Detecting overlapping coding sequences with pairwise alignments. (supplementary material)

Andrew E. Firth, Chris M. Brown
Department of Biochemistry, University of Otago, PO Box 56, Dunedin, New Zealand

August 16, 2005

Extract (with a few minor updates) from Supplementary Material to Firth A. E., Brown C. M., 2005, Detecting overlapping coding sequences with pairwise alignments, ***Bioinformatics*****, 21, 282–92.**

Contents

1 Frame notation

While the meaning of the forward read-direction frames $F = +1$ and $F = +2$ (relative to a reference $F = 0$ sequence) is clear, the meaning of $F = -1$ or $F = -2$ is less clear. In fact -1 and -2 are sometimes used to describe forward read-direction frames – e.g. for frameshifting, where a -1 frameshift corresponds to the $F = +2$ frame and a -2 frameshift corresponds to the $F = +1$ frame. There is no general standard for naming the reverse read-direction frames. Our convention is illustrated in Figure 1 and we give conversions to other standards in Table 1. In our system, $F = -1$ corresponds to the same frame as $F = +1$ but read in the reverse direction, $F = -2$ corresponds to $F = +2$ but read in the reverse direction, and $F = -3$ corresponds to $F = 0$ but read in the reverse direction. So the number represents the register, while the sign represents the read direction.

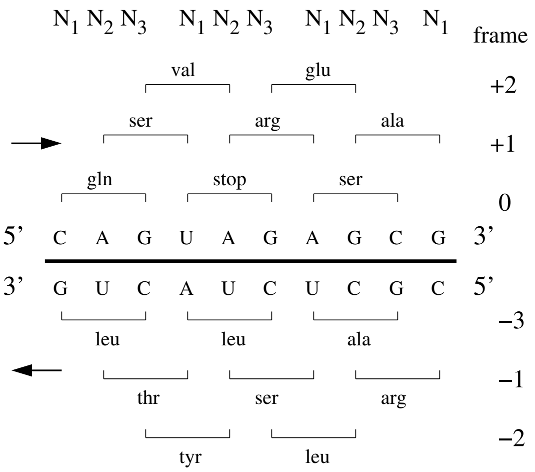


Figure 1: Naming convention for the different possible read-frames of a secondary ORF relative to a frame = 0 primary ORF. In the paper we use the terms 'primary' and 'secondary' to clarify which read-frame we are referring to; it does not imply that one ORF is more important than the other. We use 'frame = 0' to refer to a single-coding region, and 'frame = $-3, \pm1, \pm2$' to refer to double-coding regions.

Table 1: Table comparing different double-coding frame naming conventions.

Firth & Brown (2004)	0	+1	+2	−1	−2	−3
Smith & Waterman (1980)	0	1	2	4	5	3
Krakauer (2000)	-	+1	+2	−2	−1	0
Rogozin *et al.* (2002)	-	-	-	C2	C1	C3

If ORF A is in frame $F = -3$ relative to ORF B, then it is easy to see that ORF B is in frame $F = -3$ relative to ORF A. This reciprocity is true also for the -2 and -1 frames. However if ORF A is in frame $F = +1$ relative to ORF B, then ORF B is in frame $F = +2$ relative to ORF A and *vice versa*. These relationships are summarized in Table 2.

Table 2: Table showing reciprocal frames labels: if ORF2 is in frame A relative to ORF1, then ORF1 is in frame B relative to ORF2.

A	B
+1	+2
+2	+1
0	0
−3	−3
−1	−1
−2	−2

2 Details of methods

Note: The following methods were used in Firth A. E., Brown C. M., 2005, Detecting overlapping coding sequences with pairwise alignments, Bioinformatics, 21, 282–92. In the more recent paper and webserver, there have been a few minor modifications (e.g. incorporation of non-coding and mixed models) – see Firth A. E., Brown C. M., Detecting Overlapping Coding Sequences in Viral Genomes, submitted to BMC Bioinformatics, 2005, for details.

In this section we provide further details on methods that, due to space limitations, were omitted from the paper. In §2.1 we provide a brief background of standard sequence evolution models and illustrate how these models break down in the case of overlapping CDSs. In §2.2 we describe the Monte Carlo simulation algorithm. In §2.3 we describe the input nucleotide mutation matrix, amino acid substitution matrix and codon usage table. In §2.4 we describe the various statistics that we use to test for double-coding. In §2.5 we describe how we use the results of these statistics to classify a sequence as single-coding or double-coding.

2.1 Sequence evolution models

2.1.1 Sequences of nucleotides

The evolution of one sequence to another is often modelled as a Markov process. Typically one starts with a 4×4 matrix of 'instantaneous' nucleotide mutation probabilities $\mathbf{Q} = [q_{ij}]$, where q_{ij}, $i \neq j$, is the probability of nucleotide i mutating to nucleotide j in some small time interval, and the q_{ii} are fixed by the requirement that row sums are zero, i.e.

$$q_{ii} = -\sum_{j \neq i} q_{ij}. \tag{1}$$

A completely general nucleotide transition matrix has 12 free parameters. For a reversible Markov process (i.e. one for which sequence evolution is symmetric with respect to time) there are only 6 free parameters and it is convenient to parametrize the matrix as follows

$$\mathbf{Q} = \begin{bmatrix} - & a.\pi_C & b.\pi_A & c.\pi_G \\ a.\pi_U & - & d.\pi_A & e.\pi_G \\ b.\pi_U & d.\pi_C & - & f.\pi_G \\ c.\pi_U & e.\pi_C & f.\pi_A & - \end{bmatrix}, \tag{2}$$

where the π_j are the equilibrium nucleotide frequencies, the order of the rows and columns is U, C, A, G, and the diagonals are set according to equation 1. Equation 2 describes the general reversible model of Yang (1994).

