonplace in Biology, having been observed many times, for example, between mitochondrial and nuclear codes (Duchêne et al. 2009).

Creation of the SGC by fusion of separate partial codes was thought (Yarus 2021c) to have specific advantages; for example, realizing the SGC within a smaller code population. Below, hypothetical advantages of code fusion during SGC construction are tested.

The model

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

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

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

Fusion must be a major evolutionary event. Codes must often arise via fusion in order that fusion affect code evolution. This requirement is profound, altering so many things that we discuss fusion rates first.

Code fusion requires two successive events. Firstly, new coding tables arise to create a population of codes alongside the first code. Here, this happens at a fixed probability per passage (Ptab); a passage is the time required for completion of one evolutionary step for a coding table. Thus, time is measured in passages, which simultaneously host evolution in all existing codes (Methods). Either a new assignment, capture of a sequence-related codon, assignment decay, or a code fusion may occur within a passage, accompanied by creation of a new coding compartment that begins with one assigned codon, and evolves alongside, and by the same rules as, the initial code.

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

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

A majority of fusions. Fig. 1 plots the fraction of codes with >= 20 assigned functions that have benefitted from a fusion contribution, versus the probability of new tables (Ptab in Fig. 1A, Pfus is favorable), or versus the probability of fusion (Pfus in Fig. 1B, Ptab is favorable). Both more tables and more likely fusion, as expected, increase the fraction of SGC-like (>=20 function) codes that acquired fused assignments. Because this ms concerns changed properties in a code shaped by fusions, Ptab = 0.08 and Pfus = 0.002 (Fig. 1, rightward) are used below. This requirement for elevated fusion parallels other studies (Aggarwal et al. 2016), and provides a simple rationale for this: fusion must be a significant route to final codes.

Altered population history. Fig. 2 contains averages for every coding table in 1000 code populations carried to 750 passages. Beginning with a single initial code, this ranges from about 5400 potential codes at 60 passages to about 61000 potential codes at 750 passages. Thus, accurate values for mean code kinetics are available. In Fig. 2, evolving coding tables are divided into 4 classes - they have had no fusions, have engaged in fusion, have been annihilated by an incompatible fusion, or have received a successful fusion (see Methods).

The striking fact is that after these times, sufficient to allow unfused tables to become infrequent (only about 6% are unfused at 750 passages), the predominant fate of coding tables is loss in fusion, and this has been true since early times, just before 140 passages. Most codes are lost because they fused into others, and a significant minority were annihilated by trying to fuse to codes with incompatible assignments. Putting the same data in other words, only 9.4% of once-existent coding compartments still exist at 750 passages (those with no fusion or successful fusions). Successful fusions themselves have a peak around 110 passages, after which they are also lost in later destructive fusions. Fig. 7, which sketches the history of one simplified environment, may help conceptualize these losses. We will return to the early successful fusions (Fig. 2), and to their later decline, below.

Superior codes follow fusion. We now examine codes existing later in Fig. 2; this minority of fusion survivors includes codes that closely approach the SGC. Fig. 3 shows the properties of the most complete codes from 10000 code populations evolved throughout the interesting era of Fig. 2, from 150 to 750 passages, when fusions begin, then decisively shape, an environment's codes.

Competence. Fig. 3A depicts abundance of SGC-like codes. That is, these codes survive: they have either experienced no fusion, or a complementary fusion that adds assignments. They have assigned codons to >= 20 of the standard functions, and so are almost complete. Moreover, plotted compartments encode their functions very similarly to the SGC, having assignments completely overlapping (mis0, blue line), a single differing assignment (mis1, red), two differences (mis2, gray) or 3 differently-assigned triplets (mis3, yellow).

The crescendo. The fraction of competent codes tends to rise from an origin just after the appearance of population-wide fusions (Fig. 2) to the end of calculations (Fig. 3A; the crescendo). In addition, these triply-unusual codes, with completeness, fusion contributions and accuracy that are all exceptional, are quite frequent, seemingly well within the reach of a search for SGC-like translation. For example, the completely SGC-like class (mis0) are detected in 10000 environments early, at 150 passages, and are 1/250 among the best codes at 750 passages. Even supposing a demanding coding selection, requiring precise SGC mimicry, such codes might not only be found relatively early, but require finding a superior later translational system in ≈ 1/250 environments. This seems easily possible.

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

The crescendo is not static. Existing codes (Fig. 3) quickly acquire additional assignments. A bit less quickly, new codes arise and a code may fuse. Less frequently yet, new codons are captured for existing assignments and assignments decay. A large flux of change is absorbed by evolving codes. The implication is that the competent code population is constantly changing on a timescale comparable to its initial appearance. The coding crescendo is constantly evolving, with new codes replacing the previously competent: for example, those lost to unproductive fusions (Fig. 2). Thus the crescendo offers a changing face to selection, as well as increasingly frequent competent codes. Selection can wait for a particularly effective SGC-like code to appear from the jumble.

The crescendo and its competence come from fusion I. The crescendo of competence is produced by code fusion.

Fig. 3B shows code evolution similar to Fig. 3A, but without fusion (Pfus = 0). As in other Fig. 3 panels, fractions of the most complete codes from 10000 environments (Fig. 12) are plotted versus passages (time). Having multiple codes itself facilitates the evolution of more complex coding. Thus environments with parallel coding tables as in Fig. 3A, but with no fusions between them (Fig. 3B), present a useful comparison to multiple tables and fusion together (Fig. 3A).

Fig. 3B is plotted with the same y-axis to facilitate comparison with Fig. 3A: clearly, no crescendo exists. In fact, for multiple tables without fusion, all levels of completeness with assignment accuracy arise later, achieve lower frequencies among most complete codes, and do not persist. Competence ultimately declines instead of increasing as in Fig. 3A.

The crescendo and its competence come from fusion II. Fig. 3C completes controls for Fig. 3A; it describes a similar set of code evolutions, but lacking both multiple tables and fusion (Ptab = 0, Pfus = 0). This resembles the system previously analyzed (Yarus 2021b), where code evolution takes place in a single initial coding table in each environment, each evolving until it resembles the SGC. However, Fig. 3C is useful here because its individual codes are precisely the same as those of Fig. 3A and 3B. Its single codes without fusion are seen against the same passage time scale, making comparisons obvious.

Fig. 3C also has the same frequency scale and colors as Fig. 3A and 3B. Thus it is clear that restriction to one code makes SGC resemblance later than in Fig. 3A, restricts such competence to lower levels even than for multiple tables in Fig. 3B, and again shows no crescendo as seen for fusing systems (Fig. 3A). In fact, SGC-like coding is everywhere lower than for multiple tables without fusion (Fig. 3B). For example, complete resemblance to the SGC (mis0) is not detected among 10000 environments until 450 passages and then at too low a frequency (≈ 10^{-4}) to be deciphered on Fig. 3C's ordinate.

Origin of competence. It is clear why environments that fuse code compartments, cells or partial coding tables are superior. Fig. 4A compares mean misassignments for the most complete codes (having assigned >= 20 functions) from 10000 environments. Multiple codes with fusion (red, Fig. 4A) are similar to multiple codes without fusion (green, Fig. 4A) and to codes without multiple tables and fusion (blue, Fig. 4A) until fusion becomes predominant in code evolution (see **Altered population history**, above). After this time (≈ 150 passages, Fig. 4A) errors in different evolutionary modes diverge greatly. Strikingly, other modes almost double mean misassignment in codes with fusion.

There are only two sources of misassignment in present code environments. The more straightforward is that random assignment is allowed by a variable probability of random association between functions and triplets (Prand). Because these assignments are genuinely randomized, they are unlikely to be the same as for the SGC.

Fate of random assignments. Fig. 4B shows the mean number of randomized assignments in the most complete codes from 10000 coding environments (Prand = 0.05) versus time. The pattern strikingly reproduces overall accuracy in Fig. 4A. That is, codes derived by fusing multiple tables (blue, Fig. 4B) have made 2-3 fold fewer random assignments than without fusion (red, Fig. 4B) or without both multiple tables and fusion (gray, Fig. 4B).

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

Fig. 4C shows that codes derived using fusion more strongly discriminate against captures of mutationally related triplets for related functions. Again, the pattern follows that in Fig. 4A: before prevalent fusion, the three modes of code evolution are similar. Afterwards, they progressively diverge under fusion; late fused codes (blue, Fig. 4C) utilize 4-5 fold fewer error-prone captures at 750 passages than do unfused (red, Fig. 4C) or single codes (gray, Fig. 4C).

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

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

Data in Fig. 5 are for tens of thousands of environments at 300 passages, a time when all modes are evolving SGC-like codes (Fig. 3). The frequencies of unfused, near-complete, accurately assigned codes are plotted logarithmically versus the linear probability of random assignment (Prand). Log frequencies of such SGC-like codes tend to decrease linearly with Prand (see also (Yarus 2021c)), with decrease somewhat more rapid as the rigor of requirement increases. Inerrant codes (mis0) decrease somewhat more than those with three misassignments (mis3). But even if Prand = 0, coding is not perfect, because of other sources of error remain, like capture of mutationally-related triplets for the same or similar assignments, which is error-prone (Fig. 4). Frequencies for good coding shown are higher than we have previously observed by roughly 10-fold. Nonetheless, the prior rule-of-thumb (Yarus 2021b) need not be changed for fused code evolution: 1, 2 or 3 encoded functions may have been assigned for different reasons, or for no specific reason. Origin hypotheses still have substantial access to the SGC even if they randomly assign triplet functions in roughly 1 of 10 cases.

