J Mol Evol (2006) 63:297–313
DOI: 10.1007/s00239-004-0176-7

JOURNAL OF MOLECULAR EVOLUTION

The Coevolution of Genes and Genetic Codes: Crick's Frozen Accident Revisited

Guy Sella,[1] David H. Ardell[2]

[1] Center for the Study of Rationality, The Hebrew University, Givat Ram, 91904 Jerusalem, Israel
[2] Linnaeus Centre for Bioinformatics, Uppsala University Biomedical Center, Husargaten 3, Box 598, 751 24 Uppsala, Sweden

Received: 8 June 2004 / Accepted: 21 October 2005 [*Reviewing Editor:* Dr. Martin Kreitman]

Abstract. The standard genetic code is the nearly universal system for the translation of genes into proteins. The code exhibits two salient structural characteristics: it possesses a distinct organization that makes it extremely robust to errors in replication and translation, and it is highly redundant. The origin of these properties has intrigued researchers since the code was first discovered. One suggestion, which is the subject of this review, is that the code's organization is the outcome of the coevolution of genes and genetic codes. In 1968, Francis Crick explored the possible implications of coevolution at different stages of code evolution. Although he argues that coevolution was likely to influence the evolution of the code, he concludes that it falls short of explaining the organization of the code we see today. The recent application of mathematical modeling to study the effects of errors on the course of coevolution, suggests a different conclusion. It shows that coevolution readily generates genetic codes that are highly redundant and similar in their error-correcting organization to the standard code. We review this recent work and suggest that further affirmation of the role of coevolution can be attained by investigating the extent to which the outcome of coevolution is robust to other influences that were present during the evolution of the code.

Key words: Genetic code — Coevolution

Correspondence to: Guy Sella; *email:* gsella@math.huji.ac.il

Introduction

A genetic code, which associates codons and amino acids, has two basic characteristics that reflect on its function. One is the code's amino acid *vocabulary*. The vocabulary of a genetic code affects its function because it dictates the family of proteins that genes can encode. Therefore, the addition of useful amino acids to the vocabulary of a code allows genes to encode for a greater repertoire of proteins. The second characteristic is the code's *robustness to error*, which is an outcome of the code's organization. The replication and translation of genes are inherently prone to error. A code is robust to error if it is organized such that codons that are frequently interchanged by these errors are associated with functionally similar amino acids. A robust code thus reduces the deleterious effects of errors in the replication and translation of proteins.

How is the ubiquitous standard genetic code organized in these respects? The redundancy of the standard code suggests that it could have been modified to incorporate more amino acids (Ardell and Sella 2001). There are 61 codons that are translated into 20 amino acids, i.e., more than a threefold redundancy. Even though some of this redundancy may be attributed to biochemical constraints that are inherent to the operation of the translation apparatus (the wobble coding), a conservative estimate of the redundancy that is not the outcome of such constraints (Osawa et al. 1992) leaves us with a redundancy that is at least twofold that needs to be explained. The variety of posttranslational covalent modifications to amino acids that occur in modern

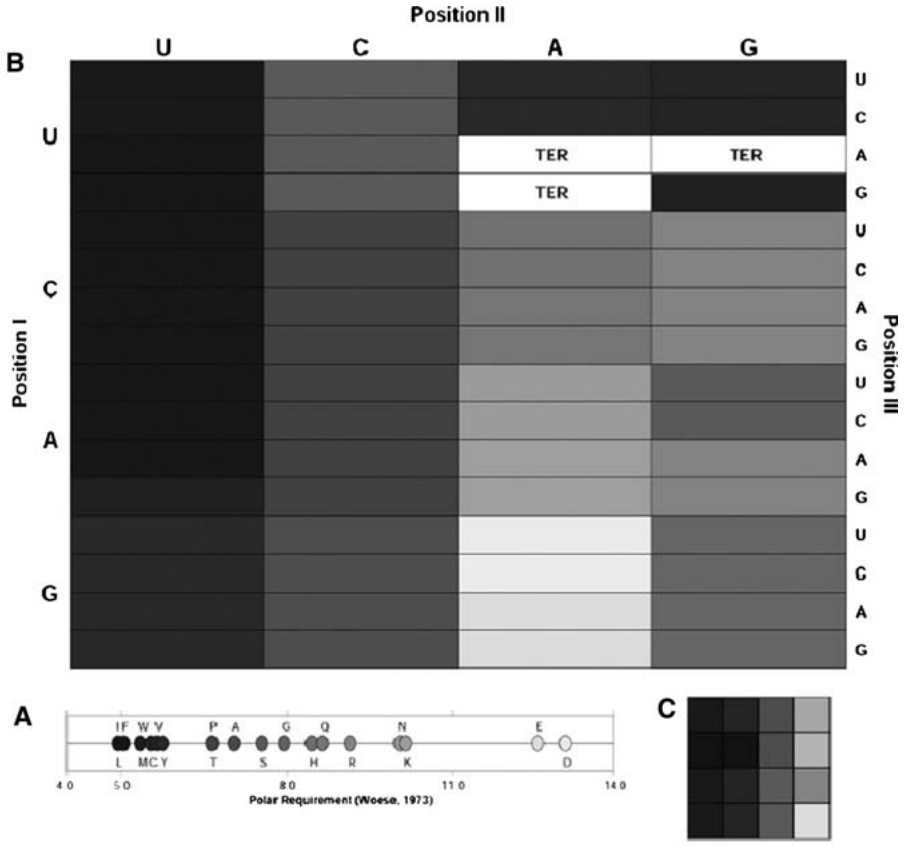

Fig. 1. A representation of the standard code according to the physicochemical properties of the amino acid it encodes. **A** The 20 amino acids, denoted according to their standard abbreviations, are each ascribed a shade that corresponds to their physicochemical properties (we use Woese's [1967] polar requirement). Thus, when two amino acids are ascribed a similar shade here, it means that they can, on average, replace each other more easily in proteins, and with less deleterious consequences. **B** The entry corresponding to codon UCG is the rectangle corresponding to first position U, second position C, and third position G. The polar requirement of the amino acid encoded by codon UCG, which is serine (S), is represented by the shade in the entry corresponding to serine. The regularities in the organization of the SGC that are apparent in this representation are reviewed in the text. **C** A genetic code generated by our simulations of gene-code coevolution. In this simulation the codons consist of two bases, and mutation and misreading were incorporated according to their qualitative characteristics in reality.

proteins (see, e.g., Krishna and Wold 1993) and the existence of noncanonical cotranslationally inserted amino acids—selenocysteine and pyrrolysine (see e.g., Francklyn et al. 2002)—suggests that the addition of amino acids to the 20 canonical ones would be useful to the production of proteins. On the other hand, the intricate organization of the standard code makes it extremely robust to error (Woese 1965; Woese et al. 1966; Alff-Steinberger 1969; Swanson 1984; Haig and Hurst 1991; Ardell 1998; Freeland and Hurst 1998; Freeland et al. 2003). This intricate organization is illustrated in Fig. 1, where the following is apparent. (i) Amino acids are more similar to each other along the first codon position than they are along the second. This "column-like" pattern is associated with translational misreading, which is much higher in the first codon position than it is in the second (Davies et al. 1964, 1966; El'skaya and Soldatkin 1985; Parker 1989). (ii) Along the second codon position, amino acids associated with *pyrimidine* bases Y = {U, C} or *purine* bases R = {A, G} are more similar within these sets than between them. This regularity is associated with mutations in replication, in which transitions (mutations within these base sets) occur more frequently than transversions (mutations of a base in one set to a base in the other set) (Freese 1961; Sankoff et al. 1973; Topal and Fresco 1976; Gojobori et al. 1982; Hixon and Brown 1986; Petrov and Hartl 1999; Vigilant et al. 1991; Wakeley 1996). Note that these regularities are not associated with the code's redundancy but, rather, with the way different amino acids are organized in the code. The two can be separated by comparing the standard code with randomized permuted codes that share the same family boxes but differ in the assignment of amino acids to these boxes (Haig and Hurst 1991; Ardell 1998; Freeland and Hurst 1998; Freeland et al. 2003). It has been established that the integration of these regularities places the standard code in the top millionth of randomized codes at minimizing the deleterious effects of errors in replication and translation (Freeland and Hurst 1998).

To explain how the standard genetic code was endowed with these properties we must consider its evolution. However, none of the theories on the evolution of the code (see Knight et al. [1999] and Freeland et al. [2003] for a review) have been shown to account for the salient organization of the standard code. Here we review recent work (Ardell and Sella 2001, 2002; Sella and Ardell 2002) demonstrating how both the redundancy and the robustness to error observed in the standard code can be the outcome of the coevolution of genes and genetic codes in the presence of errors in replication and translation. (Unless specified otherwise, we use the term coevolution to refer to the coevolution of genes and the genetic code, rather than to the coevolution of metabolism and the genetic code [Wong 1975, 2005; Taylor and Coates 1989; Di Giulio 2004].)

A simple consideration suggests that genes and genetic codes should have coevolved. At any stage in the evolution of the code, the genes and the evolving code were allied through the selection on proteins. To avoid confusion in the definition of genes in the presence of an evolving code, we refer to an organism's protein-coding regions, be they DNA or its precursor, as its *genetic message*. Given a genetic code, the selection on proteins determines the composition of genetic messages. In turn, given the existing genetic messages, the genetic code is under selection to produce useful proteins with the messages presented to it.