The normalization of the matrix $\mathbf{Q}$ is arbitrary, but it is standard practice to normalize it so that

$$\sum_i -\pi_i . q_{ii} = 1, \tag{3}$$

which corresponds to a mean of one mutation per nucleotide per unit time interval. The matrix $[p_{ij}(t)] = \mathbf{P}(t) = \exp(\mathbf{Q}t)$ describes the nucleotide mutation probabilities after a given evolutionary time t. $\mathbf{P}(t)$ is most easily calculated by diagonalizing $\mathbf{Q}$, i.e. writing $\mathbf{Q} = \mathbf{U}^{-1}\mathbf{D}\mathbf{U}$, where $\mathbf{D}$ is diagonal. Then $\exp(\mathbf{Q}t) = \mathbf{U}^{-1}\exp(\mathbf{D}t)\mathbf{U}$, where $\exp(\mathbf{D}t)$ is just the diagonal matrix with diagonal terms $\exp(d_{ii}t)$ (see Ewens & Grant 2001 for details). To calculate the probability, or likelihood, that sequence S_1 mutates to sequence S_2 (S_1, S_2 members of a sequence alignment), after time t, one simply has to multiply the probabilities $p_{ij}(t)$ over all aligned nucleotides, viz.

$$P(S_1 \rightarrow S_2; t) = \prod_{k=1}^{N_{\text{nucleotides}}} p_{N_1^k N_2^k}(t), \tag{4}$$

where N_1^k and N_2^k are the kth nucleotides of sequences S_1 and S_2 respectively. To avoid numerical underflow in computations, this is often replaced with

$$\log P(S_1 \rightarrow S_2; t) = \sum_{k=1}^{N_{\text{nucleotides}}} \log p_{N_1^k N_2^k}(t). \tag{5}$$

Note that throughout the paper we use natural logarithms.

2.1.2 Sequences of codons

In the case of coding sequences, the above models have an obvious short-coming: they take no account of the selection pressure acting on the encoded amino acids that will, for example, tend to make synonymous mutations relatively more frequently observed than nonsynonymous mutations. In this case one may use an analogous codon-based model utilizing a 64×64 matrix $\mathbf{Q}$. Details of one such model are given in Goldman & Yang (1994), who use a matrix incorporating equilibrium codon frequencies, the transition:transversion ratio κ, an amino acid distance scale (based on the Grantham 1974 chemical properties distances) and a distance scaling parameter V to allow for some genes being under greater selection pressure than others. Other models (Yang *et al.* 1998) use amino acid distances based on the empirical PAM or BLOSUM matrices, or even set the distances as free parameters. Despite the increased complexity, the probability that one sequence mutates to another after time t may still be expressed in the simple form

$$\log P(S_1 \to S_2; t) = \sum_{k=1}^{N_{\rm codons}} \log p_{C_1^k C_2^k}(t), \tag{6}$$

where C_1^k and C_2^k are now the kth codons of sequences S_1 and S_2.

2.1.3 Overlapping sequences of codons

The above framework breaks down for overlapping CDSs. Consider, for example, the sequence GGUA..., reading *gly...* in the primary read-frame and *val...* in the +1 secondary read-frame. Suppose we wanted to produce a 64×64 codon mutation matrix for codons in the primary read-frame. Then a reasonable way to proceed would be to weight codon mutation probabilities by the 'acceptability' of the amino acid substitutions in both the primary and secondary read-frames. E.g. for the above sequence, $P(\text{GGU} \to \text{GAU}) = \mathbf{P}(\text{G} \to \text{A}) \times \mathbf{A}(gly \to asp) \times \mathbf{A}(val \to ile)$, where $\mathbf{P}$ is the nucleotide mutation matrix and $\mathbf{A}$ is an amino acid substitution matrix. The problem is that the amino acid substitution in the secondary read-frame depends, in this case, on the 4th nucleotide position. In general, except for the -3 frame, the probabilities in the 64×64 matrix $\mathbf{Q}$ depend on three neighbouring nucleotide positions. Just allowing $\mathbf{Q}$ to be positionally-dependent does not help since the nucleotides at these neighbouring positions may vary over the course of evolution, so $\mathbf{Q}$ needs to be time-dependent also. In principle this cross-talk between adjacent codon positions can propagate along the entire length of the sequence so that the mutation pathway of any codon in the sequence is dependent on the mutation pathways of every other codon. Hence it is impossible to calculate $\mathbf{P}(t) = \exp(\mathbf{Q}t)$ in a way that allows the factorization-by-codon seen in equation 6.

In theory, we must define a single matrix that describes the entire sequence at once. For example, for a 60-nucleotide overlap region, we must define a $4^{60} \times 4^{60}$ $\mathbf{Q}$ matrix. This is clearly computationally impractical. Pedersen & Jensen (2001) describe one computationally-intensive way to proceed via simulations. In our paper we develop a simplified, but very quick, model (viz. MLOGD).

2.2 Monte Carlo sequence evolution model

The problem outlined in §2.1.3 does not arise when generating simulated sequences. Random nucleotide mutations may be applied sequentially and the neighbourhood of any mutating nucleotide is known at the time of mutation, thus allowing correct amino acid substitution acceptability weights to be applied. We therefore use this approach in our simulations.

Given a starting sequence, a frame, a mutation rate or time t, a 4×4 instantaneous nucleotide mutation matrix $\mathbf{Q}$, a 64-entry codon usage table $\mathbf{C}$, and a 20×20 amino acid substitution matrix $\mathbf{A}$, (see §2.3 for details of $\mathbf{Q}$, $\mathbf{C}$, $\mathbf{A}$), our Monte Carlo simulation applies nucleotide mutations one-by-one until the required total number of mutations have occurred. The mutation rate is tied to the number of observed mutations rather than the number of accepted mutations (e.g. A $\to$ C $\to$ G counts as two accepted mutations but only one observed mutation) since, for real sequence data, only the number of observed mutations is known. We refer to the observed mutation rate as λ. For equal π_j and no codon or amino acid constraints, $0 \le \lambda \le 0.75$ and $\lambda \to 0.75$ as $t \to \infty$.