Discussion

The Standard Genetic Code. Because all modern Terran organisms possess the SGC or a close relative, the most economical hypothesis is that the SGC already existed in the last common ancestor. The SGC's origin has therefore attracted an immense literature that defies summary, especially within the space of a single manuscript. However, below I attempt a brief synopsis of the beliefs of a plurality of code workers (almost no beliefs exist that can be confidently attributed to a majority), in order to put present findings in context.

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

Late Crick wobble (lCw). It does not seem likely that accurate third position wobble (Crick 1966) was a primordial form of genetic coding. tRNA-rRNA interaction on the ribosome is extensive, spanning the tRNA molecule and sites in both large rRNAs (Moazed and Noller 1986, 1989). Some of these contacts appose rRNA nucleotides with codon-anticodon triplets to check their conformations (Ogle et al. 2001). Such checks determine whether the first two base pairs are Watson-Crick (Demeshkina et al. 2012), as well as whether wobble positions lie within the multiple conformations allowed for normal, tautomeric and charged wobble pairs (Westhof et al. 2019). Such sophisticated 3-dimensional error checking is unlikely for primordial encoding, but would evolve later in coding history. Thus, here coding is assumed to involve normal base pairing until a late time when wobble becomes possible, probably in a ribonucleopeptide proto-ribosome. Given that Crick wobble readily fills the coding table (Yarus 2021b), simplified Crick wobble is here assumed to be adopted quickly throughout a nascent SGC once it becomes possible. Others have also treated wobble as a significant late coding development (Lei and Burton 2021).

Late Crick wobble is shown in explanatory diagrams (Fig. 6, 7), but actually plays no part in present discussion of code capabilities because completeness (all functions encoded) is the chosen criterion for code progress, and completeness is unaffected by Crick wobble, which only extends already-existent assignments.

In fact, reliance on all molecular mechanics is minimized here because, in part, ancient coding machineries are still uncertain. Thus c3-iCw avoids dependence on particular structures - fusion advantages would likely be similar for any means of encoding in which fused parts of the coding table developed separately.

Summary of c3-lCw. Fusion joins partial codes which may have diverse encodings. Fusion must occur as frequently if fusion is to alter final codes (Fig. 1), because even a single coding table without such fusion may evolve to resemble the SGC (Yarus 2021b; Fig. 3C). Fusion profoundly alters coding environments (Fig. 2). Code fusions will likely be undirected, because fused codes cannot express a phenotype until after fusion itself. Thus, fusions that facilitate expansion toward the SGC will be accompanied by those that do not – if fused codes are incompatible, two participants with incompatible codes can even be annihilated (Fig. 2, 7).

A fusing population shrinks as fusion expands. Under these conditions, the long-term fate of codes is entry into effective, innocuous or disastrous combinations – only 9.4% of ≈ 61000 codes that once existed survive in a late environment (750 passages, Fig. 2).

Such losses change the average survivor (Fig. 3, 4). A fusing population shows a crescendo of individuals (Fig. 3A) with near-complete codes (>= 20 functions) that also resemble majority encoding (0, 1, 2 or 3 differences from the SGC). This continuing rise is a specific product of fusion. It does not appear for similar coexisting multiple codes lacking fusion (Fig. 3B). In addition, SGC-like codes are everywhere fewer for a similar system in which similar fusion partners do not appear (Fig. 3 C). Moreover, competence is delayed in both non-fusing systems (Fig. 3). Code fusions are more readily selected: they present better codes sooner, more frequently, and in a smaller population, for a longer time.

The reasons for crescendo competence are clear: average misassignment is suppressed in fused populations, compared to non-fusing ones (Fig. 4A). That is: variant codes have been selectively extinguished in unproductive fusion attempts (Fig. 2, 6). In more detail, a fused population has made fewer erroneous random assignments (Fig. 4B), as well as fewer error-prone captures of mutational-related triplets (Fig. 4C). This requires only elementary hypotheses about selection - that it will favor encoding in which codons have unique meanings. This would occur with many different fusion mechanisms; thus fusion purification of a dominant encoding is only slightly dependent on evolutionary details.

Environments evolving via code fusions produce many competent codes, more frequently (Yarus 2021b) than do prior models. For example, ≈ 1% to ≈ 11% of the environments in Fig. 3A have <=3 misassignments with >= 20 functions - these frequencies that seem within easy reach of selection for superior SGC translation. Or more specifically, exact (mis0 class) SGC codes exist in > 1 in 1000 environments (Fig. 5), though only selection intrinsic to fusion has yet been applied. For this latter class, the only selection required is harvesting exact SGC's among other codes. This exemplifies distribution fitness (Yarus 2021b), where a beneficial distributed property is selected among an excellent minority. Nevertheless, self-refinement due to fusion still requires accurate assignments (Yarus 2021c), still plausibly estimated as <= 10% random (Fig. 5).

C3-lCw is consistent with previous conjecture. Fusion was initially suggested in order to allow the SGC to incorporate different forms of ordered assignments (Yarus 2021c); for example, related amino acids in the same column or row. It was also suggested that code fusion could both speed appearance of the SGC and allow it to appear in a smaller population. These predictions are borne out in Fig. 2 and Fig. 3

The surprising crescendo. Simple assumptions (fusions between incompatible partial codes are deleterious) yield striking convergence on an existing code, potentially selecting the SGC (Fig. 3, 4). Convergence suppresses error from varied sources (Fig. 4A, 4B, 4C), and thus is likely to be broadly applicable to other variant codes. But most significantly, complete and accurate codes accumulate continuously during a long era of prevalent fusion (Fig. 3A). The same events remove codes from the group of accurate codes as well as create them (Fig. 2), so particular accurate codes present at one time (Fig. 3A) will differ from those earlier and later. Therefore, a population exists, comprised of different

SGC-like codes over a long period, such codes increasing in the environment, until selection of a superior one among them. It is difficult to imagine a more effective protocol for evolving SGC-like encoding.

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

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

Second definition: RNA agents vs ribonucleoprotein agents. The first definition of coding eras is approximately echoed in the distinction between an early era resting solely on the capabilities of RNA (Gilbert 1986) and a second later era of ribonucleopeptide agents. This second transition must lie somewhere near the freezing point because aminoacyl-RNA synthetases themselves are very complex proteins (they have multiple specific substrate binding sites and perform multiple catalyses), so a modern variety of them is only plausible after a time when the code is strongly constrained by its extensive prior encodings.

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

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

Current evidence suggests that the most ancient organisms took the form of close laminations, with different organisms densely layered within 3.43 Gya stromatolites (Allwood et al. 2006), or close bundles of filaments 3.75 to 4.28 Gya in seafloor jasper (Papineau et al. 2022). Either formation permits close approach of cells of different origins and competency, and potentially encourages fusion of their codes.

Given these microscopic fossils, c3-lCw might have occurred ≈ 4 billion years ago, as proposed in Fig. 6.

Late assignments to late amino acids. SGC coding triplets can be sequences extracted from ancient RNA binding sites for cognate amino acids (Yarus and Christian 1989; Rodin et al. 2011; reviewed in Yarus 2017). A counterargument sometimes offered (e.g., Koonin and Novozhilov 2017) is that metabolically complex, likely late-appearing amino acids like arginine (Janas et al. 2010) and tryptophan (Majerfeld et al. 2010) exist among amino acids associated with triplet-containing sites. However, simpler, early amino acids like isoleucine also have prominent coding triplets (Lozupone et al. 2003). Also: consider two coding eras (Fig. 6): assignment of triplets from RNA binding sites would probably not cease in a later ribonucleopeptide era. Instead, excellent RNA sites for late amino acids like arginine (Yarus and Christian 1989) would be used when advantageous. For this reason, more competent ribonucleopeptide enzymes could expand anabolism to complex amino acids, while their SGC assignments still utilized excellent fits for RNA binding sites.

HGT and universality. (Vetsigian et al. 2006) expresses Carl Woese's conviction that early HGT was a crucial "innovation-sharing protocol". Before the genome and inheritance, only creatures possessing the same or similar genetic codes could share innovations evolved independently. Such communal innovation-sharing was a sufficient advantage to decrease variations in coding, before accurate inheritance had evolved. An independent learning model confirms that mutual genetic intelligibility via HGT, without vertical inheritance, could have universalized the SGT (Froese et al. 2018). Present work suggests that in a yet-earlier era, when only partial codes existed, HGT would have speeded the assembly of, purified the population of, and facilitated the selection of, SGC-like codes.

Methods

The program that repeatedly evolves coding environments and reports code properties was written using the integrated development environment in Lazarus v. 2.20RC1 in its console mode, with the free Pascal FPC 3.2.2 compiler. Compiled mechanisms were run on a Dell XPS computer under 64-bit Microsoft Windows 10 @ 2.9 GHz on an Intel Core i9-8950HK CPU, using 32 GB of RAM.

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

Time is measured in cycles through code evolution, called passages. Passages, and other details of program action, are depicted in Fig. 7. Assignment, decay and codon capture, which occur mutually exclusively during passage through a single code, have been discussed previously (Yarus 2021b).

Acknowledgement Thanks to Norm Pace for an informative discussion of stromatolites.