The implications of coevolution at different stages in the evolution of the code were explored by Francis Crick in a seminal paper from 1968. His "frozen accident" theory laid the foundation for future coevolutionary thinking about the evolution of the code. We therefore briefly review the principle assumptions and deductions of this theory (we review only the ideas from this paper that are associated with coevolution). Crick begins by characterizing the primitive genetic code. He reasons that the size of codons was unlikely to change during the evolution of the code because "a change in codon size makes nonsense of all previous messages and would almost certainly be lethal." Therefore the primitive code must have been a triplet code. Crick then suggests that the primitive code was likely to involve only a few amino acids, although it may have encoded ambiguously for classes of amino acids (Woese 1965). At that stage, too many nonsense codons would be strongly selected against, because mutation would introduce them into messages where they would cause severe interruptions in translation (Sonneborn 1965). Crick assumes that the easiest way to produce new tRNAs that translate these nonsense codons would have been to alter the anticodons of existing tRNAs. Therefore, these few amino acids, or amino acid classes, would have quickly spread all over the code such that most codons would quickly be brought into use. Furthermore, the codons associated with each amino acid, or amino acid class, were likely to be related.

After the primitive code was established, the coevolution of codes and messages would have taken a different course. Crick suggests that at that stage the formation of the code proceeded by the introduction of new amino acids and by an increase in the precision of recognition of amino acids within classes. Such changes in the code were likely to be disruptive at some protein sites and advantageous at others. For changes in the code to succeed, they should have, in balance, given the cell a reproductive advantage. This was more likely if the amino acids that replaced each other were similar, because then the deleterious effects were likely to be smaller. Each change in the code was consolidated by corresponding changes in messages. After the meaning of a codon changed, selection on messages would establish the use of that codon at sites where the new amino acid meaning was advantageous, and replace it where that meaning was disruptive. The net effect of the whole series of changes would be that similar amino acids would tend to have similar codons, which is what we observe in the present code. If this process left its mark in modern tRNA sequences one might expect tRNAs associated with similar amino acids to be nearby on a phylogenetic tree. A prediction along these lines (for codons rather than amino acids) was corroborated by Fitch and Upper (1987). However, this corroboration should be taken with a grain of salt, as other studies of tRNA phylogeny have reached different conclusions (Di Giulio 1994; Xue et al. 2003) and, more generally, tRNA's may be too short and functionally constraint to allow for reliable deductions about a process as primordial as the evolution of the standard code (Knight et al. 1999; Freeland et al. 2003).

As the process of code evolution proceeded the number of proteins in the genome became larger and their design became more sophisticated. Crick postulates that when the code reached its current developed form, any change in it would have introduced new amino acids into numerous highly evolved proteins, and would have therefore been catastrophic for the organism. At that stage the code became a "frozen accident". This abrupt freezing may account for the standard code's considerable redundancy.

Despite his conviction that the coevolutionary ideas reviewed above are "crucial to the evolution of the code," Crick is very critical of his own theory. His main criticism is that the theory is "too accommodating," and "in a loose sort of way it can explain anything". Although the theory suggests how the standard code may have become redundant, it would have worked equally well if the standard code had incorporated 15, or 25, amino acids. And although it

suggests why similar amino acids are associated with similar codons, it falls short of explaining the detailed organization of the code we have reviewed above. According to Crick's "frozen accident" theory, coevolution only preserved the associations between codons and amino acid properties that can be traced back to the original division of the primitive code among the first amino acids. Thus, in spite of the theory's appeal, its explanatory value is questionable if Crick was right in suggesting that it places very little constrains on the properties of the frozen code.

The work we review below suggests that this picture changes significantly if we consider the way errors in replication and translation affect the course of code-message coevolution. We show that a coevolutionary process, which begins with a highly ambiguous primitive code (Sonneborn 1965; Fitch 1966; de Duve 1995) and ends with a frozen code (Crick 1968), does substantially more than associate similar codons with related amino acids, it generates a code that is organized along the lines we have reviewed above. To assure that this organization is the result of coevolution, as opposed to being an indirect outcome of a predisposed primitive code, we assume an initial code with no specificity in translation. We further assume that the course of coevolution was affected by two main factors. The first, which is similar in spirit to Crick's (1968) proposal, is an effective selection on the level of amino acid residues in translated proteins. Here we assume this selection acts along one physicochemical dimension that corresponds to amino acid polarity. The second is the errors in replication and translation. Here we assume that during the evolution of the code these errors were qualitatively similar to those we see today. Although Crick (1968) and Woese (1965) considered the effect of errors on the coevolutionary process, they may have underestimated its importance—especially for mutations (Sella and Ardell 2002).

This review is divided into four parts. First, we introduce a mathematical framework that allows us to follow through the "mechanics" of code-message interactions. The framework incorporates selection on amino acid residues and errors in replication and translation, and it describes how a given genetic code determines the composition of messages and how the composition of messages condition changes in the code. Next, we study a very simple model of coevolution—the double-ring toy model. In the double-ring toy model, we can directly observe how genetic codes affect the composition of messages and how messages condition changes in the code. These observations reveal that a coevolutionary process that begins with a highly ambiguous primitive code gives rise to three well-defined categories of code modifications which we call load-minimizing steps, diversifying steps, and reassignments. The effect of errors in replication and translation on the evolving code is mediated through load-minimizing and diversifying modifications. The evolution of a code that proceeds through these modifications generates a frozen code that is both suboptimally redundant and extremely robust to the errors that were present during its evolution. The principles revealed in the toy model carry over to more complex and realistic models of coevolution. In the following section, we apply these principles to explain why coevolution in more realistic models, which begins with a highly ambiguous primitive code and incorporates the qualitative errors in replication and translation we see today, consistently generates frozen codes that have the error-correcting organization of the standard code and a characteristic level of suboptimal redundancy (compare the code in Fig. 1C, which is the outcome of our simulations of code-message coevolution, with the standard code in a similar representation in Fig. 1B). These results suggest that the theory of coevolution provides a much "tighter" explanation for the properties of the standard genetic code than that envisioned by Crick. In the Discussion we reevaluate the potential of coevolution as an explanation for the properties of the standard code and suggest that further understanding of its role in shaping the code may be gained by studying the extent to which the outcome of coevolution is robust to other factors that were present during the evolution of the code.

A Mathematical Framework for Code-Message Coevolution

We describe the mathematical framework for the study of code-message coevolution in two steps. First, we describe how we model an individual's genes and genetic code, and how they determine an individual's protein distribution and fitness. Second, we describe how we model the dynamic process of code-message coevolution in a population of such individuals. The mathematical form of the coevolutionary relations is described in the following section; however, they are not required for the reading of the rest of the paper. The assumptions of the model are the result of a combination of biological and methodological considerations. Unless these considerations are absolutely necessary for the understanding of the model we have deferred them, as well as the formal definition of the models, to the supplementary online material.

Figure 2 depicts an individual in our models. The *genotype* of each individual consists of a message (A), which is a vector of codons that stands for the concatenation of all the protein-coding regions, and a genetic code (C). In the models studied here we do not consider stop codons. The *phenotype* of each

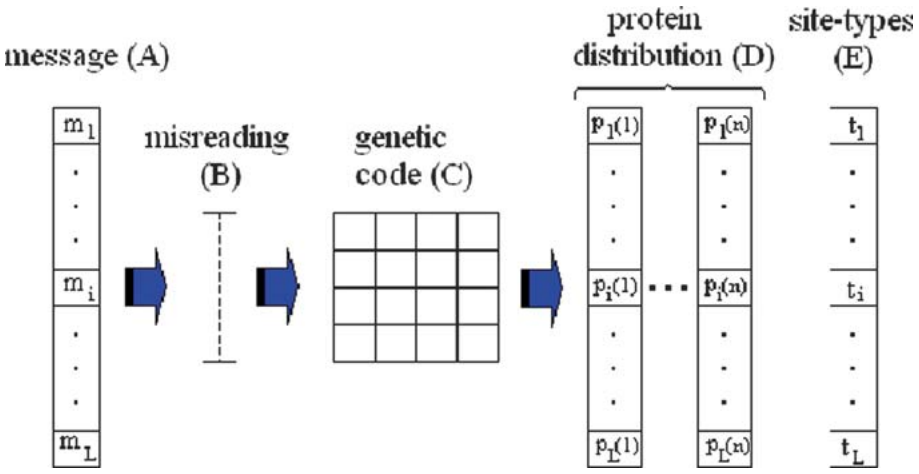

Fig. 2. The representation of an individual in our models. The message is a vector of codons that stand for all the protein coding regions (**A**). The message is translated in accordance with the code (**C**), where translation incorporates errors such as misreading (**B**). The product of translation is a protein distribution (**D**). Protein sites are classified into types (**E**), where site i is of type t_i.

individual consists of a protein distribution (D), which is the outcome of translating the message using the code, where translation incorporates errors such as misreading (B). For simplicity we refer to a single translational product of the message as a protein rather than as a concatenation of proteins. Because of the errors in translation and because we allow an evolving genetic code to be probabilistic, each codon can translate into more than one amino acid. Therefore the outcome of translation is a distribution of proteins rather than a single long protein.