Initially, the position and type of each nucleotide mutation is chosen randomly, but with the probabilities weighted using the off-diagonal terms of $\mathbf{Q}$ so that, for example, transitions are more common than transversions. Further weighting is applied to maintain codon usage bias. Specifically, $\mathbf{C}$ is scaled so that its maximum value is unity, and each nucleotide mutation is retained with probability equal to the value for the corresponding new codon. The rows of $\mathbf{A} = [a_{ij}]$ are similarly scaled so that diagonal terms are unity and a nucleotide mutation causing an amino acid substitution $X_i \to X_j$ is retained with probability

a_{0j}, where X_0 is the *original* amino acid at that site (see below). If the sequence is double-coding, the codon usage and amino acid weights are also applied in the secondary read-frame.

We have diverged from the standard Markov model of sequence evolution in one respect, viz. the acceptability of any amino acid substitution is calculated relative to the *original* amino acid at that sequence position rather than the amino acid at that sequence position immediately prior to the substitution. E.g. ignoring the $\mathbf{Q}$ and $\mathbf{C}$ contributions,

$$P(X_0 \to X_j \to X_k) = P(X_0 \to X_j) \times P(X_0 \to X_k) \tag{7}$$

rather than

$$P(X_0 \to X_j \to X_k) = P(X_0 \to X_j) \times P(X_j \to X_k). \tag{8}$$

The rationale for this is that we wish to test whether X_j and X_k are acceptable substitutions at a site for which only X_0 is *known* to be acceptable. Indeed if the latter scheme (equation 8) is used, then the sequence will tend to a random sequence of amino acids, bearing no relation to the original sequence, as $t \to \infty$.

2.3 Input parameters

In this section we describe the input 4×4 nucleotide mutation matrix $\mathbf{Q}$, 64-entry codon usage table $\mathbf{C}$, and 20×20 amino acid substitution matrix $\mathbf{A}$, used in the simulations (§2.2) and maximum likelihood overlapping CDS detection algorithm (§2.4.3).

2.3.1 Nucleotide substitution matrix

In the paper we use a $\kappa = 3$ Kimura (1980) nucleotide matrix, i.e.

$$\mathbf{Q} = \begin{bmatrix} -1.0 & 0.6 & 0.2 & 0.2 \\ 0.6 & -1.0 & 0.2 & 0.2 \\ 0.2 & 0.2 & -1.0 & 0.6 \\ 0.2 & 0.2 & 0.6 & -1.0 \end{bmatrix}. \tag{9}$$

This corresponds to equal equilibrium nucleotide frequencies π_j and a transition:transversion ratio of $\kappa = 3$.

The value $\kappa = 3$ is similar to that for a human–mouse comparison (Chiaromonte *et al.* 2002). Nucleotide substitution matrices are often calculated from sequence alignments by studying substitutions at neutral sites, such as non-coding regions or four-fold degenerate third codon positions. However, due to the large number of overlapping coding and non-coding features, it is often not straightforward to identify neutral sites in compact viral genomes.

In the Monte Carlo simulations, $\mathbf{Q}$ is used to establish the relative frequencies of the 12 possible non-null nucleotide mutations. In the maximum likelihood overlapping CDS detection algorithm, $\mathbf{P}(t) = \exp(\mathbf{Q}t)$ is used to calculate the probability of an observed null (e.g. A $\to$ A) or non-null (e.g. A $\to$ C) nucleotide mutation in a pairwise sequence alignment, as a function of time (i.e. sequence divergence) t.

2.3.2 Codon usage table

In the paper we use a null codon usage table (CUT) – i.e. equal codon frequencies. Due to the large number of overlapping coding and non-coding features in compact viral genomes, it is not clear that a CUT generated directly from the genome of such a virus will be representative of mutation probabilities. A host species (e.g. human) CUT may be more appropriate. However the human CUT is not strongly biased and so we use a null CUT for simplicity. More generally a non-null CUT may be incorporated as follows.

Suppose we have the codon GGU. Mutations in the 3rd position are synonymous (all code for *gly*) and their relative frequencies are controlled by the nucleotide mutation matrix $\mathbf{Q}$. However we also wish to preserve codon bias as a sequence mutates. Since we are always working from an initial known amino acid, we must use relative (instead of absolute) codon frequencies but each frequency must be multiplied by the degeneracy of the corresponding amino acid, otherwise, for example, ?UG $\to$ CUG (*leu*) will be a factor of six less probable than ?UG $\to$ AUG (*met*) simply because there are six codons for *leu* but only one for *met*. In addition, codon usage statistics implicitly include any nucleotide bias and, conversely, any nucleotide bias described by $\mathbf{Q}$ will automatically lead to a codon bias. Hence the nucleotide equilibrium frequencies π_j must be factored out by dividing each codon usage value by $\pi_i.\pi_j.\pi_k$ where i, j, k are the ith, jth, kth nucleotides in the codon.

Note that no allowance is made for the hypermutability of GC dinucleotides, which can affect the codon and nucleotide transition probabilities (cf. Pedersen *et al.* 1998).

2.3.3 Amino acid substitution matrix

In our model, the probability that a nucleotide mutation occurs at the DNA level and the probability that the mutation is accepted (i.e. is functional) at the protein level are separated into the nucleotide and amino acid matrices. In contrast, the widely used BLOSUM (Henikoff & Henikoff 1992) and PAM (Dayhoff *et al.* 1978) matrices incorporate both effects into one matrix. In the PAM matrices, the small-t amino acid substitution frequencies are extrapolated to larger t. This is a serious short-coming since, in reality, at small t a mutating sequence is constrained to resemble the original sequence at both the nucleotide and amino acid levels, whereas at large t a mutating sequence is only constrained to resemble the original at the amino acid level. On the other hand, the BLOSUM matrices are calculated, in effect, for a series of t values: BLOSUM100, BLOSUM95, ... BLOSUM35, with the lower indices corresponding to more divergent sequences. By choosing a low-index BLOSUM matrix (viz. BLOSUM40) as our default amino acid distance matrix $\mathbf{A}$, we minimize the effect of the nucleotide mutation constraint relative to the amino acid acceptability constraint.

We use the scaled observed frequencies (Henikoff & Henikoff $\frac{q_{ij}}{e_{ij}}$ values) rather than log odds scores, and treat $\frac{q_{ij}}{e_{ij}}/\frac{q_{ii}}{e_{ii}}$ as the probability of acceptance for the amino acid substitution $X_i \to X_j$ relative to $X_i \to X_i$ which is unity. Stop codons are also included, with the acceptabilities for mutations between stops and non-stops set to zero.