References

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

Allwood AC, Walter MR, Kamber BS, Marshall CP, Burch IW. 2006. Stromatolite reef from the Early Archaean era of Australia. *Nature* **441**: 714–718.

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

Błażej P, Wnętrzak M, Mackiewicz D, Mackiewicz P. 2018. Optimization of the standard genetic code according to three codon positions using an evolutionary algorithm. *PloS One* **13**: e0201715.

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.

Carter CW. 2015. What RNA World? Why a Peptide/RNA Partnership Merits Renewed Experimental Attention. *Life Basel Switz* **5**: 294–320.

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.

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

Di Giulio M. 2008. An extension of the coevolution theory of the origin of the genetic code. *Biol Direct* **3**: 37.

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

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

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

Gilbert W. 1986. The RNA world. *Nature* **319**: 618.

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

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

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

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

Jukes T, Osawa S. 1993. Evolutionary changes in the genetic code. *Comp Biochem Physiol B* **106**: 489–494.

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

Koonin EV, Novozhilov AS. 2017. Origin and Evolution of the Universal Genetic Code. *Annu Rev Genet* **51**: 45–62.

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

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

Majerfeld I, Chocholousova J, Malaiya V, Widmann J, McDonald D, Reeder J, Iyer M, Illangasekare M, Yarus M, Knight R. 2010. Nucleotides that are essential but not conserved; a sufficient L-tryptophan site in RNA. *RNA* **16:** 1915-24.

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

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

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

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

Moazed D, Noller HF. 1989. Interaction of tRNA with 23S rRNA in the ribosomal A, P, and E sites. *Cell* **57**: 585–97.

Moazed D, Noller HF. 1986. Transfer RNA shields specific nucleotides in 16S ribosomal RNA from attack by chemical probes. *Cell* **47**: 985–994.

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

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

Ogle JM, Brodersen DE, Clemons WM, Tarry MJ, Carter AP, Ramakrishnan V. 2001. Recognition of Cognate Transfer RNA by the 30S Ribosomal Subunit. *Science* **292**: 897–902.

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

Ribas de Pouplana L. 2020. The evolution of aminoacyl-tRNA synthetases: From dawn to LUCA. *The Enzymes* **48**: 11–37.

Rodin AS, Szathmáry E, Rodin SN. 2011. On origin of genetic code and tRNA before translation. *Biol Direct* **6**: 14.

Sengupta S, Higgs PG. 2015. Pathways of Genetic Code Evolution in Ancient and Modern Organisms. *J Mol Evol* **80**: 229–243.

Taylor FJ, Coates D. 1989. The code within the codons. *Biosystems* **22**: 177–187.

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

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

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

Wong JT-F. 1975. A Co-Evolution Theory of the Genetic Code. *Proc Natl Acad Sci USA* **72**: 1909–1912.

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

Yarus M. 2021a. Crick Wobble and Superwobble in Standard Genetic Code Evolution. *J Mol Evol* **89**: 50–61.

Yarus M. 2021b. Evolution of the Standard Genetic Code. *J Mol Evol* **89**: 19–44.

Yarus M. 2021c. Fitting the standard genetic code into its triplet table. *Proc Natl Acad Sci USA* **118**: e2021103118.

Yarus M. 2021d. Optimal Evolution of the Standard Genetic Code. *J Mol Evol* **89**: 45–49.

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

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

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

Figure legends

Figure 1A – **Complete fused codes vs probability of fusion.** Evolution was stopped when the first code with >= 20 functions appeared in an environment. The fraction, among 500 environments, of such >= 20 function codes that acquired assignments from fusion is plotted versus the probability of fusion during a passage, Pfus. Pmut = 0.00975, Pdecay = 0.00975, Pinit = 0.150, Prand = 0.050, Ptab = 0.08.

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

Figure 2 – **Code fate vs time.** Mean fractions of total codes are plotted versus passages (time). Kinetics for several fates are shown: donors – codes that have successfully fused / annihilations – codes lost via incompatible fusions / no fusion – unfused codes / total donors – successful fusions plus unsuccessful fusions. Pmut = 0.00975, Pdecay = 0.00975, Pinit = 0.150, Prand = 0.050, Pfus = 0.00200, Ptab = 0.08. At 60 passages, there were 5412 total codes; at 750 passages, there were 60790 codes in these 1000 environments.

Figure 3A – **SGC-like codes vs time: tables and fusion.** The fraction of complete codes (>= 20 functions) with cited levels of misassignment (relative to the SGC) is plotted for 10000 environments that have both new code initiation and fusion. mis0 = no misassignments / mis1 = 1 misassignment, and so on. 10000 environments evolved to the times/passages shown, and codes with >= 20 assigned functions were characterized. Pmut = 0.00975, Pdecay = 0.00975, Pinit = 0.150, Prand = 0.050, Pfus = 0.002, Ptab = 0.08.

Figure 3B – **SGC-like codes vs time: tables, no fusion.** The fraction of complete codes with cited levels of misassignment (relative to the SGC) is plotted for 10000 environments that have new code initiation, but no fusion. mis0 = no misassignments / mis1 = 1 misassignment, and so on. 10000 environments evolved to the times/passages shown, and codes with >= 20 assigned functions were characterized. Pmut = 0.00975, Pdecay = 0.00975, Pinit = 0.150, Prand = 0.050, Pfus = 0.000, Ptab = 0.08.

Figure 3C – **SGC-like codes vs time: no new tables, no fusion.** The fraction of complete codes with cited levels of misassignment (relative to the SGC) is plotted for 10000 environments that have no new code initiation, and no fusion. mis0 = no misassignments / mis1 = 1 misassignment, and so on. 10000 environments with single tables evolved to the times/passages shown, and codes with >= 20 assigned functions were characterized. Pmut = 0.00975, Pdecay = 0.00975, Pinit = 0.150, Prand = 0.050, Pfus = 0.000, Ptab = 0.00.

Figure 4A – **Mean misassignments vs time: tables and fusion, tables no fusion, no tables no fusion.** 10000 environments were evolved to the times shown, and misassignments relative to the SGC were counted among most complete codes in each environment. Probabilities are the same as in Fig. 3 for the three kinds of evolution.

Figure 4B – **Randomly assigned codons vs time: tables and fusion, tables no fusion, no tables no fusion.** 10000 environments were evolved to the times shown, and randomly assigned codon triplets

were counted among most complete codes in each environment. Probabilities are the same as in Fig. 3 for the three kinds of evolution.

Figure 4C – **Capture of mutationally related codons vs time: tables and fusion, tables no fusion, no tables no fusion.** 10000 environments were evolved to the times shown, and capture of triplets one mutation distant from assigned codons were counted among most completely assigned codes in each environment. Probabilities are the same as in Fig. 3 for the three kinds of evolution.

Figure 5 – **Fraction SGC-like codes vs probability of random assignments.** 10000 (or 20000 for greatest Prand) environments were evolved to 300 passages. Among substantially complete codes (>= 20 assigned functions), fractions with different levels of misassignment were counted. mis0 = no misassignments relative to the SGC / mis1 = 1 misassignment, and so on. Pmut = 0.00975, Pdecay = 0.00975, Pinit = 0.150, Pfus = 0.00200, Ptab1 = 0.08.

Figure 6 – **A coding crescendo separates two early code eras.** Three definitions for divided early code evolution (see text) are interpreted as outcomes of c3-lCw. The initiation of code fusion (“fusions”), evolution of wobble (“Crick wobble”) and selection of the SGC from the crescendo (“selection”) are marked relative to the crescendo (Fig. 3A). Late wobble evolves just before selection, because wobble appears in the SGC. The last common ancestor lies off to the right of Fig. 6. The c3-lCw’s approximate time before the present (bottom, upward arrow) reflects that of the most ancient fossil biota (see text).