To define the *fitness* associated with the protein distribution, we take advantage of an important simplification: the selection on proteins that is relevant to code evolution takes a much simpler form than the selection at a given protein site. The fitness effects that are relevant to code evolution are those that arise when a mutant code appears. Because a mutation in the genetic code changes the meaning of a codon wherever that codon appears, it causes the simultaneous substitution of one amino acid by another across many sites in many protein contexts. Thus, the fitness effects associated with a change in the code can be expressed as an average across many sites in many protein contexts. Such an average amplifies the fitness effects of amino acid substitutions that are shared by many protein sites while it diminishes the importance of fitness effects that are idiosyncratic to any specific protein site (see Sella and Ardell [2002] and the online supplementary material for further considerations and evidence that support this assumption). Because of this consideration, we assume that protein sites can be classified into a small number of types (E), which we call site-types, where each site-type reflects a form of selection on amino acid residues that is shared by the protein sites of that type. Here we use the simplest version of this assumption: we assume that sites in proteins can be classified into types according to the amino acid that best fits their requirements. When an amino acid β appears at a protein site in which amino acid α is the most fit it will contribute an amount $w(\beta,\alpha)$ to the overall fitness. To define this amount we assign a coordinate, which stands for a physicochemical property such as polar requirement, to each amino in our models. We then define this amount as

$$w(\beta,\alpha) = \phi^{d(\beta,\alpha)}, \qquad 0 < \phi < 1 \tag{1}$$

where $d(\beta,\alpha)$ represents the physicochemical distance between amino acid β and amino acid α, and the selection parameter ϕ determines the intensity of selection for biochemical accuracy. Note that unlike the conventional definition of a selective coefficient, here selection becomes weaker when the selection parameter ϕ is increased. Overall fitness is calculated by multiplying the fitness contributions across the sites of a single protein and arithmetically averaging the products across proteins in the distribution.

We assume that messages change much faster than codes. Code-message coevolution under this assumption is described in Fig. 3. At the initial step $t = 0$ all the individuals in the population have the same initial code c_0 (1). In all the models studied here we assume that this initial code is uniformly ambiguous; i.e., each codon can translate into any amino acid with equal probability. Because we assume messages change much faster than codes, the composition of messages in a population with a given code c_t ($t \geq 0$) always attains a state of equilibrium that balances between mutations in the messages and the selection exerted on the amino acids in proteins before any change in the code occurs (2). The composition of messages at mutation-selection balance with code c_t can be calculated and characterized in terms of the codon usage at each of the site-types (see mathematical section below). After the messages attain mutation-selection balance, the set of mutant codes that derive from c_t is generated (3). Here we assume that a mutation in the code changes the meaning of one codon so that it encodes for a single

amino acid, where previously it was in the ambiguous initial condition or coded for a different amino acid. We say that a mutant code meets the invasion criterion (4) if the fitness of an individual with the mutant code and a wild-type message—the messages at mutation-selection balance conditional on the wild-type code—is greater than that of an individual with both the wild-type code and a wild-type message (mathematical expressions for both are derived in the next section). Here we assume that if any of the code mutants meet the invasion criterion, the mutant code that confers the maximal fitness when it appears will succeed the existing code (5). Once a new code takes over the population the process returns to (2). When none of the code mutants meets the invasion criterion, the coevolutionary process comes to a halt, and the existing code is then the frozen outcome of evolution (6). Note that these dynamics capture the essential nature of the coevolutionary process: a given genetic code determines the composition of messages at mutation-selection balance, and in turn, this composition of messages conditions how the genetic code is allowed to change.

A model of code-message coevolution within this framework is defined by specifying the codon and amino acid spaces. The codon space is defined by the set of codons and the systematic errors of replication and of misreading in translation, which determine the relations between codons. We say that codons are "close" or "distant" if they are often or are rarely, respectively, interchanged for one another in replication and translation. The amino acid space and the corresponding site-type space are defined by the set of possible amino acids and the physicochemical relations between them, which we summarize in terms of the distance d. We say that amino acids are "close" or "distant" if they are physicochemically similar or dissimilar, respectively, to one another. In all the models we investigate here, we assume that each of the site-types is present at an equal frequency.

The Mathematical Form of Code-Message Coevolutionary Relations

First we describe how a given genetic code determines composition of messages at mutation-selection balance. The effective genetic code, which incorporates both the evolvable component of the code and the errors in translation, takes a matrix form $\mathbf{c}_{\text{eff}} = \{c_{eff}(\beta|i)\}$, where $c_{eff}(\beta|i)$ is the probability that codon i is translated into amino acid β. Throughout this review, for simplicity we assume that errors in replication and translation remain constant during the evolution of the code. When we incorporate translational misreading the effective code is given by

$$c_{eff}(\beta|i) = \sum_j c_{ev}(\beta|j)R(j,i) \tag{2}$$

where the $\mathbf{c}_{ev}$ matrix is the evolvable component of the code, and the matrix $\mathbf{R}$ describes the misreading probabilities, namely, $R(j,i)$ is the probability that codon i is read as codon j. It is easy to show that, under our fitness scheme, the codon distributions at different sites are independent at mutation-selection balance (Sella and Ardell 2002). Therefore, we can describe the composition of messages at mutation-selection balance in terms of the codon usage at each type of site. The codon usage at a site of type α is a vector $\vec{\mathbf{u}}(\alpha) = \{\mathrm{u}(i|\alpha)\}$, where $u(i|\alpha)$ is the frequency at which codon i is used at a site of type α. At mutation-selection balance the codon usage $\vec{\mathbf{u}}(\alpha)$is uniquely characterized by the fact that it remains constant under the action of selection and mutation. Namely, given an effective code $\mathbf{c}$, the codon usage at a site of type α is given as the unique positive eigenvector solution to the equation (Sella and Ardell 2002)

$$\boldsymbol{\mu}\mathbf{S}_\alpha\vec{\mathbf{u}}_c(\alpha) = \lambda_c(\alpha)\vec{\mathbf{u}}_\mathrm{c}(\alpha) \tag{3}$$

where μ is the matrix that describes mutation rates between codons, $\mathbf{S}_\alpha$ is the selection matrix that describes the selection on codons at a site of type α, which is given by

$$\mathbf{S}_\alpha = diag(\sum_\beta w(\beta,\alpha)c(\beta|i)) \tag{4}$$

and $\lambda_\mathrm{c}(\alpha)$ is the positive eigenvalue. We solve Eq. 3 at the beginning of each cycle of the coevolutionary dynamic (Fig. 3 [2]), to find the composition of messages at each of the site-types given the genetic code.

Second, we describe how the codon usage conditions the invasion of mutant codes. According to our invasion criterion, a mutant code $\mathbf{c}'$ can invade if the fitness it confers when it is presented with the existing messages is greater than or equal to that of the wild-type code $\mathbf{c}$ with the existing messages (both $\mathbf{c}$ and $\mathbf{c}'$ refer to effective codes). To determine the mathematical form of the invasion criterion, we therefore require expressions for the fitness corresponding to a combination of a code and a message distribution. In our multiplicative fitness scheme across sites within a protein, given a code $\mathbf{c}$ and codon usages $\{\vec{\mathbf{u}}(\alpha)\}_\alpha$, the fitness of the protein distribution is

$$w(\{\vec{\mathbf{u}}(\alpha)\}_\alpha, \mathbf{c}) = \prod_\alpha (w(\vec{\mathbf{u}}(\alpha), \mathbf{c}))^{l_\alpha} \tag{5}$$

where $w(\vec{\mathbf{u}}(\alpha), \mathbf{c})$is the fitness contribution of a site of type α, and l_α is the number of protein sites of type α. In our additive fitness scheme across proteins in the distribution, the fitness contribution of a site of type α across the distribution is given by

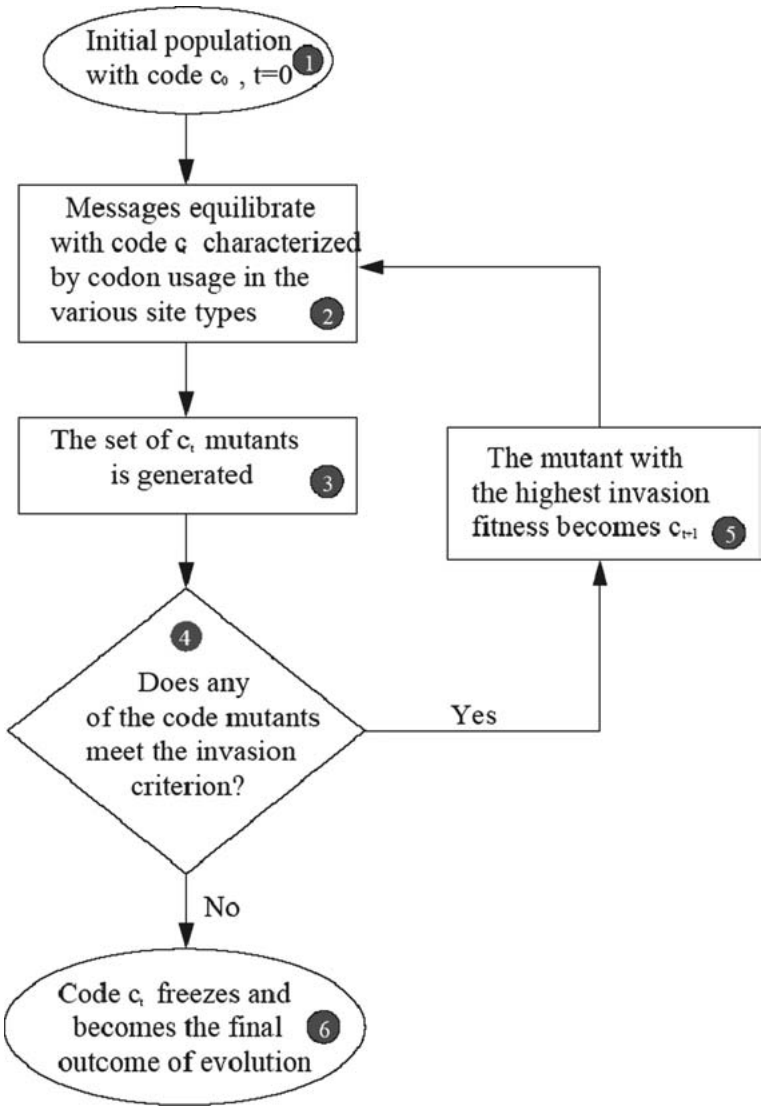

Fig. 3. The simplified code coevolutionary dynamics in the quasistatic approximation. The numbers that appear in the boxes refer to the explanation in the text.