An alternative to the BLOSUM matrices would be to use a chemical properties distance matrix (Grantham 1974), but there is no clear way to convert such distances into acceptance probabilities.

2.4 Double-coding test statistics

We investigate three test statistics designed to detect the mutation signature of overlapping CDSs in pairwise alignments of two sequences S_1 and S_2.

2.4.1 N123: Mutation rate in N_1, N_2, N_3

For the N123 method we simply count the number of nucleotide differences between S_1 and S_2 in each of N_1, N_2, N_3 (i.e. the 1st, 2nd, 3rd nucleotide positions in codons in the primary read-frame) and express each count as a fraction of the total number of N_1, N_2, N_3 loci. We label these statistics f_{N1}, f_{N2}, f_{N3} respectively.

2.4.2 NsNn: Synonymous and nonsynonymous mutation rates

For the NsNn method we step through aligned codon pairs in the primary read-frame of S_1 and S_2 and count which codon pairs are identical, non-identical but synonymous, and nonsynonymous. The numbers of synonymous and nonsynonymous codon pairs are expressed as a fraction of the total number of codon pairs. We label these statistics f_{syn}, f_{non} respectively.

2.4.3 MLOGD: Maximum likelihood method

For the MLOGD (Maximum Likelihood Overlapping Gene Detector) method we step through aligned nucleotide pairs N_1^k, N_2^k in S_1, S_2 respectively. For each nucleotide pair we estimate the probability that N_1^k mutates to N_2^k under each of a single-coding model and a double-coding model. Specifically, we first define $b(N_1^k \to i; t, m)$, $i = \mathrm{U, C, A, G}$, by

$$\begin{aligned} b(N_1^k \to i; t, \mathrm{s}) \quad = \quad & \mathbf{P}(N_1^k \to i; t) \times \mathbf{C}(X_2) \\ & \times \mathbf{A}(X_1 \to X_2) \end{aligned} \tag{10}$$

for the single-coding ($m =$'s') model, and

$$\begin{aligned} b(N_1^k \to i; t, \mathrm{d}) \quad = \quad & \mathbf{P}(N_1^k \to i; t) \times \mathbf{C}(X_2) \\ & \times \mathbf{A}(X_1 \to X_2) \times \mathbf{C}(X_2') \\ & \times \mathbf{A}(X_1' \to X_2') \end{aligned} \tag{11}$$

for the double-coding (m = 'd') model. Here X_1 and X_2 are the original and final amino acids or codons in the primary read-frame and X_1' and X_2' are the original and final amino acids or codons in the secondary read-frame for the nucleotide mutation $N_1^k \to i$. Also $\mathbf{P}(t) = \exp(\mathbf{Q}t)$, and $\mathbf{Q}$, $\mathbf{C}$, $\mathbf{A}$ are the nucleotide, codon and amino acid matrices described in §2.3.

The probability that N_1^k mutates to N_2^k, after time t, is then given by

$$P(N_1^k \to N_2^k; t, m) = \frac{b(N_1^k \to N_2^k; t, m)}{\sum_{i=\mathrm{U,C,A,G}} b(N_1^k \to i; t, m)}. \tag{12}$$

As in §2.1.1, we sum over the sequence alignment as follows

$$\log P(S_1 \to S_2; t, m) = \sum_{k=1}^{N_{\text{nucleotides}}} \log P(N_1^k \to N_2^k; t, m). \tag{13}$$

We maximize equation 13 with respect to t for each of the single-coding and double-coding models, giving t^{s} and t^{d} respectively. Then the log likelihood ratio of the two models is

$$\Delta_{\mathrm{s,d}} = \log P(S_1 \to S_2; t^{\mathrm{d}}, \mathrm{d}) - \log P(S_1 \to S_2; t^{\mathrm{s}}, \mathrm{s}). \tag{14}$$

If $\Delta_{\mathrm{s,d}}$ is positive, then the observed mutations between S_1 and S_2 are more consistent with a double-coding model. If $\Delta_{\mathrm{s,d}}$ is negative, then the observed mutations are more consistent with a single-coding model. Due to non-independence of mutations in adjacent nucleotides, $\Delta_{\mathrm{s,d}}$ is not strictly speaking a likelihood ratio, but may still be used as a classifier.

Note that for the reasons described in §2.1.3 the above methodology involves several approximations. For each component of the probability sum (equation 13), we consider only a single nucleotide and the single codon in each of the primary and secondary read-frames containing that nucleotide. In addition, as far as codon and amino acid weightings are concerned, we consider only the start and end points of the unknown mutation pathway connecting an aligned codon pair in S_1 and S_2. It seems reasonable that these simplifications are justified provided S_1 and S_2 are not too divergent (so that mutation pathways are short and inter-codon cross-talk is low). Tests with Monte Carlo simulations (which are not subject to these simplifications) show that the model provides useful results over a wide range of circumstances (details in the paper). Furthermore, we ignore the fact that different genes may be under differing selection pressure. This effect could be incorporated with extra parameters V_1 and V_2 scaling the amino acid substitution acceptabilities in the primary and secondary ORFs respectively. We have chosen not to include such parameters because (a) selection pressure is expected to vary from one codon to another rather than just from one gene to another, (b) the BLOSUM amino acid matrices represent *average* amino acid substitution acceptabilities over all codons, and (c) we wish to keep the number of free parameters in our model small. In any case, our fundamental goal is not to create an accurate model of sequence evolution (cf. Pedersen & Jensen 2001), but rather to develop a convenient tool for distinguishing double-coding sequences from single-coding sequences.