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

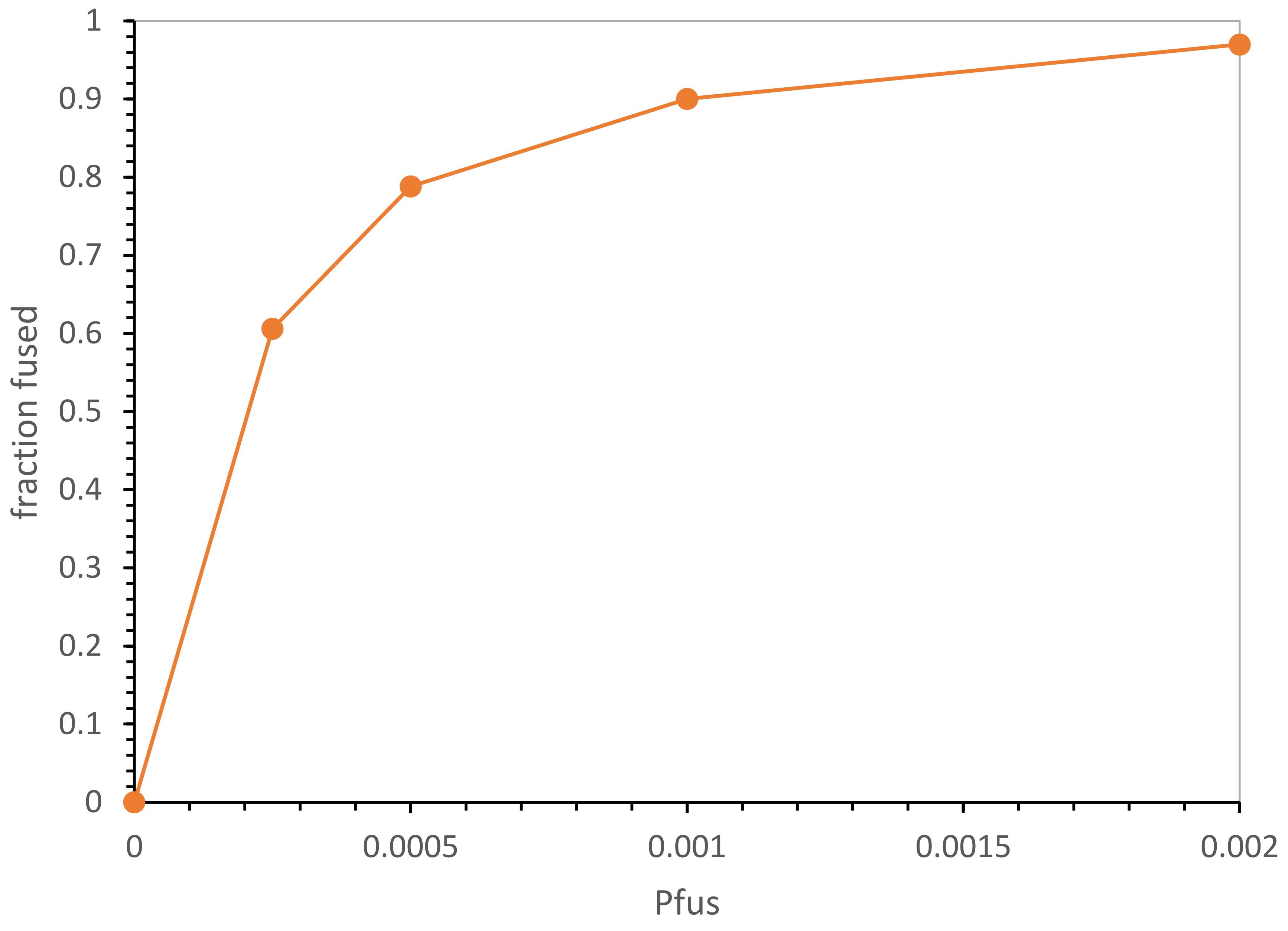

1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
fraction fused
0
0.0005
0.001
0.0015
0.002
Pfus

Figure 1A

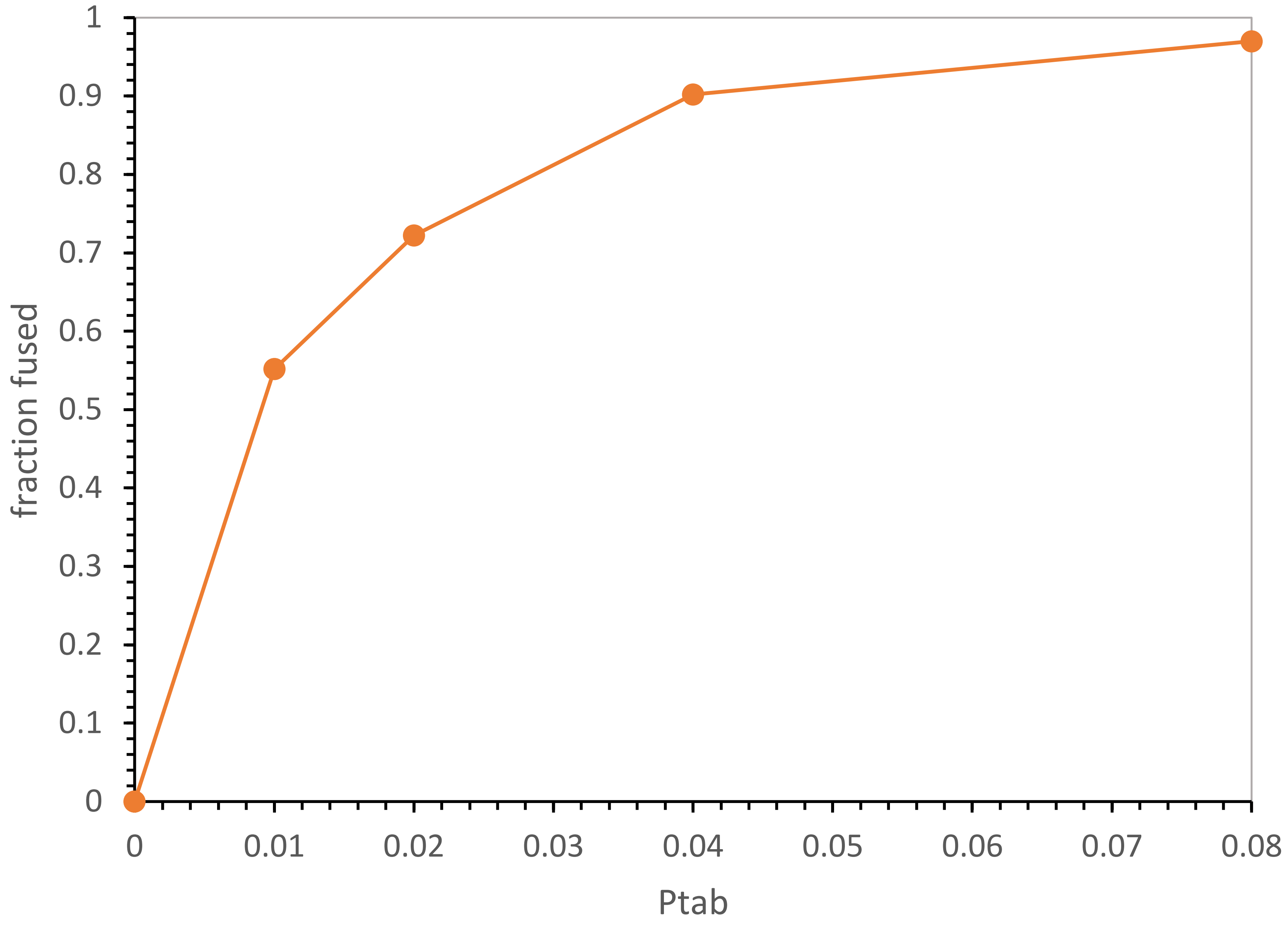
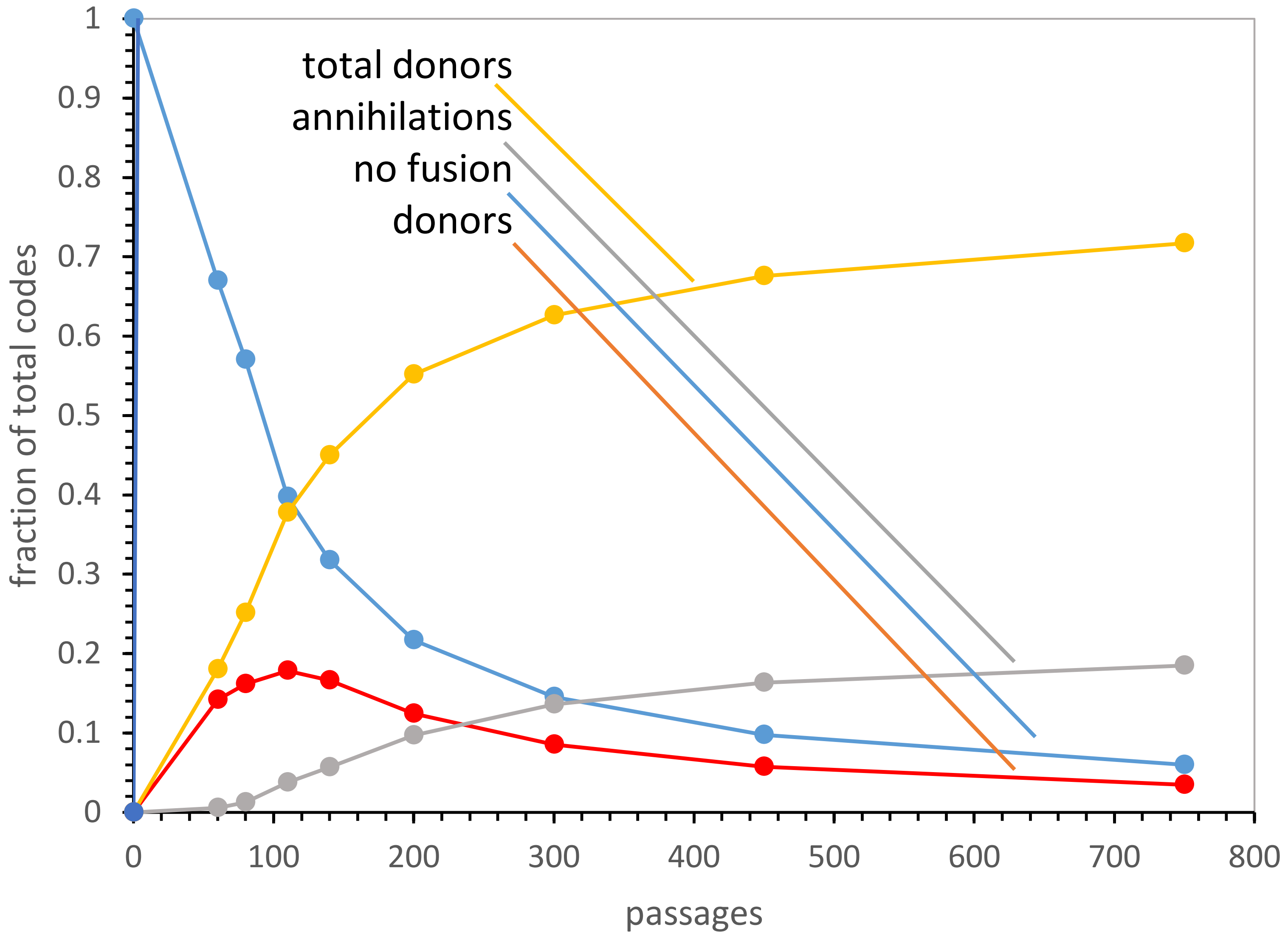
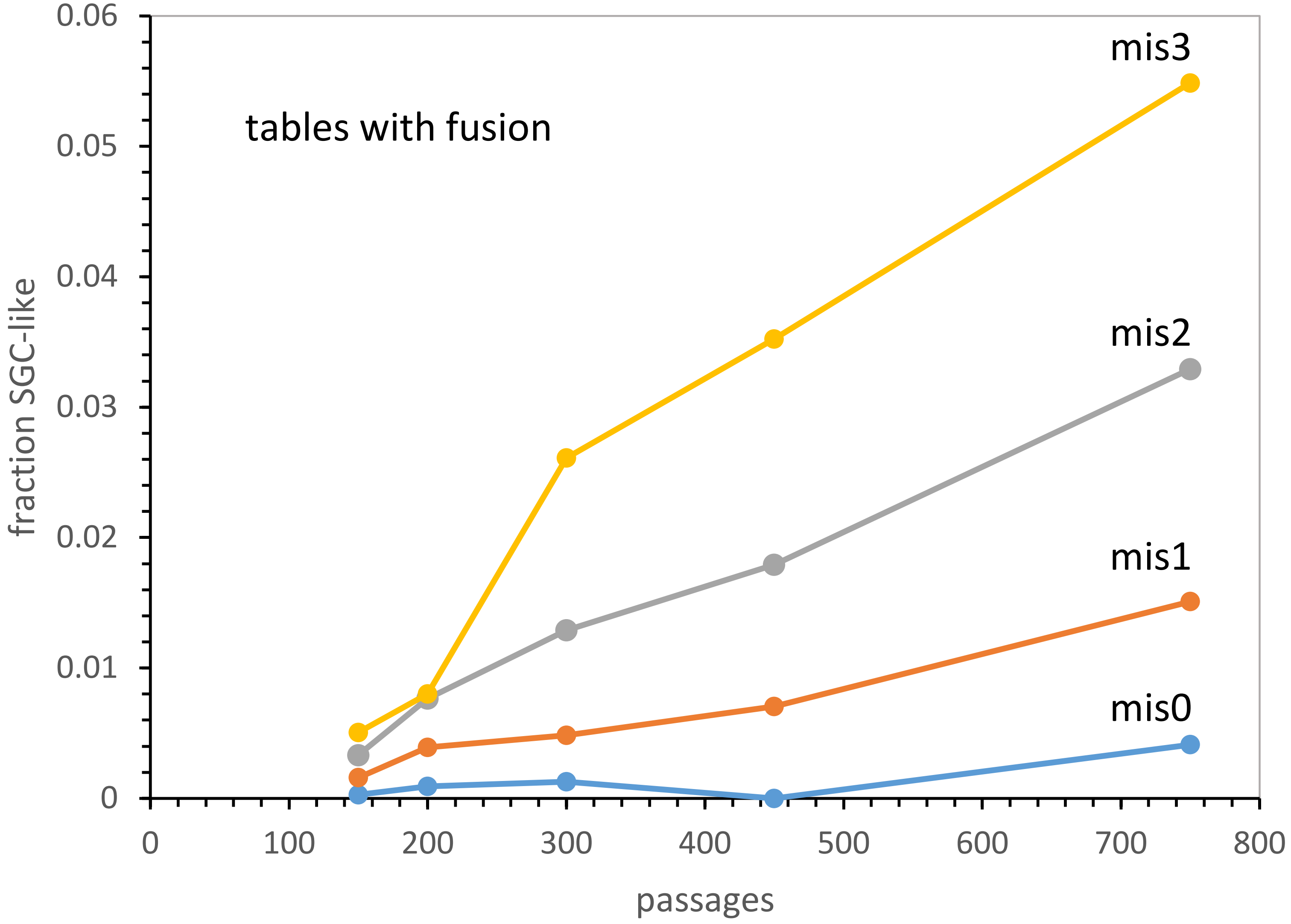