$$w(\vec{\mathbf{u}}(\alpha), \mathbf{c}) = \sum_{\beta} \sum_{\mathrm{j}} w(\beta, \alpha) c(\beta|j) u(j|\alpha) \qquad (6)$$

When the codon usage is at mutation-selection balance with the code **c**, this fitness contribution is precisely the eigenvalue in Eq. 3 (Sella and Ardell 2002), namely,

$$w(\vec{\mathbf{u}}_{\mathrm{c}}(\alpha), \mathbf{c}) = \lambda_c(\alpha) \qquad (7)$$

Thus, the mathematical form of the criterion that determines when a code mutant $\mathbf{c}'$ can invade a wild-type code **c** is

$$w(\{\vec{\mathbf{u}}_{\mathrm{c}}(\alpha)\}_{\alpha}, \mathbf{c}') = \prod_{\alpha} (w(\{\vec{\mathbf{u}}_{\mathrm{c}}(\alpha)\}, \mathbf{c}'))^{l_\alpha} \geq \prod_{\alpha} (\lambda_c(\alpha))^{l_\alpha} = w(\{\vec{\mathbf{u}}_{\mathrm{c}}(\alpha)\}_{\alpha}, \mathbf{c}) \qquad (8)$$

Criterion 8 is evaluated for each of the code mutants at the end of each cycle of the coevolutionary dynamic (Fig. 3 [4]).

A Simple Model of Code-Message Coevolution

To gain an intuitive understanding of the way a genetic code is shaped through code-message coevolution we examine a very simple model of coevolution: the double-ring toy model, depicted in Fig. 4. The term toy model, borrowed from physics, refers to a model that is unrealistically simple and serves for explicatory purposes. The rules of code evolution take their simplest form in the double-ring toy model because the codon and amino acid spaces in this model have the same simple topology. In this model codons are organized on a ring (Fig. 4A, left). This means that each codon can mutate to become each of its neighbors on the ring (the probability of mutation per generation is $\mu = 0.01$). Amino acids and the site-types that correspond to them are organized on a ring of circumference 1, which stands for a normalized physicochemical index (Fig. 4A, right). Namely, the fitness contribution of an amino acid β at a site of

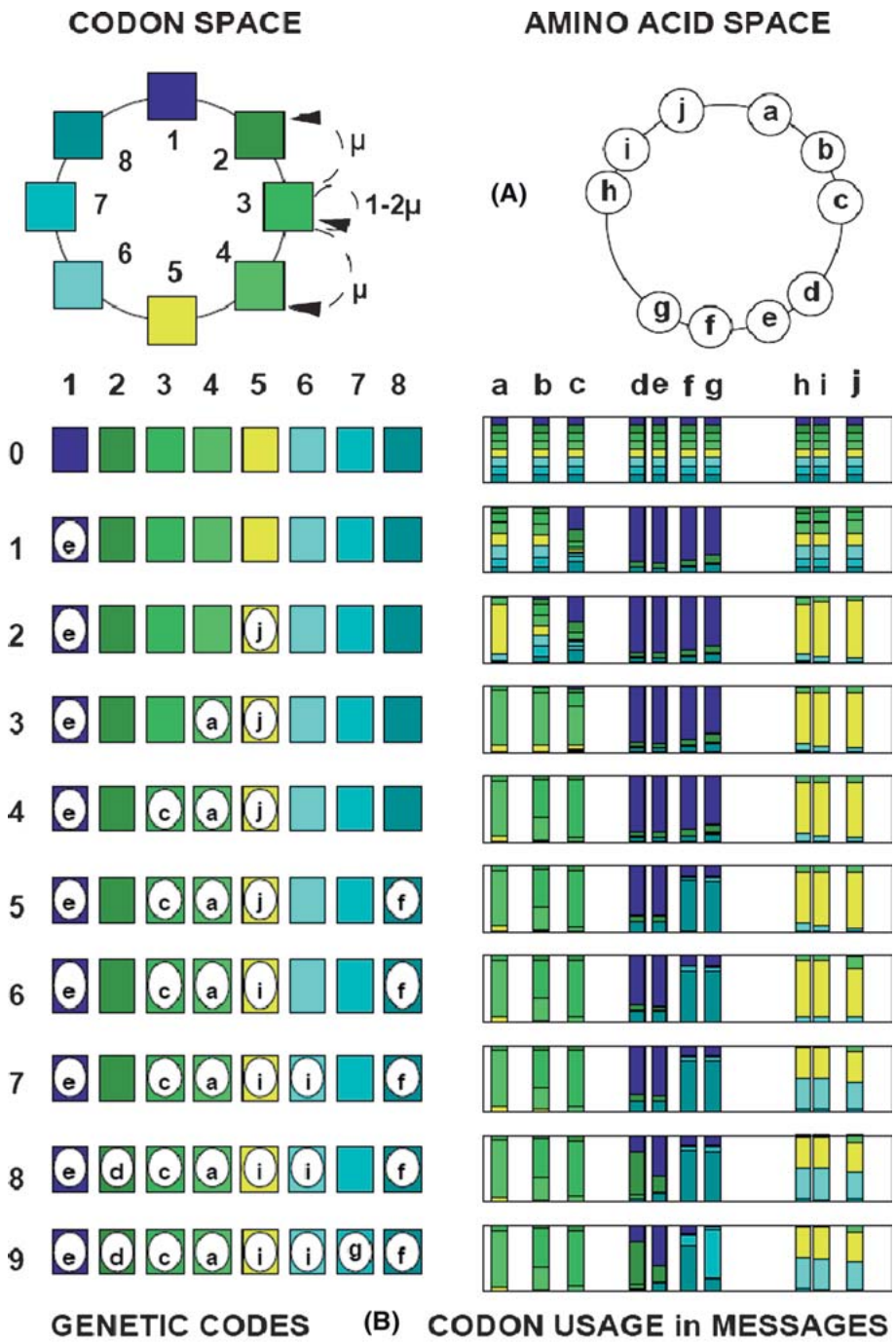

Fig. 4. Code-message coevolution in the double-ring toy model. **A** Codons and amino acids are organized on rings. Codons are assigned a number and a color, corresponding to their position on the ring. Amino acids and the corresponding site-types are assigned a letter according to their position on the ring. **B** The codes corresponding to successive evolutionary steps are presented on the left. In step 1, codon 1 is assigned amino acid **e**; the other codons which are still in the initial, uniformly ambiguous, state appear with no letter. The equilibrium codon usage corresponding to each code appears on its right. For example, at step 0, when all the codons are in the initial, uniformly ambiguous state, all the codons are equivalent and therefore their usage at the different site-types is equal. See text for the analysis of the whole evolutionary trajectory.

type α is determined by the distance between amino acid α and amino acid β, $d(\beta, \alpha)$, that is measured on the ring (the selection parameter is taken to be $\phi = 0.25$).

Before we analyze this model in detail, let us examine how the propensity of a code to change depends on the amino acid vocabulary it encodes. Figure 5 depicts the number of code mutants that meet the invasion criterion as a function of the code's amino acid vocabulary. As the coevolutionary process proceeds, the code, message, and proteins are endowed with a finer structure, such that the possibilities of change that do not reduce fitness are gradually narrowed down. When the ambiguity in the code is reduced and its vocabulary grows, the proteins become more intricate and the messages impose more restrictive conditions on changes in the code. Ultimately, the code freezes, as reasoned by Crick (1968). This funneling behavior, which Woese (1998) called evolutionary annealing, is a general property of code-message coevolution.

Figure 4B describes a simulation of code-message coevolution in the double-ring model. Once the genetic codes encode for some amino acids, the forces

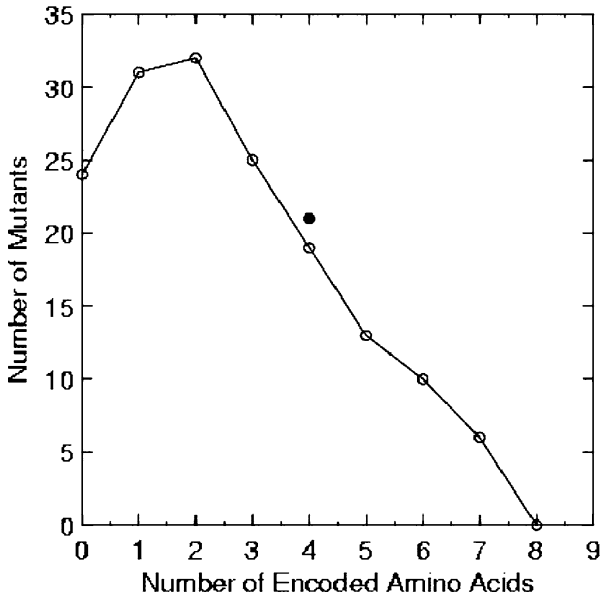

Fig. 5. Evolutionary annealing in the double-ring model. The graph depicts the number of code mutants that meet the invasion criterion as a function of the number of explicitly encoded amino acids. The initial rise in the number of possible mutants reflects the increase in possible changes after the symmetry in codon usage, which corresponds to the initial code, is broken (see Fig. 4B). The black point refers to the reassignment at time step $t = 6$ (see Fig. 4B).