2.5 Classification

Having calculated the statistics $\Delta_{\mathrm{s,d}}$, f_{N1}, f_{N2}, f_{N3}, f_{syn}, f_{non} for a particular pairwise sequence alignment $S_1 + S_2$, as described in §2.4, we then wish to classify it as either single-coding or double-coding according to which of the two models is most consistent with the observed statistics. For the MLOGD method, classification is straightforward: if $\Delta_{\mathrm{s,d}} > 0$ we classify $S_1 + S_2$ as double-coding, while if $\Delta_{\mathrm{s,d}} \leq 0$ we classify $S_1 + S_2$ as single-coding. For the N123 and NsNn methods we resort to simulations to find out what range of values we would expect to observe for the two models. We may then find the log likelihood ratios

$$\Delta_{\mathrm{s,d}}^{\mathrm{N123}} = \log\left[\frac{P(f_{\mathrm{N1}}, f_{\mathrm{N2}}, f_{\mathrm{N3}}|\mathrm{d}, \lambda)}{P(f_{\mathrm{N1}}, f_{\mathrm{N2}}, f_{\mathrm{N3}}|\mathrm{s}, \lambda)}\right] \tag{15}$$

and

$$\Delta_{\mathrm{s,d}}^{\mathrm{NsNn}} = \log\left[\frac{P(f_{\mathrm{syn}}, f_{\mathrm{non}}|\mathrm{d}, \lambda)}{P(f_{\mathrm{syn}}, f_{\mathrm{non}}|\mathrm{s}, \lambda)}\right]. \tag{16}$$

A similar statistic may be defined for $\Delta_{\mathrm{s,d}}$, viz.

$$\Delta_{\mathrm{s,d}}^{\mathrm{MLOGD}} = \log\left[\frac{P(\Delta_{\mathrm{s,d}}|\mathrm{d}, \lambda)}{P(\Delta_{\mathrm{s,d}}|\mathrm{s}, \lambda)}\right]. \tag{17}$$

Ideally one would find the expected probability distributions of $\Delta_{\mathrm{s,d}}$, f_{N1}, f_{N2}, f_{N3}, f_{syn}, f_{non} directly from the simulations. In practice it would require a very large number of simulations to estimate the near-zero probabilities that occur whenever the observed statistics deviate from either of the models. Instead we assume a probability distribution (e.g. normal) and use the simulations to estimate the distribution parameters (e.g. μ, σ).

We generate simulated sequences (§2.2), starting with the sequence S_1 and with the mutation rate fixed by the total number of point differences between S_1 and S_2. We generate 100 simulated sequences assuming a single-coding model and 100 simulated sequences assuming a double-coding model. We then calculate $\Delta_{\mathrm{s,d}}$, f_{N1}, f_{N2}, f_{syn}, f_{non} for each of the 200 sequence pairs comprising S_1 and one of the simulated sequences, and calculate the mean and standard deviation for each statistic for each of the two models, giving μ_x^m, σ_x^m where m = 's' or 'd' is the model, and x is one of the statistics $\Delta_{\mathrm{s,d}}$, f_{N1}, f_{N2}, f_{syn}, f_{non}. Note that we omit f_{N3} since, having fixed $\lambda = f_{\mathrm{N1}} + f_{\mathrm{N2}} + f_{\mathrm{N3}}$, f_{N3} is redundant. Assuming normal distributions, and writing $z_x^m = \frac{x-\mu_x^m}{\sigma_x^m}$,

$$P(x|m,\lambda) = \frac{1}{\sqrt{2\pi}\sigma_x^m} \exp\left[-\frac{1}{2}(z_x^m)^2\right] \tag{18}$$

giving the log likelihood ratio

$$\log\left[\frac{P(x|\mathrm{d},\lambda)}{P(x|\mathrm{s},\lambda)}\right] = \log\left(\frac{\sigma_x^{\mathrm{s}}}{\sigma_x^{\mathrm{d}}}\right) + \frac{1}{2}\left((z_x^{\mathrm{s}})^2 - (z_x^{\mathrm{d}})^2\right). \tag{19}$$

Assuming independence of f_{N1}, f_{N2} and of f_{syn}, f_{non}, we may combine these as follows

$$\Delta_{\mathrm{s,d}}^{\mathrm{N123}} = \log\left[\frac{P(f_{\mathrm{N1}}|\mathrm{d},\lambda)}{P(f_{\mathrm{N1}}|\mathrm{s},\lambda)}\right] + \log\left[\frac{P(f_{\mathrm{N2}}|\mathrm{d},\lambda)}{P(f_{\mathrm{N2}}|\mathrm{s},\lambda)}\right] \tag{20}$$

and

$$\Delta_{\mathrm{s,d}}^{\mathrm{NsNn}} = \log\left[\frac{P(f_{\mathrm{syn}}|\mathrm{d},\lambda)}{P(f_{\mathrm{syn}}|\mathrm{s},\lambda)}\right] + \log\left[\frac{P(f_{\mathrm{non}}|\mathrm{d},\lambda)}{P(f_{\mathrm{non}}|\mathrm{s},\lambda)}\right]. \tag{21}$$

The sequence pair $S_1 + S_2$ is then classified as single-coding if $\Delta_{\mathrm{s,d}}^{\mathrm{MLOGD}} \leq 0$, $\Delta_{\mathrm{s,d}}^{\mathrm{N123}} \leq 0$ or $\Delta_{\mathrm{s,d}}^{\mathrm{NsNn}} \leq 0$ and as double-coding if $\Delta_{\mathrm{s,d}}^{\mathrm{MLOGD}} > 0$, $\Delta_{\mathrm{s,d}}^{\mathrm{N123}} > 0$ or $\Delta_{\mathrm{s,d}}^{\mathrm{NsNn}} > 0$.