1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
fraction fused
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
Ptab

Figure 1B



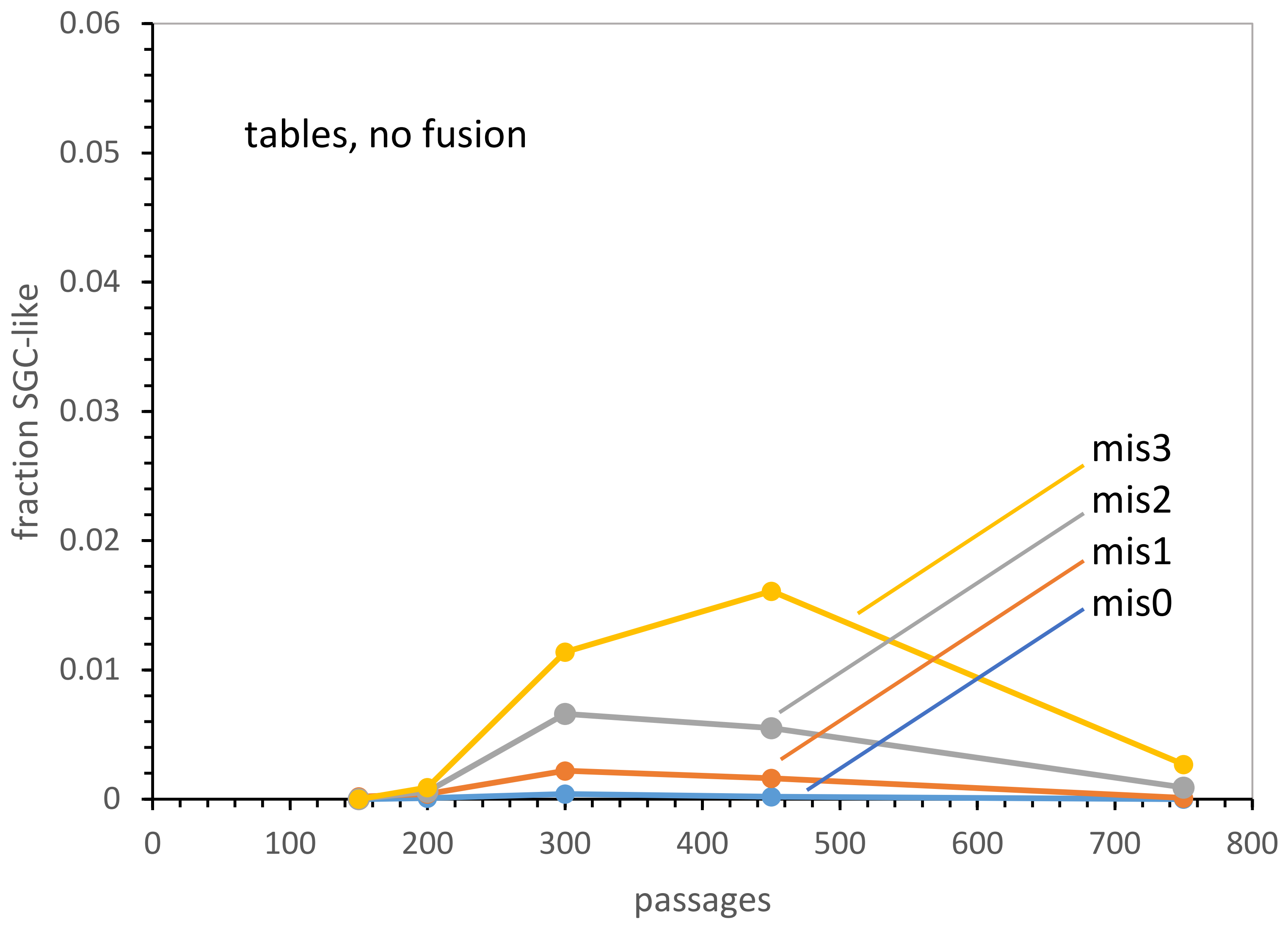
total donors
annihilations
no fusion
donors
fraction of total codes
passages
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
100
200
300
400
500
600
700
800