of mutation and selection determine the profiles of codon usage at the different site-types. For example, consider the profiles of codon usage at step 1, when codon 1 encodes for amino acid **e**, while the other codons are still in the uniformly ambiguous initial state. Due to selection, the usage of codon 1 is high at sites where amino acid **e** is desirable, such as at sites of types **d**, **e**, **f**, and **g**, and low at sites where amino acid **e** is undesirable, such as at sites of type **j**. The codons close to codon 1, such as 2 and 8, have a usage profile that differs in magnitude but is similar in trend to that of their encoding neighbor, because of the abundance, or lack, of mutational flow from codon 1. By the same token, the codons that are farthest away from codon 1, such as codon 5, have a usage profile that is opposite in trend from that of codon 1: their usage is highest at sites of type **j** and lowest at sites of type **e**. The profiles of codon usage in messages dispose the coevolutionary process toward three kinds of modifications in genetic codes:

- *A diversifying step*. Step 2 in Fig. 4 is a diversifying step. In this step, codon 5, which is antipodal to codon 1 on the codon ring, is assigned amino acid **j**, which is antipodal to the encoded amino acid **e** on the amino acid ring. More generally, for two codons that are far from each other in codon space, if one codes for an amino acid, the other one is assigned an amino acid that is far from the encoded amino acid. Codon usage with the pre-existing code is disposed towards such a modification in the code because the usage of the distant non-encoding codon is already higher at sites in which the encoded amino acid is undesirable, and lower at sites in which the encoded amino acid is desirable. Therefore, assigning a distant amino acid to the distant non-encoding codon can increase fitness with the preexisting codon usage.
- *A load-minimizing step*. Steps 3, 4, 5, and 7–9 in Fig. 4 are load-minimizing. In step 3, codon 4, which is a neighbor of codon 5 on the codon ring, is assigned amino acid **a**, which is similar to amino acid **j**, encoded by codon 5. In general, when two codons are neighbors in codon space, and one of the two codes for an amino acid, the other one is assigned a similar amino acid. Codon usage with the preexisting code is disposed toward such a modification in the code because the usage of the encoding codon's neighbor is already higher at sites where the encoded amino acid is desirable and lower at sites where this amino acid is undesirable. Therefore, assigning a similar amino acid to the neighboring codon increases fitness with the preexisting codon usage.
- *A codon reassignment*. Step 6 in Fig. 4 is a reassignment. The addition of amino acid **a** in step 3 reduced the usage of codon 5 at sites of type **a**; this enables codon 5 to be reassigned amino acid **i**, which better meets the requirements of its modified usage profile. In general, the assignment of an amino acid to one codon releases usage constraints on other encoding codons, which can then be reassigned to better meet their modified usage requirements.

The notions of load-minimizing and diversifying steps are heuristic articulations of the rules of code modification that arise from the underlying coevolutionary dynamics, and govern the evolutionary shaping of genetic codes. These rules explain why code-message coevolution produces a genetic code that is both robust to error and redundant. They associate similar amino acids with close codons and very dissimilar amino acids with distant codons. As a result, in the final code in Fig. 4 the ring in amino acid space is embedded in the ring of codons in a structure-preserving manner, and therefore this frozen code is robust to error. Although code-message coevolution produces a very well-organized code, this code is not optimal: the frozen code in this example is redundant, and it is easy to show that if, for example, codon 6 would code for amino acid **j** rather than **i**, thus increasing the code's vocabulary, the code would confer higher overall fitness. Such disadvantageous redundancy is an intrinsic result of coevolutionary "traps." The usage of ambiguous codons that are close to assigned codons is strongly biased toward a profile that favors the use of amino acids that are

already encoded (for example, the usage of codon 6 at step 6 in Fig. 4). This biases load-minimizing modifications to assign amino acids that are more similar, and sometimes identical, to the amino acids that are already encoded, even though a more distinct amino acid may ultimately confer higher fitness (for example, the assignment of amino acid **i** to codon 6 at step 7 in Fig. 4). Once a codon is assigned redundantly, its usage is altered such that the redundancy becomes imprisoned. Although sometimes reassignments can set a codon free, more often, once the redundancy has appeared it remains irreversibly trapped in the code. Thus, the same principles that explain how the code becomes advantageously robust to error can also explain how it becomes disadvantageously redundant. With these concepts in place, we can explain the evolution of the patterns corresponding to both error-correction and redundancy in the standard code.

More Realistic Models of Coevolution

The Evolution of Robustness to Error

Figure 6A depicts the evolution of a genetic code in a more biologically realistic coevolutionary model, which incorporates transition bias in mutation. In the more realistic model, a codon is composed of two letters over the standard alphabet of four bases. Mutations occur among the bases, where transitions occur at a rate that is κ times higher than the rate of transversions. The amino acid space, and the site-type space that corresponds to it, consists of 20 members that are organized along a one-dimensional interval that corresponds to a normalized physicochemical property (requirement). In a transition-biased mutation structure, which characterizes replication in biological systems, the codon space consists of four blocks (see step 0 in Fig. 6A), corresponding to first- and second-position *pyrimidines* {U, C} or *purines* {A, G}. Within a block each codon has two closest neighbors, which are one transition away, and a neighbor which is two transitions away. Each block has two adjacent blocks which are one transversion away, and an antipodal block which is two transversions away. This structure of codon space participates in determining the course of code evolution by defining the regions of codon space across which load-minimizing and diversifying steps occur.

Code-message coevolution with transition bias (Fig. 6A) can be explained using the heuristic terms we have defined above:

- Step 2 is a diversifying step. It associates codon AA, which is antipodal to the existing encoding codon UU, with amino acid 2, which is at the end of the amino acid space furthest from the existing encoded amino acid 10. Steps 13 and 17 are also diversifying steps, which initiate the encoding in a block by associating amino acids that are far from those encoded by the other blocks.
- Steps 4, 6–8, 10, 12, 14–16, and 18–20 are load-minimizing steps. In these steps, codons that have encoding neighbors within their block are assigned amino acids similar to those encoded by their neighbors.
- Steps 3, 5, 9, 11, and 21–26 are reassignments.

The frozen code at the end of the evolutionary trajectory shown in Fig. 6 exhibits the four-block pattern, which is precisely the pattern that makes it robust to the errors of transitionally biased mutation. This four-block pattern was generated through load-minimizing and diversifying steps, which were induced by the mutational errors, but driven by the local fitness requirements of code-message coevolution. Similarly, in Fig. 6B, we see that the final code produced by coevolution in the presence of misreading in the first codon position (see regularity i, at the beginning of the paper) is error-correcting with respect to that error: the frozen code is organized in columns, such that amino acids are more similar to each other along the first codon position than along the second.

When the qualitative characteristics of both types of errors and their relative magnitudes are incorporated, code-message coevolution reproduces the qualitative organization of the standard code. Generally, when both misreading at the first codon position and transition bias in mutation are introduced, the organization of the frozen codes varies according to the type of errors that is dominant at the first codon position. When transitional mutation dominates over misreading, frozen codes (Fig. 7[A1]) pronounce the four-block structure that is error-correcting for transition bias in mutation. However, as we review in the supplementary online material, empirical evidence strongly suggest misreading dominated the error along the first codon position. In the more realistic case (Fig. 7[A2]), where misreading is dominant, the model reproduces the salient organizational properties of the standard code that we have set out to explain; namely, (i) amino acids are more similar along the first codon position than they are along the second, and (ii) amino acids associated with pyrimidine, or purine, bases along the second codon position, are more similar within these sets than they are between them. To quantify this transition, we apply the same method that has been used to measure the organization of the standard code (Ardell 1998). We compare each frozen code with a large set of codes, which are generated by randomly permuting the amino acid vocabulary of the frozen code between its codons. Then we measure the fraction of permuted

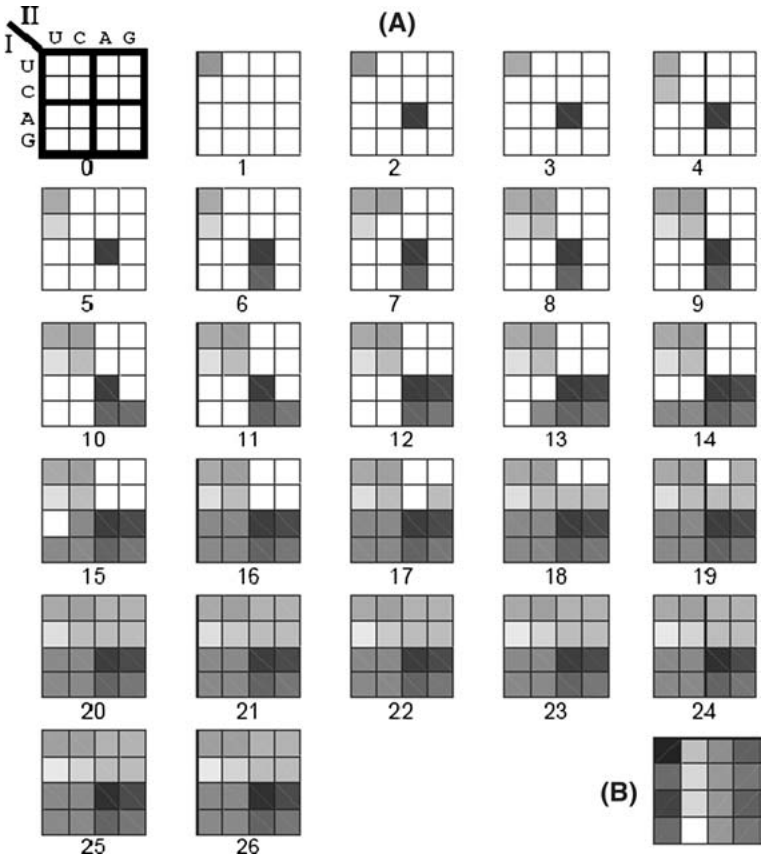

Fig. 6. **A** A typical evolution in the more biologically realistic model with transitional mutational bias (transition bias was taken to be, $\kappa = 7$, $\phi = 0.92$, and $\mu = 0.0006$). The 20 amino acids were chosen from a uniform distribution on the (0,1) interval, which stands for a physicochemical property such as polar requirement, and they are represented by a gray scale in which darker shades correspond to a position closer to 1. At step 1, codon UU is assigned amino acid 10, which is in the middle of the range of the physicochemical property. The other codons, which are still in the initial state, appear with a white entry. See text for the analysis of the whole evolutionary trajectory. **B** The frozen code that results from a typical evolution with uniform misreading in the first codon position (ϕ and μ as above; $e_1 = 0.01$).

codes $P(X)$ that are more conservative than the frozen code along the X dimension of the code. Figure 7B shows the results averaged over 42 simulations with different randomly chosen amino acid distributions: on the left, where misreading is low, $P(Itv) \approx P(\mathrm{I}Itv) >> P(Its) \approx P(Its)$, corresponding to the four-block pattern, and on the right, in the more realistic case, $P(IIts) >> P(IIts)$, $P(Its)$, $P(Itv)$, corresponding to the organization of the standard code (compare Fig. 7C). These results establish that code-message coevolution robustly reproduces the salient properties of the standard code when we vary the distribution of available amino acids. In Ardell and Sella (2002), we also show that these patterns remain robust when other parameters, such as the mutation rate and the degree of transition bias, are varied.