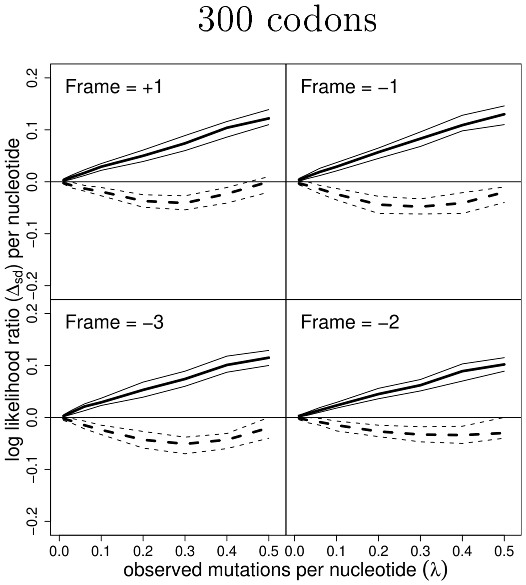
This classification scheme is simple to implement but ignores such things as non-independence of f_{N1}, f_{N2} and of f_{syn}, f_{non}, and deviations from normal distributions (e.g. f_{N1}, f_{N2}, f_{syn}, f_{non} are constrained to be non-negative), so $\Delta_{\mathrm{s,d}}^{\mathrm{MLOGD}}$, $\Delta_{\mathrm{s,d}}^{\mathrm{N123}}$ and $\Delta_{\mathrm{s,d}}^{\mathrm{NsNn}}$ are not strictly speaking likelihood ratios, but may still be used as classifiers. In some situations it may be preferable to use a more sophisticated classification method, such as a neural network. The statistics $\Delta_{\mathrm{s,d}}$, $\Delta_{\mathrm{s,d}}^{\mathrm{MLOGD}}$, $\Delta_{\mathrm{s,d}}^{\mathrm{N123}}$, $\Delta_{\mathrm{s,d}}^{\mathrm{NsNn}}$ may also be summed along the branches of a phylogenetic tree – for increased sensitivity – by calculating (for example) maximum parsimony sequences at intermediate nodes.

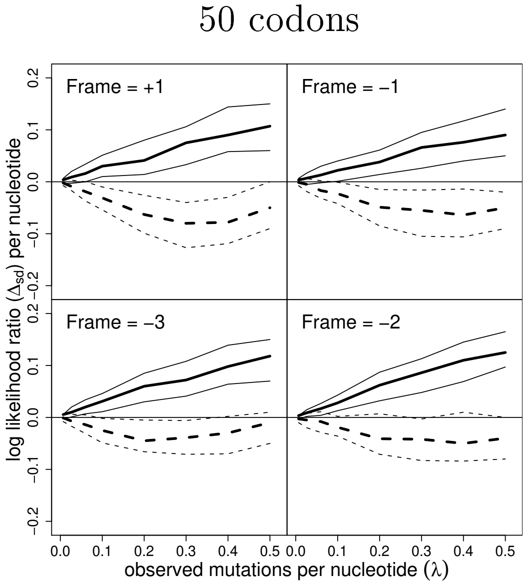
For reasons mentioned above, none of $\Delta^x = \Delta_{\mathrm{s,d}}$, $\Delta_{\mathrm{s,d}}^{\mathrm{MLOGD}}$, $\Delta_{\mathrm{s,d}}^{\mathrm{N123}}$ or $\Delta_{\mathrm{s,d}}^{\mathrm{NsNn}}$ are exactly likelihood ratios. If they were, then the relative probability of a given sequence being double-coding or single-coding is given by Bayesian statistics:

$$\begin{aligned} \frac{P(\mathrm{d}|\mathrm{S}_1+\mathrm{S}_2)}{P(\mathrm{s}|\mathrm{S}_1+\mathrm{S}_2)} &= \frac{P(\mathrm{d})}{P(\mathrm{s})}\frac{P(\mathrm{S}_1+\mathrm{S}_2|\mathrm{d})}{P(\mathrm{S}_1+\mathrm{S}_2|\mathrm{s})} \\ &= \frac{P(\mathrm{d})}{P(\mathrm{s})}\exp(\Delta^x) \end{aligned} \tag{22}$$

where $P(\mathrm{d})$ and $P(\mathrm{s})$ are prior probabilities for double-coding and single-coding – i.e. they represent the fraction of all coding nucleotides (in the genome) that are expected to be involved in double-coding or single-coding respectively.