Figure 2



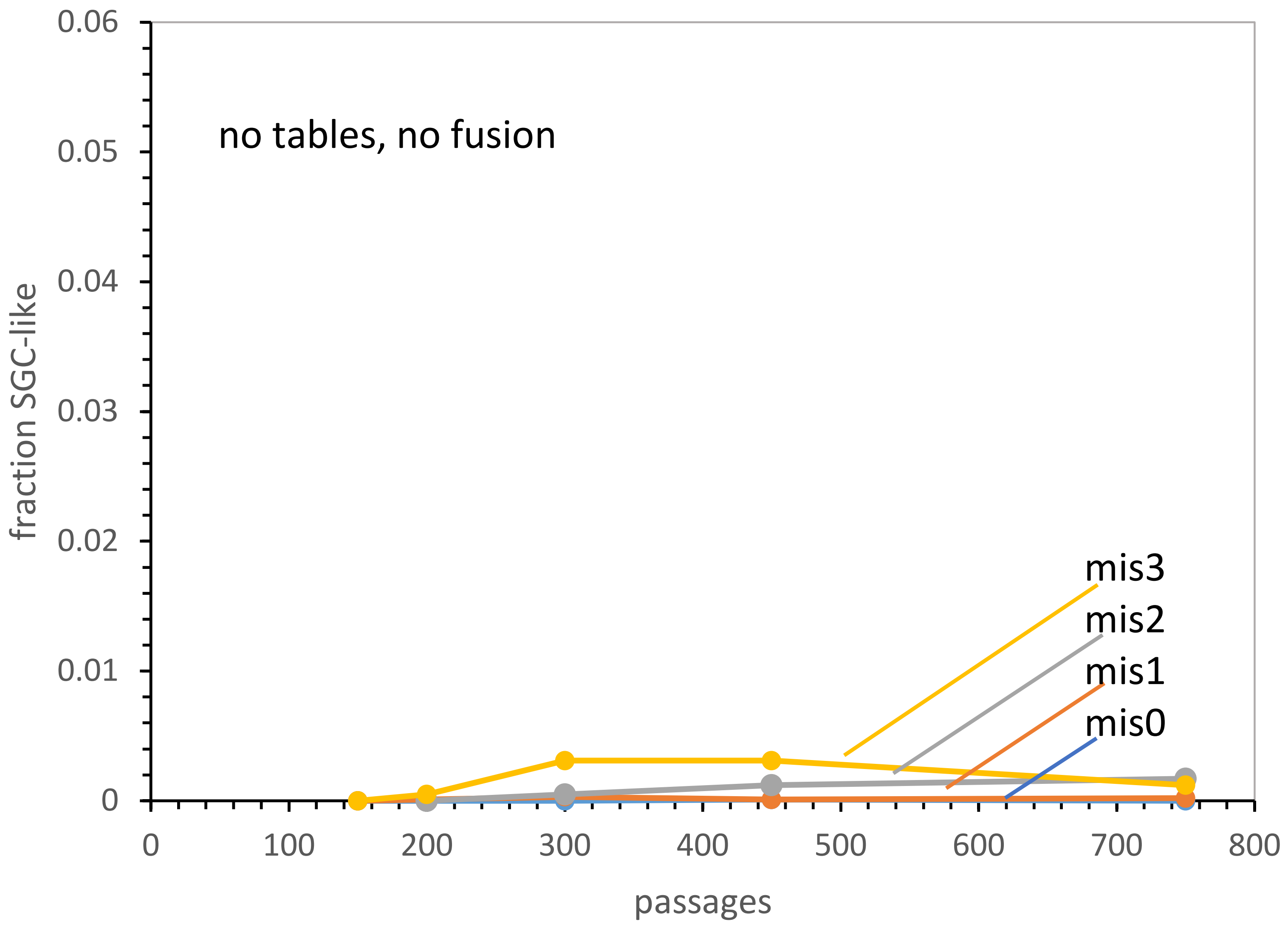
tables with fusion
mis3
mis2
mis1
mis0
fraction SGC-like
passages
0.06
0.05
0.04
0.03
0.02
0.01
0
0
100
200
300
400
500
600
700
800

Figure 3A



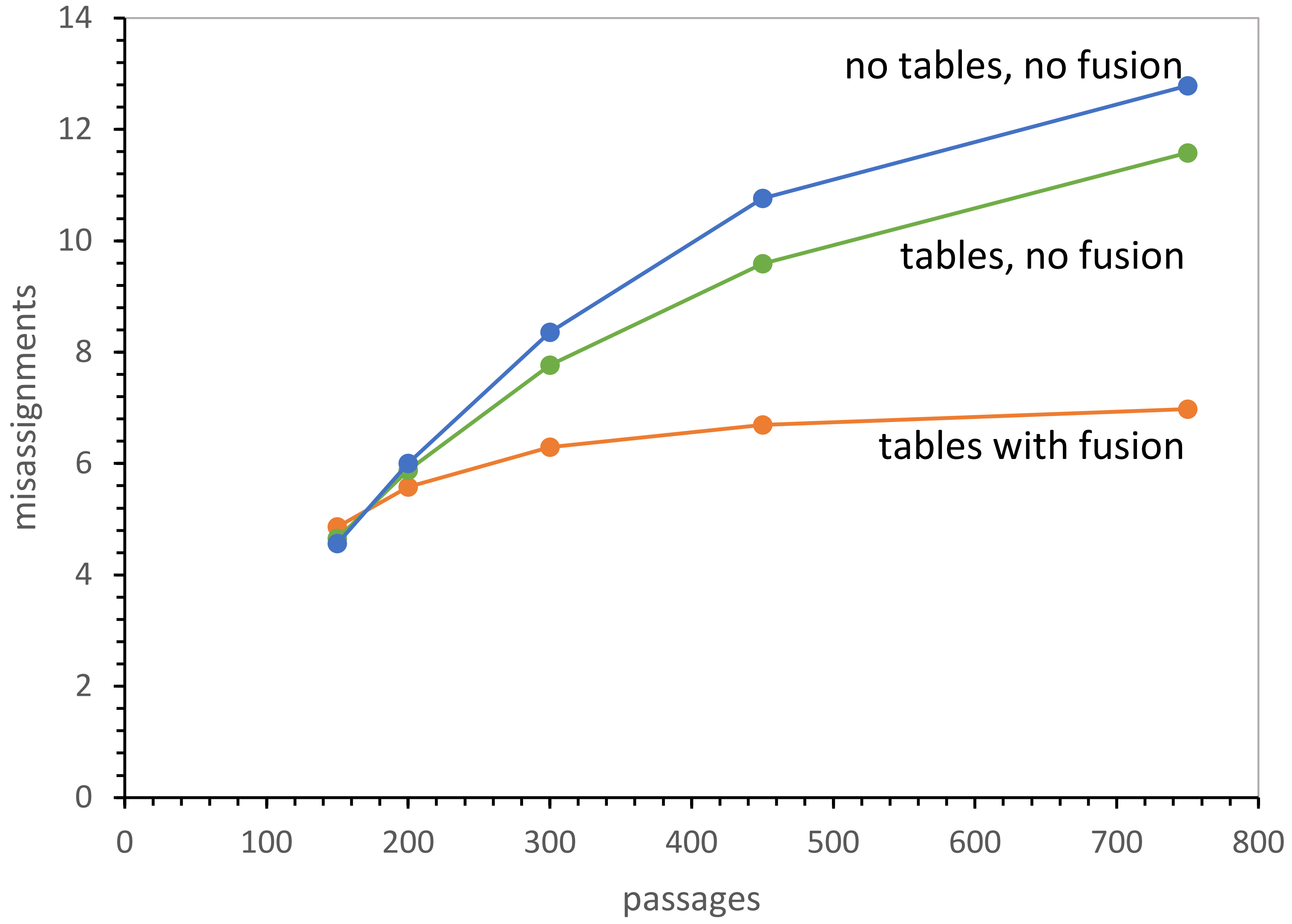
tables, no fusion
fraction SGC-like
passages
0.06
0.05
0.04
0.03
0.02
0.01
0
0
100
200
300
400
500
600
700
800
mis3
mis2
mis1
mis0

Figure 3B



0.06
0.05
0.04
0.03
0.02
0.01
0
fraction SGC-like
no tables, no fusion
mis3
mis2
mis1
mis0
0
100
200
300
400
500
600
700
800
passages

Figure 3C



14
12
10
8
6
4
2
0
misassignments
0
100
200
300
400
500
600
700
800
passages
no tables, no fusion
tables, no fusion
tables with fusion