The Evolution of Suboptimal Redundancy

Figure 8 shows the average size of the frozen code's vocabulary as a function of the intensity of selection and the rate of mutation. It is apparent that a redundant frozen code is a robust outcome of the coevolutionary dynamics. While the codes in the model are able to incorporate up to 16 amino acids, the number of amino acids that actually become encoded is considerably smaller. This redundancy can be explained in terms of the coevolutionary traps we have encountered in the double-ring toy model. Namely, at a given stage of coevolution existing codon usage creates a bias toward the incorporation of amino acids that are already encoded, even though the incorporation of novel amino acids would ultimately confer higher fitness.

The coevolutionary dependence on mutation and selection exhibits three dynamic regimes, which are reflected in the number of encoded amino acids as well as in other attributes of the coevolutionary trajectories and frozen codes (see Ardell and Sella [2001] for a detailed account). We refer to the lower ϕ region, which corresponds to extremely strong selection, as the Crick limit. Even at the Crick limit frozen codes rarely code for more than 14 amino acids. However, the intensity of selection at the Crick limit appears to be unrealistically high. For example, assuming a selective parameter of $\phi = 10^{-3}$ and a vocabulary of 10 amino acids, then substituting a single amino acid by its best alternative ($d \approx 0.1$) causes a 50% reduction in the organism's overall fitness ($\phi^{\mathrm{d}} \approx 0.5$). In contrast, empirically manipulated genetic codes in *E. coli*, which alter translation at thousands of sites, only cause a 33% decrease in overall fitness (Döring and Marliére 1998). This decrease is likely to have been even smaller during the evolution of the standard code, when proteins were highly statistical (Woese 1967) and thus less sensitive

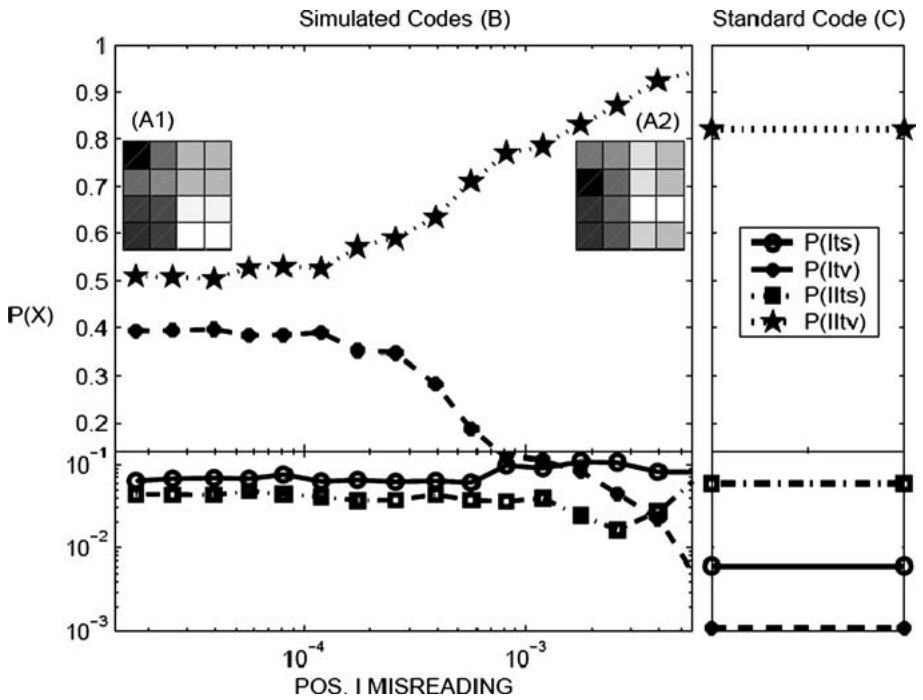

Fig. 7. A comparison of frozen evolved codes with the standard code. **A** Frozen codes that correspond to mutationally dominated (1) and misreading-dominated (2) errors. **B** Each point represents an average over 42 frozen codes that result from simulations with randomly chosen amino acid parameters, selection coefficient $\phi = 0.92$, mutation rate $\mu = 0.0006$, transition bias $\kappa = 7$, and misreading e_1 given by the horizontal axis. $P(X)$ denotes the fraction of randomly permuted codes that are more physicochemically conservative than the evolved code along the X codon dimension, where the four lines correspond to first- and second-position transitions (ts) and transversions (tv). When $P(X)$ is small, it means that the amino acids along the X codon dimension are very similar. To show the behavior of $P(X)$ on the lower end the axis is split, such that the lower values are shown on a logarithmic scale, and the higher values are shown on a linear scale. **C** The corresponding $P(X)$ values for the standard code taken from Ardell (1998).

to amino acid substitutions. We believe that the standard code evolved under conditions that correspond to the transition between the encoding plateau and the encoding catastrophe. At the encoding plateau, which refers to the midrange ϕ that extends over most of the selection parameter range, the number of encoded amino acids is stable between 10.5 and 12. At the encoding catastrophe, which corresponds to high ϕ and weak selection, the number of amino acids exhibits a dramatic decline to 2.

As we know very little about the properties of proteins at the time the code evolved, estimating the actual range of the selection parameter is very hard. However, in order to explain why we think the standard code has evolved at the boundary between the encoding plateau and the encoding catastrophe we provide two independent rough evaluations. In the supplementary online material we perform a rough calculation of the selection parameter based on the empirical results of Döring and Marliére (1998), which yields a range of $0.996 < \phi < 0.9996$ (Ardell 1999). Of course this evaluation is based on modern proteins and should therefore be taken with a grain of salt. We can also estimate the lower bound on the intensity of selection (which is an upper bound on ϕ) under the assumption that selection for physicochemical accuracy was strong enough to differentiate between two adjacent amino acids that became incorporated in the standard code. Assuming an amino acid vocabulary of 10 in our 16-codon model, 2 adjacent amino acids would be at a normalized physicochemical distance of $d \approx 0.1$. The selective coefficient corresponding to the substitution of an amino acid by its closest alternative is given by $s = 1 - (1/\phi^d)$. Therefore, if we assume an effective population size of $N = 10^5$, the requirement that $Ns \approx 1$ yields a selective intensity of $\phi = 0.9995$. Although we are reluctant to place too much weight on these estimates, we do note that they agree with each other. Within the range that extends below this bound into the encoding plateau, the redundancy of frozen codes, namely, the number of encoded amino acids per codon, ranges between 0.75 and 3. Obviously, these are very rough estimates which leave much room for further consideration and improvement. Nevertheless, we claim that these results strongly suggest that a considerable measure of the redundancy in the standard code may be an unavoidable consequence of code-message coevolution.

Discussion

Although we cannot yet ascertain the full extent to which coevolution determined the properties of the

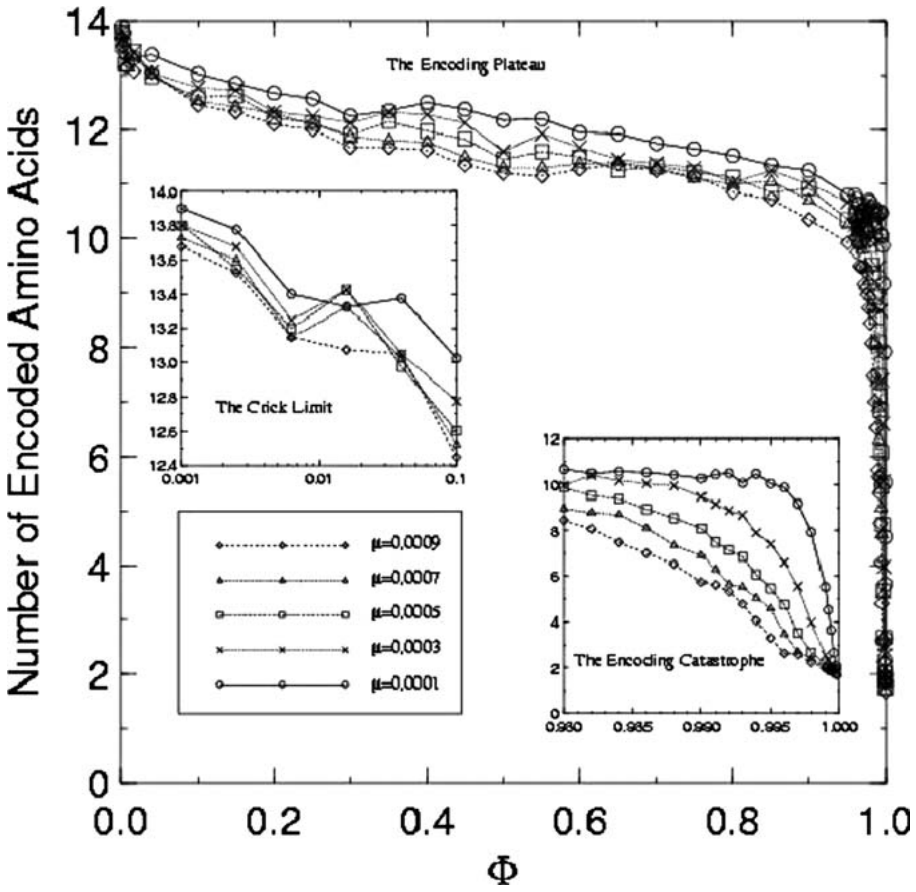

Fig. 8. The number of encoded amino acids in frozen codes as a function of selection and mutation. The model used to produce this graph incorporates uniform mutations among bases and no misreading. Each point represents an average over 40 frozen codes that result from simulating with different randomly chosen amino acid parameters and the specified selection intensity and mutation rate.