3 MLOGD, N123, NsNn plots for simulated sequences





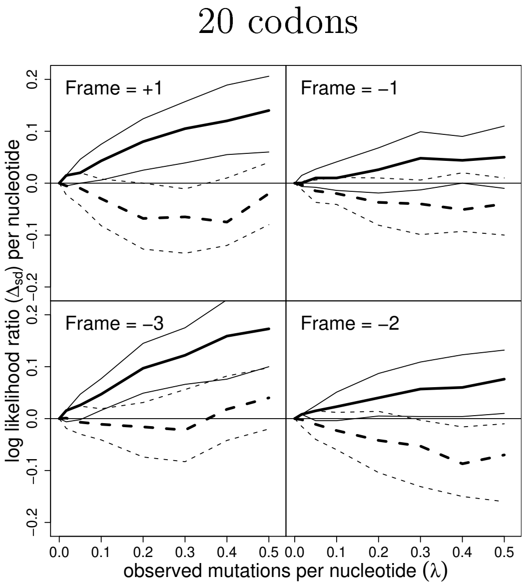
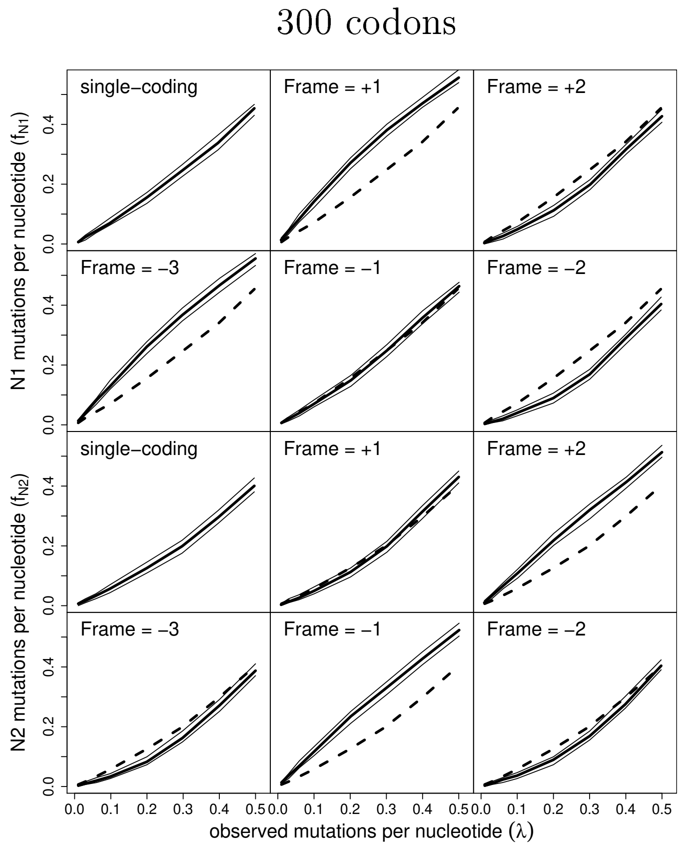
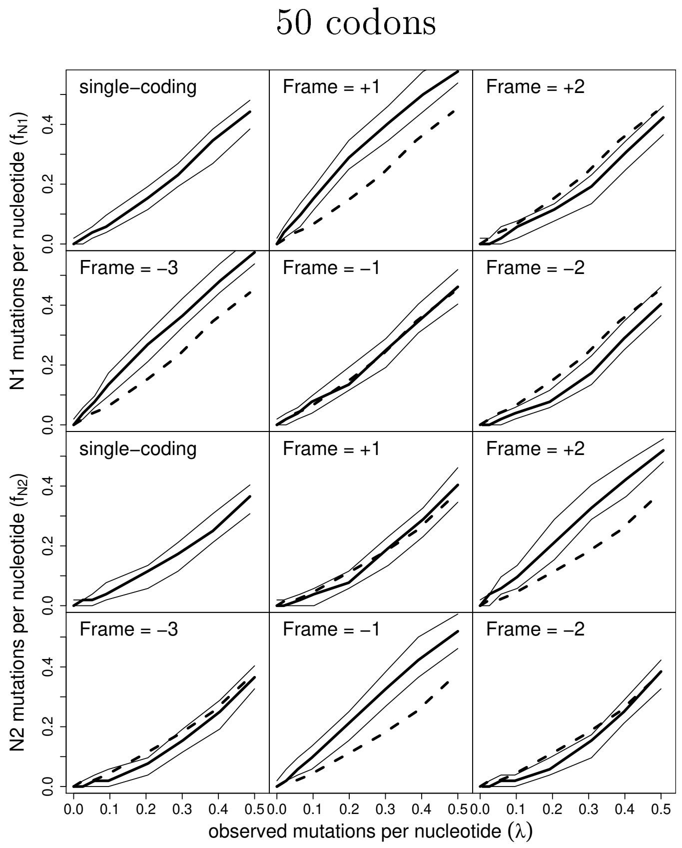


Figure 2: Loci of the MLOGD statistic $\Delta_{\mathrm{s,d}}$ (per nucleotide) for simulated single-coding (dashed lines) and double-coding (solid lines) sequences. The thick lines represents the median values while the thin lines enclose the central 68% of values. Sequence divergence is measured on the x-axis. The four panels show the locus of $\Delta_{\mathrm{s,d}}$ for single-coding sequences and four of the five possible read-frames of an overlapping secondary ORF relative to a primary ORF (the +2 frame is equivalent to the +1 frame). Note that the above graphs (for each sequence length) are for a particular starting sequence. Using a different starting sequence (different sequence composition) can lead to different, albeit similar, graphs. $\Delta_{\mathrm{s,d}} < 0 \Rightarrow$ single-coding while $\Delta_{\mathrm{s,d}} > 0 \Rightarrow$ double-coding. In contrast to Figures 3 and 4, where the single-coding loci of the f_{N1}, f_{N2}, f_{syn}, f_{non} statistics can be displayed in a separate panel, the locus of $\Delta_{\mathrm{s,d}}$ for single-coding sequences depends on which potential secondary read-frame the single-coding model is being tested against.