Figure 4A

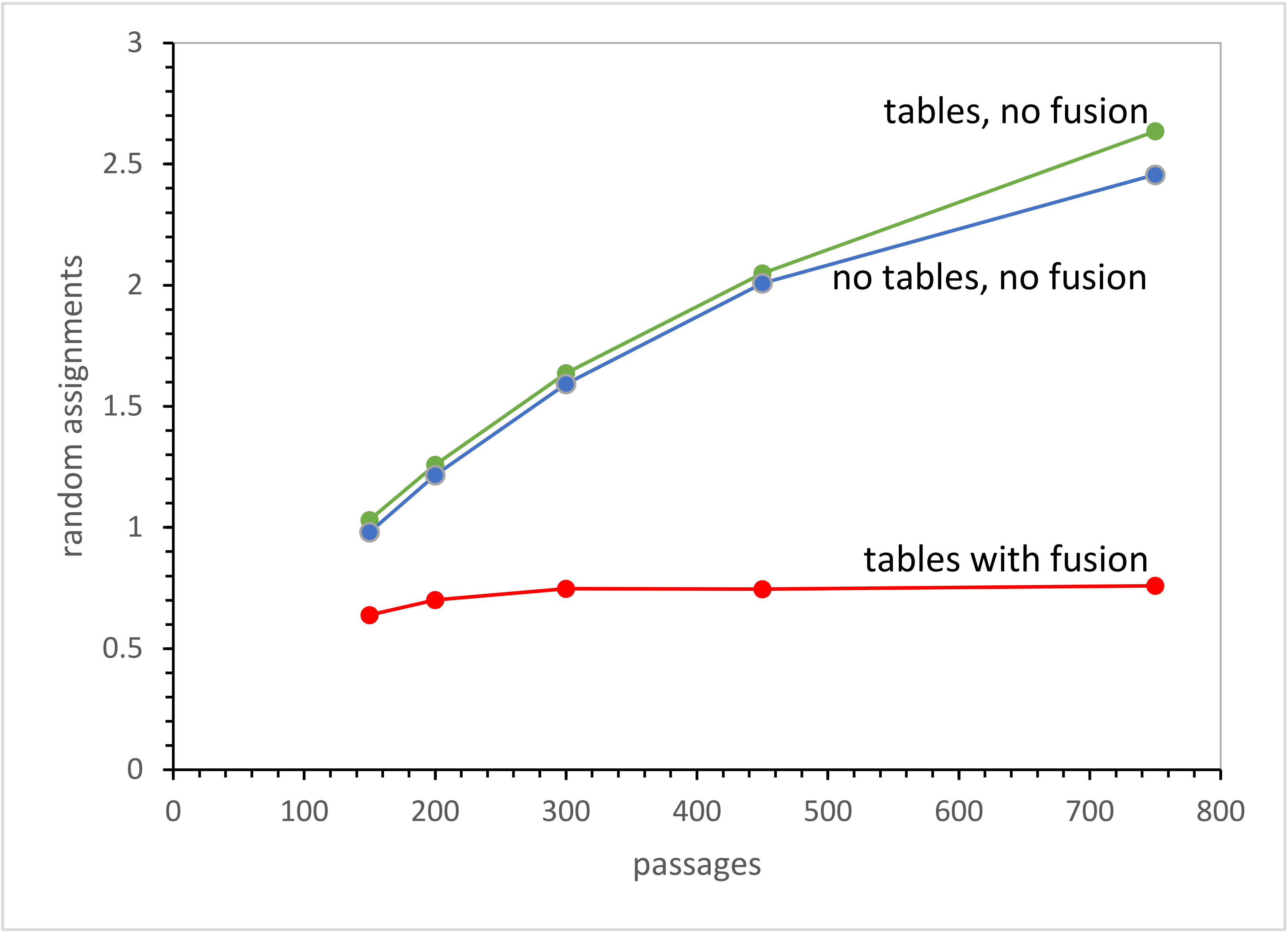
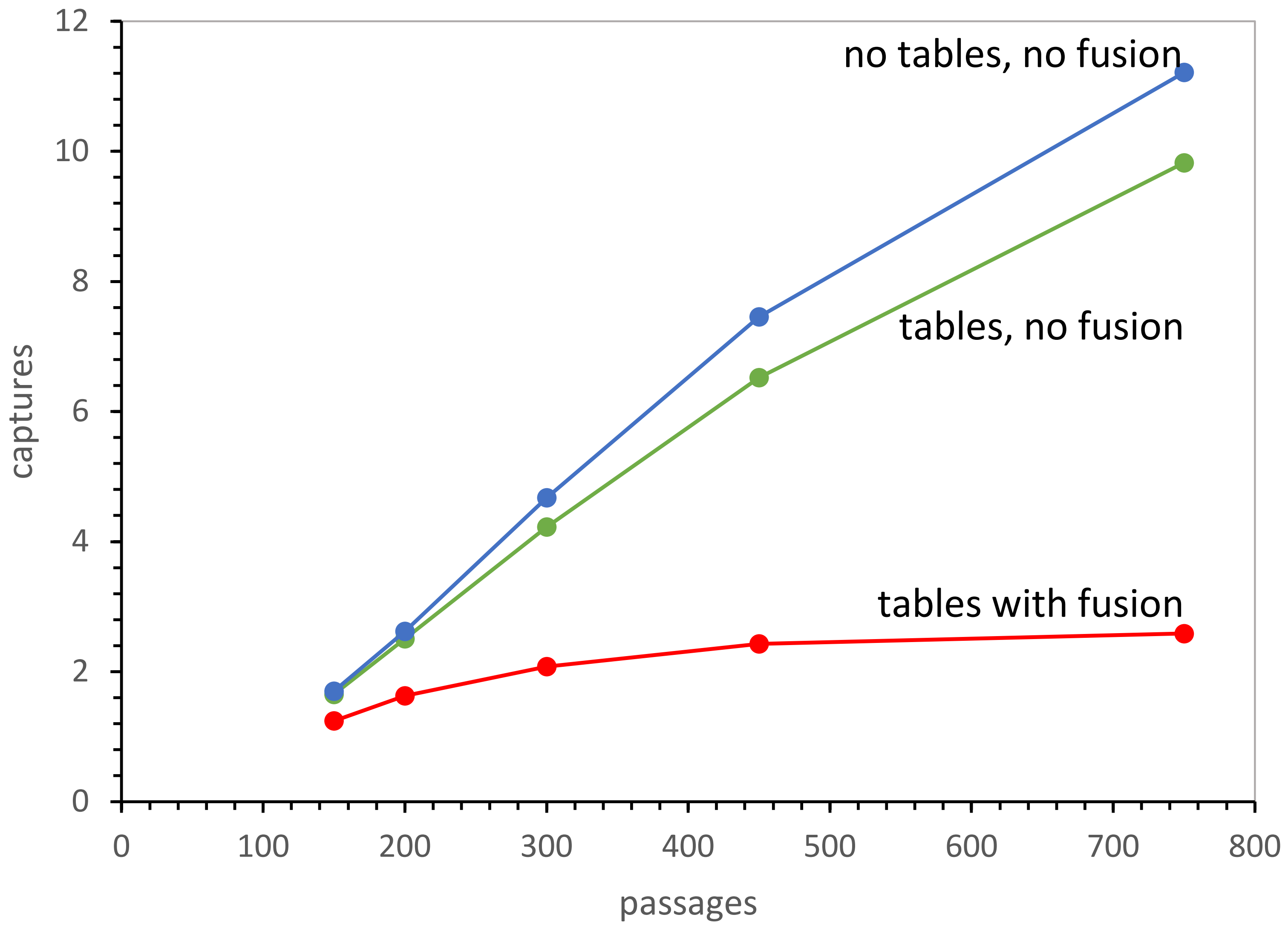
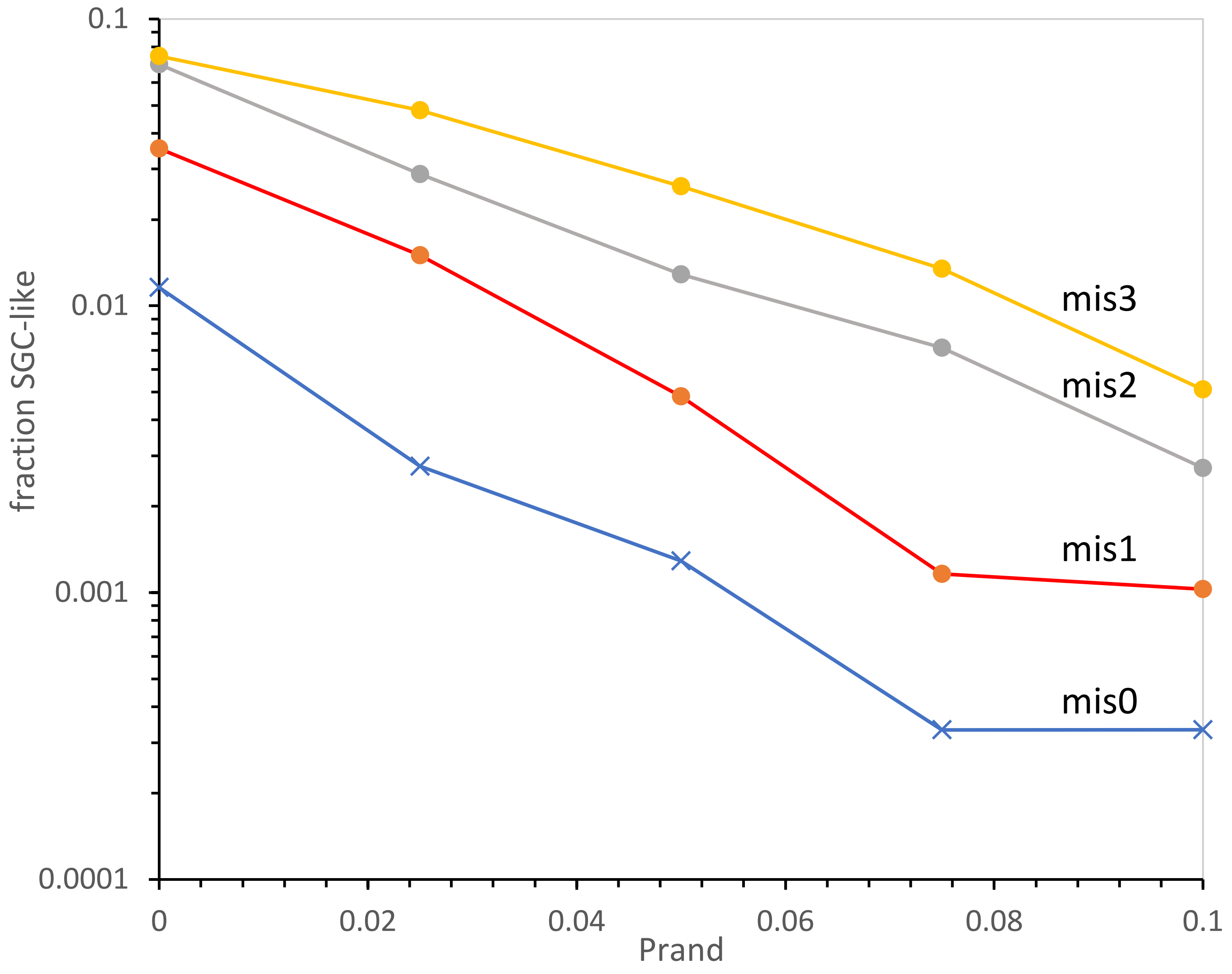