standard code, it seems clear that coevolution suggested a more powerful explanation than Crick (1968) initially proposed. According to Crick's theory, the origin of the associations between codons and amino acid properties in the standard code can be traced back to the original division of the primitive genetic code among the first encoded amino acids. He argues that the primitive code was likely to associate the same amino acid, or amino acid class, with related codons, although the more detailed form of these ancestral associations was largely accidental. Crick suggests that the amelioration of the primitive code into the code we see today proceeded through the incorporation of new amino acids and an increase in the precision of translation within amino acid classes. Coevolution would affect this process through selection to reduce the disruptive effect of changes in the code on the translation of existing messages. This selection may explain why changes in the code would tend to preserve the ancestral association between codons and amino acid properties, but it does not account for the origin of these associations or for their detailed form. The results we have reviewed suggest that coevolution in the presence of errors in replication and translation does more than preserve ancestral associations, it generates them. In all the simulations presented here, we started from a uniformly ambiguous code, which is an extreme version of the highly ambiguous code suggested by Sonneborn (1965) and others (Fitch 1966; de Duve 1995). The uniformly ambiguous code has no specificity in the association of codons and amino acids and therefore it cannot bias the outcome of evolution. The association between codons and amino acids was produced by load minimizing and diversifying modifications, which arise naturally from the coevolution of codes and messages in the presence of errors. When we incorporated the qualitative errors we see today, which represent our best guess on the characteristics of errors at the time the code evolved, the coevolutionary process did more than associate similar amino acids with related codons, it consistently generated the main organizational features observed in the standard code.

Consider, for example, the association of central pyrimidine codons (NYN) with hydrophobic amino acids and central purine codons (NRN) with hydrophilic amino acids. A theory of coevolution can explain why amino acid polarity would be divided between these sets of codons, although other factors may have determined which one of these sets is associated with hydrophobic (hydrophilic) amino acids. This division has been suggested to be the first effective discrimination incorporated during the evo-

lution of the standard code, and the analysis of tRNA phylogenies supports this assertion (Fitch and Upper 1987). Studies in protein design (Kamtekar et al. 1993; West and Hecht 1995) also suggest that this discrimination would have allowed for an effective encoding of polypeptides with characteristic structures, consisting of localized α-helices and β-sheets. It is therefore plausible that the selection on primitive polypeptides was directed primarily toward discriminating between hydrophilic and hydrophobic amino acids. Crick's theory explains why similar amino acids would be associated with related codons, but that is a long way from explaining why amino acid polarity would be divided according to the central position pyrimidines and purines. The results reviewed here show that code-message coevolution provides a direct explanation for this division. We suggest that elevated misreading in the first and third codon position in concert with transition bias in mutation would have created the most effective discrimination between codons with central pyrimidines and purines. Therefore, once the code exhibited a slight bias in polarity, even in the ambiguous translation of a single codon, diversifying modifications across these two sets of codons and load-minimizing modifications within them would have associated one set with hydrophilic amino acids and the other with hydrophobic amino acids. The original seeds of specificity that provided the bias toward one of the two alternatives may have been purely accidental, although they also may have been influenced by codon bias (Eigen and Schuster 1979), the metabolic production of amino acids (Wong 1975, 2005; Taylor and Coates 1989; Di Giulio 2004), stereochemical biases (Knight et al. 1999), and selection for the use of versatile amino acids (Ardell and Sella 2001, 2002). Although coevolution does not determine the direction of this association, it reduces numerous possible organizations to two well defined structural alternatives. It would be interesting to see whether models of coevolution that incorporate amino acid properties other than polarity, such as those proposed by Fitch and Upper (1987), can account for the division of the standard code according to these properties.

Although coevolution explains why a frozen code is likely to be redundant, it is not yet clear whether, and to what extent, it can account for the precise degree of redundancy observed in the standard code. The results we have reviewed indicate that the vocabulary of frozen codes depend on the mutational and selective parameters during evolution. In a large region of these parameters, which we call the encoding plateau, the number of amino acids in the frozen code appears to be stable. For given parameters within that region, the number of encoded amino acids does not exhibit much variation (Ardell and Sella 2001), and this stability in number does not change when typical errors in replication and translation are introduced (Ardell and Sella 2002). However, two main issues would have to be addressed before we can assess the degree to which coevolution restricts the size of the amino acid vocabulary. First, as we have seen, the size of the amino acid vocabulary does vary significantly when we cross into the parameter region we call the encoding catastrophe. Therefore, ascertaining the constraints on amino acid vocabulary would require a better evaluation of evolutionary parameters together with a more detailed study of the coevolutionary dynamics they imply. Second, the degree of redundancy imposed by coevolution appears to depend on the topology of the code. Intuitively, the probability of encountering coevolutionary traps that result in redundant amino acid assignments, such as the one we observed in the double-ring model, increases with the number of neighbors associated with each codon. We would therefore expect the doublet code to have more redundancy than a code on a ring, and by the same token, we would expect a triplet code to have more redundancy than a doublet code. It would therefore be useful to study how the vocabulary of frozen codes varies in a triplet system, which incorporates reasonable alternatives on the constraints at the third codon position.

We have recently learned about ongoing research that suggests that a theory of coevolution may provide a much "tighter" explanation for the redundancy of the standard code than we have expected. This study applies analytical methods from high-energy physics to analyze the relationship between the topology of codes and their frozen vocabularies (Tlusty 2006). It incorporates a simplified version of the coevolutionary dynamics presented here and focuses on the redundancy at the transition between the encoding plateau and the encoding catastrophe, which represents our best guess on the parameter region in which the standard code evolved. In correspondence with the results presented here, the theory predicts a vocabulary of 11 amino acids in the doublet code. Quite surprisingly, this study also indicates that triplet codes, which incorporate reasonable assumptions about the constraints at the third codon position, are expected to encode 20 amino acids. In correspondence to the crude intuition provided above, the redundancy in triplet codes ($0.56 = 1-20/45$) turns out to be greater than that in doublet codes ($0.45 = 1-11/20$). Although various perturbations can change the number of amino acids that eventually get encoded, Tlusty's work suggests that these changes would not be large and that the expected number corresponds to that observed in the code.

Asserting that coevolution is disposed toward generating the organization observed in the standard

code does not necessarily imply that this tendency is strong enough to overcome other influences that were present during the evolution of the code. Many other factors may have affected the evolving code. For example, it has been suggested that stereochemical biases in the association of codons, or anticodons, and amino acids were important at the early stages of code evolution (Crick 1968; Jukes 1973; Knight and Landweber 2000). Because empirical studies (Knight and Landweber 2000; Yarus 2000) suggest only a very partial picture on the possible direction of these associations, we have to assume some of them could have biased the primitive code in directions that opposed the course of coevolution. In view of our ignorance concerning the conditions during the evolution of the code, an affirmation of the ability of coevolution to explain the shaping of the standard code has to rely on its potency to overcome other influences. In other words, we would like to know whether, and to what extent, the outcome of coevolution is robust to other influences. This question can be addressed using the mathematical framework we have presented. For example, to study the robustness to stereochemical association one can examine how biased primitive codes and code mutation schemes affect the shape of the frozen code. Although the work we have reviewed began to address some aspects of robustness, such as the dependence of frozen codes on evolutionary parameters, much further work is required before we can assess the robustness of, and hence our confidence in, the coevolutionary explanation.

The proposition that errors in replication and translation were important in the coevolutionary shaping of the standard code is substantially different from claims that the standard code was shaped by selection to minimize the deleterious effects of these errors (Alff-Steinberger 1969; Ardell 1998; Freeland and Hurst 1998; Goldberg and Wittes 1966; Nirenberg et al. 1963; Sonneborn 1965; Swanson 1984; Woese 1965; Woese et al. 1966; Zuckerkandl and Pauling 1965; see Sella and Ardell [2002] for a review of the various error-correcting hypotheses). Because these proposals have often been confused as being similar (despite the fact that this distinction was convincingly explained by Crick [1968] and Woese [1965] and, more recently, by Freeland [2002] and Freeland et al. [2003]), we briefly review the main differences between them. First, they differ in mechanism. While theories of optimization assume that code variants, which differ in their organization, compete at minimizing the deleterious effects of errors, a theory of coevolution postulates that code variants, characterized by the addition of an amino acid or a reduction in ambiguity, compete with their predecessor at translating the messages bequeathed by that predecessor. Second, the difference in mechanism implies a difference in plausibility. For selection on error minimization to distinguish effectively between alternative codes very special conditions must be met. For example, at each stage in evolution, alternative codes must have the same amino acid vocabulary and the messages that accompany them must be equilibrated; otherwise, differences in their amino acid vocabulary, or in the transient use of amino acids in their messages, are likely to overwhelm the relatively small advantage associated with an error correcting organization. In contrast, the coevolution of codes and messages does not require special conditions. It only requires that whenever a variant code appears, it is presented with the messages bequeathed by its precursor. Third, they differ in the way they explain the organization of the standard code. A theory of error minimization explains the organization of the code as an outcome of an optimization process. The optimization principle cannot be associated with error correction alone, because the most error-correcting codes are those which encode for a single amino acid. A theory of error minimization should therefore be founded on a justifiable optimization function that balances between the vocabulary of a code and its robustness to error, and this function should be maximal at the levels of redundancy and error correction observed in the standard code. A theory of code-massage coevolution explains both robustness to error and redundancy as an outcome of the local selection on codes and messages at different stages in the evolutionary process. This outcome depends on the parameters during the evolutionary process.

The theory of code-message coevolution illustrates a general principle that may apply to the evolution of other biological systems. Namely, the shaping of biological systems is, to a varying extent, the outcome of the coevolution of their parts, and the laws that govern this coevolution derive from the way these parts operate together. In the case of code evolution, these laws derive from the way genes are translated according to the genetic code to produce proteins. This simple functional relation determines how a given genetic code and the errors in replication and translation shape the composition of messages, and how a given composition of messages conditions changes in the code, where both are mediated through the selection on proteins. The work reviewed here shows that recurrent application of these rules has the potential to explain how the code was shaped and, most importantly, how the code was endowed with the properties we see today. The extent to which the coevolutionary principle has dictated the shaping of the code would have to be resolved, as is often the case with other evolutionary principles, based on its potency to overcome historical and biochemical factors that may have disrupted its operation.

Acknowledgments. We thanks Marcus W. Feldman, Ilan Eshel, Aaron Hirsh, Dmitri Petrov, Michael Lachmann, Tuvik Becker, Ben Kerr, Jennifer Hughes, Steve Freeland, Rob Knight, Erel Levine, Emile Zuckerkandl, and three anonymous reviewers for valuable comments at various stages of this work. We also thank Tsvi Tlusty for his comments and for sharing his exciting results with us. The research of D.A. and G.S. was partly supported by NIH Grants GM28016 and GM28428 to Marcus W. Feldman. G.S. was also supported by a Koshland Scholarship and by the Center for Complexity Science of the Yashaya Horowitz Association.

References

Alff-Steinberger C (1969) The genetic code and error transmission. Proc Natl Acad Sci USA 64:584–591

Ardell DH (1998) On error-minimization in a sequential origin of the standard genetic code. J Mol Evol 47:1–13

Ardell DH (1999) Statistical and dynamical studies in the evolution of the standard genetic code and a biochemical study of variation in resilin from Schistocerca gregaria. *PhD thesis. Stanford University*, Stanford, CA

Ardell DH, Sella G (2001) On the evolution of redundancy in genetic codes. J Mol Evol 53:269–281

Ardell DH, Sella G (2002) No accident: genetic codes freeze in error-correcting patterns of the standard genetic code. Phil Trans R Soc Lond B 357:1625–1642

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

Davies J, Gilbert W, Gorini L (1964) Streptomycin, suppression and the code. Proc Natl Acad Sci USA 51:883–890

Davies J, Jones DS, Khorana HG (1966) A further study of misreading of codons induced by streptomycin and neomycin using ribopolynucleotides containing two nucleotides in alternating sequence as templates. J Mol Biol 18:48–57

de Duve CR (1995) Vital dust Basic Books, New York

Di Giulio M (1994) The phylogeny of tRNAs seems to confirm the coevolution of the origin of the genetic code. Orig Life Evol Biosph 25:549–564

Di Giulio M (2004) The coevolution theory of the origin of the genetic code. Physics Life Rev 1:128–137

Döring V, Marliére P (1998) Reassigning cysteine in the genetic code of *Escherichia coli*. Genetics 150:543–551

Eigen M, Schuster P (1979) The hypercycle: a principle of natural self-organization Springer, Berlin

El'skaya AV, Soldatkin AP (1985) The bases of translational fidelity. Molekulyarna Biol 18:1163–1180

Fitch WM (1966) Evidence suggesting a partial, internal duplication in the ancestral gene for heme-containing globins. J Mol Biol 16:9–16

Fitch WM, Upper K (1987) The phylogeny of tRNA sequences provides evidence for ambiguity reduction in the origin of the genetic code. Cold Spring Harbor Symp Quant Biol 52:759–767

Francklyn C, Perona JJ, Puetz J, Hou YM (2002) Aminoacyl-tRNA synthetases: versatile players in the changing theater of translation. RNA 8:1363–1372

Freeland SJ (2002) The Darwinian code: An adaptation for adapting. J Gen Progr Evolv Machines 3:113–127

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

Freeland SJ, Wu T, Keulmann N (2003) The case for an error minimizing standard genetic code. Orig Life Evol Biosph 4–5:457–477

Freese E (1961) Transitions and transversions induced by depurinating agents. Proc Natl Acad Sci USA 47:540–545

Gojobori T, Li W-H, Graur D (1982) Patterns of nucleotide substitution in pseudogenes and functional genes. J Mol Evol 18:360–369

Goldberg AL, Wittes RE (1966) Genetic code: aspects of organization. Science 153:420–424

Haig D, Hurst LD (1991) A Quantitative measure of error minimization in the genetic code. J Mol Evol 33:412–417

Hixon JE, Brown WM (1986) A comparison of small ribosomal RNA genes from the mitochondrial DNA of great apes and humans: sequence, structure, evolution and phylogenetic implications. Mol Biol Evol 3:1–18

Jukes TH (1973) Arginine as an evolutionary intruder into protein synthesis. Biochem Biophys Res Commun 53:709–714

Kamtekar S, et al. (1993) Protein design by binary patterning of polar and nonpolar amino acids. Science 262:1680–1685

Knight RD, Landweber LF (2000) Guilt by association: the arginine case revisited. RNA 6:499–510

Knight RD, Freeland SJ, Landweber LF (1999) Selection, history and chemistry: the three faces of the genetic code. Trends Biochem Sci 24:241–249

Krishna RG, Wold F (1993) Posttranslational modification of proteins. Adv Enzymol Relat Areas Mol Biol 67:265–298

Nirenberg MW, Jones OW, Leder P, Clark BFC, Sly WS, Pestka S (1963) On the coding of genetic information. Cold Spring Harbor Symp Quant Biol 28:549–558

Osawa S, Jukes TH, Watanabe K, Muto A (1992) Recent evidence for evolution of the genetic code. Microbiol Rev 56:229–264

Parker J (1989) Errors and alternatives in reading the universal genetic code. Microbiol Rev 53:273–298

Petrov DA, Hartl DL (1999) Patterns of substitution in *Drosophila* and mammalian genomes. Proc Natl Acad Sci USA 96:1475–1479

Sankoff D, Morel C, Cedergren RJ (1973) Evolution of 5S RNA and the non-randomness of base replacement. Nature New Biol 245:232–234

Sella G, Ardell DH (2002) The impact of message mutation on the fitness of a genetic code. J Mol Evol 54:638–651

Sonneborn TM (1965) Degeneracy of the genetic code: extent, nature, and genetic implications. In: Bryson V, Vogel HJ (eds) Evolving genes and proteins. Academic Press, New York, pp 377–397

Swanson R (1984) A unifying concept for the amino acid code. Bull Math Biol 46:187–203

Taylor FJR, Coates D (1989) The code within the codons. BioSystems 22:177–187

Tlusty T (2006) Emergence of a genetic code as a phase transition induced by error-load topology (Submitted)

Topal MD, Fresco JR (1976) Complementary base pairing and the origin of substitution matrices. Nature 263:285–293

Vigilant L, Stoneking M, Harpending H, Hawkes K, Wilson AC (1991) African populations and the evolution of human mitochondrial DNA. Science 253:1503–1507

Wakeley J (1996) The excess of transitions among nucleotide substitutions: new methods of estimating transition bias underscore its significance. Trends Ecol. Evol 4:158–163

West WW, Hecht H (1995) Binary patterning of polar and nonpolar amino acids in the sequence and structure of native proteins. Protein Sci 4:2032–2039

Woese CR (1965) On the evolution of the genetic code. Proc Natl Acad Sci USA 54:1546–1552

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

Woese CR (1998) The universal ancestor. Proc Natl Acad Sci USA 95:6854–6859

Woese CR, Dugre DH, Dugre SA, Kondo M, Saxinger WC (1966) On the fundamental nature and evolution of the genetic code. Cold Spring Harbor Symp Quant Biol 31:723–736

Wong J (1975) A co-evolution theory of the genetic code. Proc Natl Acad Sci USA 77:1083–1086

Wong JT (2005) Coevolution theory of the genetic code at age thirty. Bioessays 4:416–425

Xue H, Tong KL, Marck C, Grosjean H, Wong JT (2003) Transfer RNA paralogs: evidence for genetic code-amino acid biosynthesis coevolution and an archaeal root of life. Gene 310:59–66

Yarus M (2000) RNA-ligand chemistry: a testable source for the genetic code. RNA 6:475–484

Zuckerkandl E, Pauling L (1965) Evolutionary divergence and convergence in proteins. In: Bryson V, Vogel HJ (eds) Evolving genes and proteins. Academic Press, New York, pp 97–166