3
2.5
2
1.5
1
0.5
0
random assignments
0
100
200
300
400
500
600
700
800
passages
tables, no fusion
no tables, no fusion
tables with fusion

Figure 4B



no tables, no fusion
tables, no fusion
tables with fusion
captures
passages
0
2
4
6
8
10
12
0
100
200
300
400
500
600
700
800

Figure 4C



0.1
0.01
0.001
0.0001
fraction SGC-like
0
0.02
0.04
0.06
0.08
0.1
Prand
mis3
mis2
mis1
mis0

Figure 5

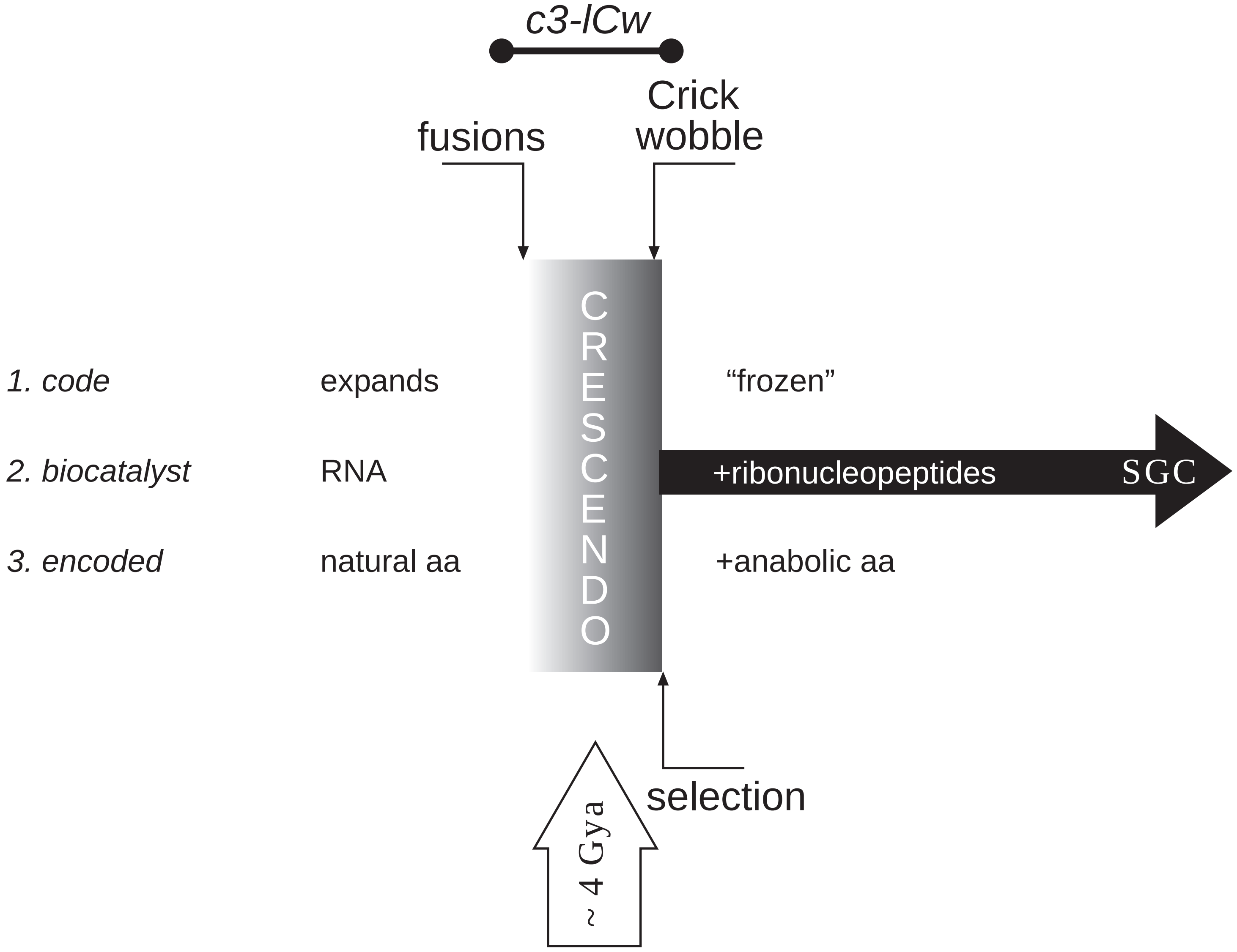

c3-lCw
fusions
Crick wobble
CRESCENDO
1. code
expands
"frozen"
2. biocatalyst
RNA
+ribonucleopeptides
SGC
3. encoded
natural aa
+anabolic aa
selection
~ 4 Gya

Figure 6

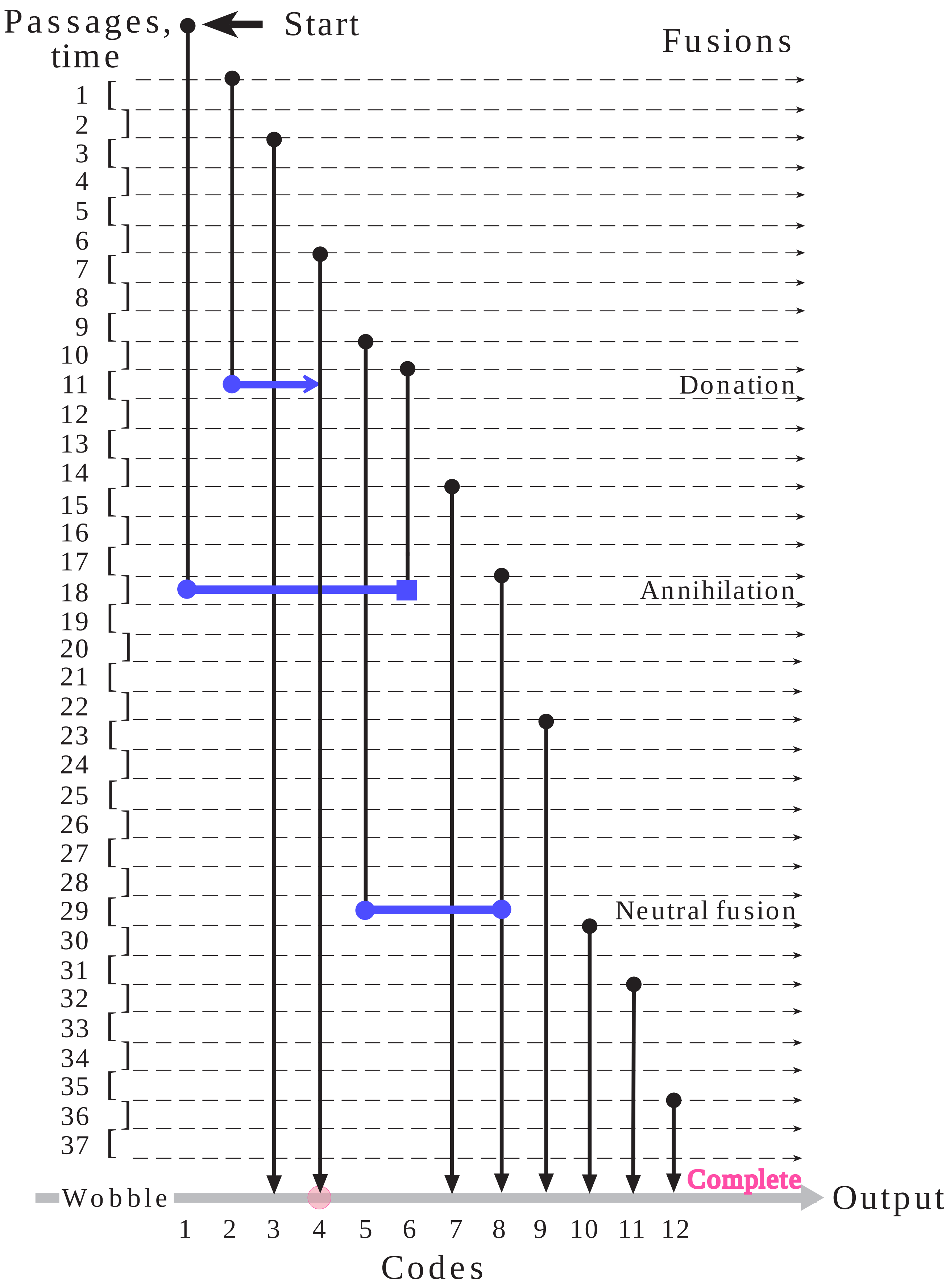

Passages, time
Start
Fusions
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
Donation
Annihilation
Neutral fusion
Complete
Wobble
Output
1 2 3 4 5 6 7 8 9 10 11 12
Codes

Figure 7