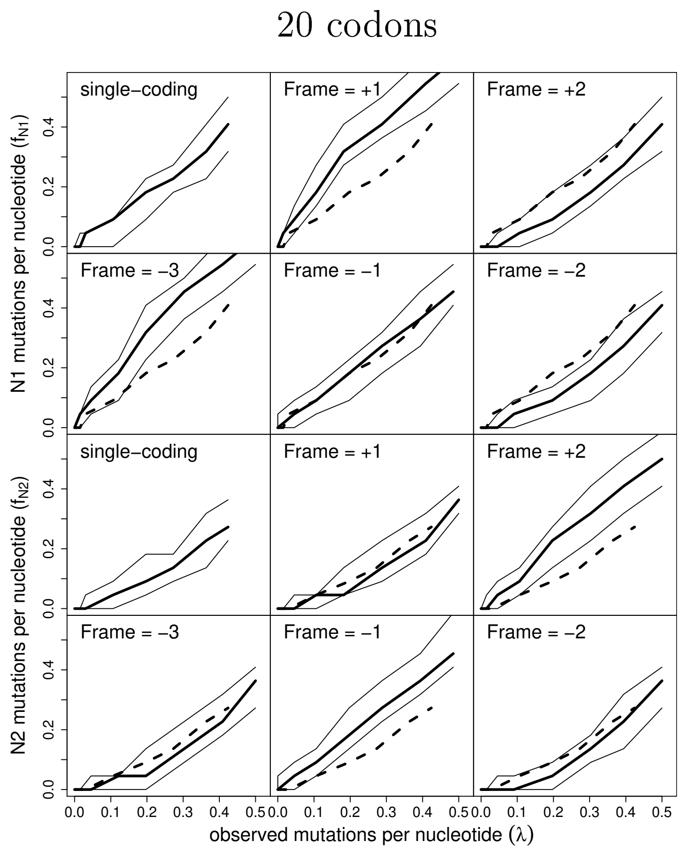
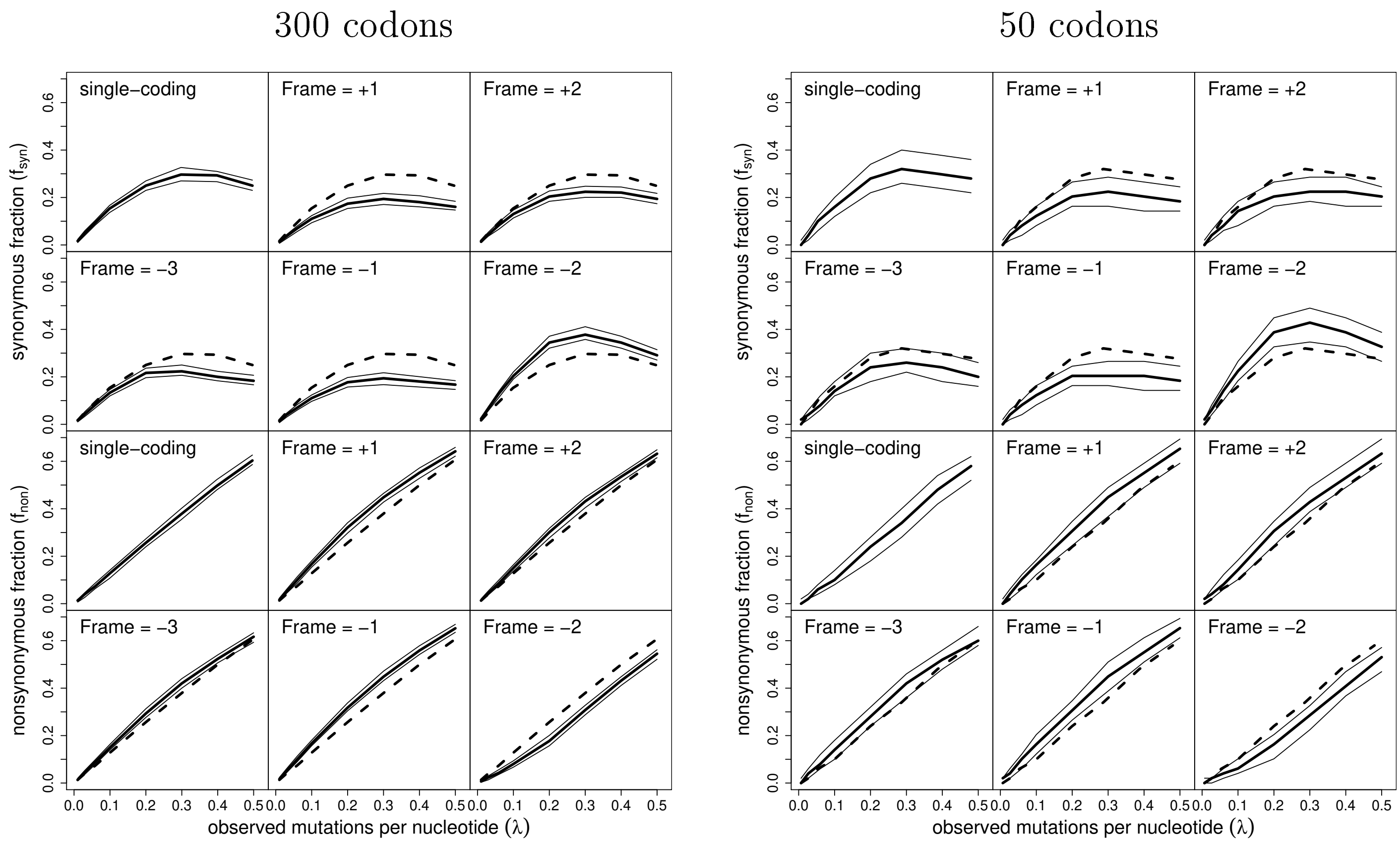


Figure 3: Loci of the 1st and 2nd codon position mutation rates, f_{N1} and f_{N2}, as a function of mean mutation rate λ, for simulated sequences. The thick lines represents the median values while the thin lines enclose the central 68% of values. Sequence divergence is measured on the x-axis. The twelve panels show the loci of f_{N1} and f_{N2} for single-coding sequences and the five possible read-frames of an overlapping secondary ORF relative to a primary ORF (solid lines). The dashed line – marking the median locus for single-coding sequences – is included in each plot for reference. Note that the above graphs (for each sequence length) are for a particular starting sequence. Using a different starting sequence (different sequence composition) can lead to different, albeit similar, graphs.



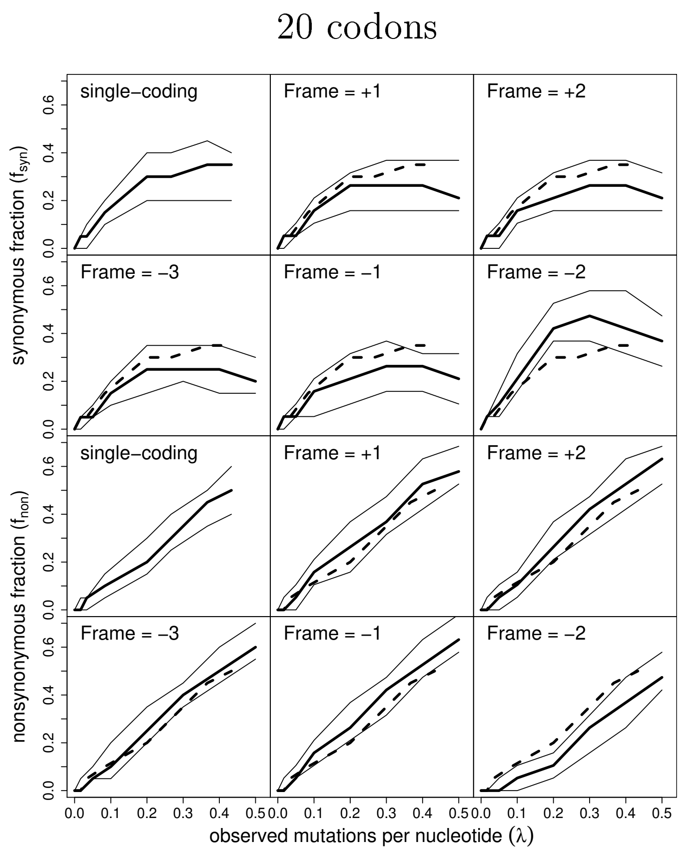
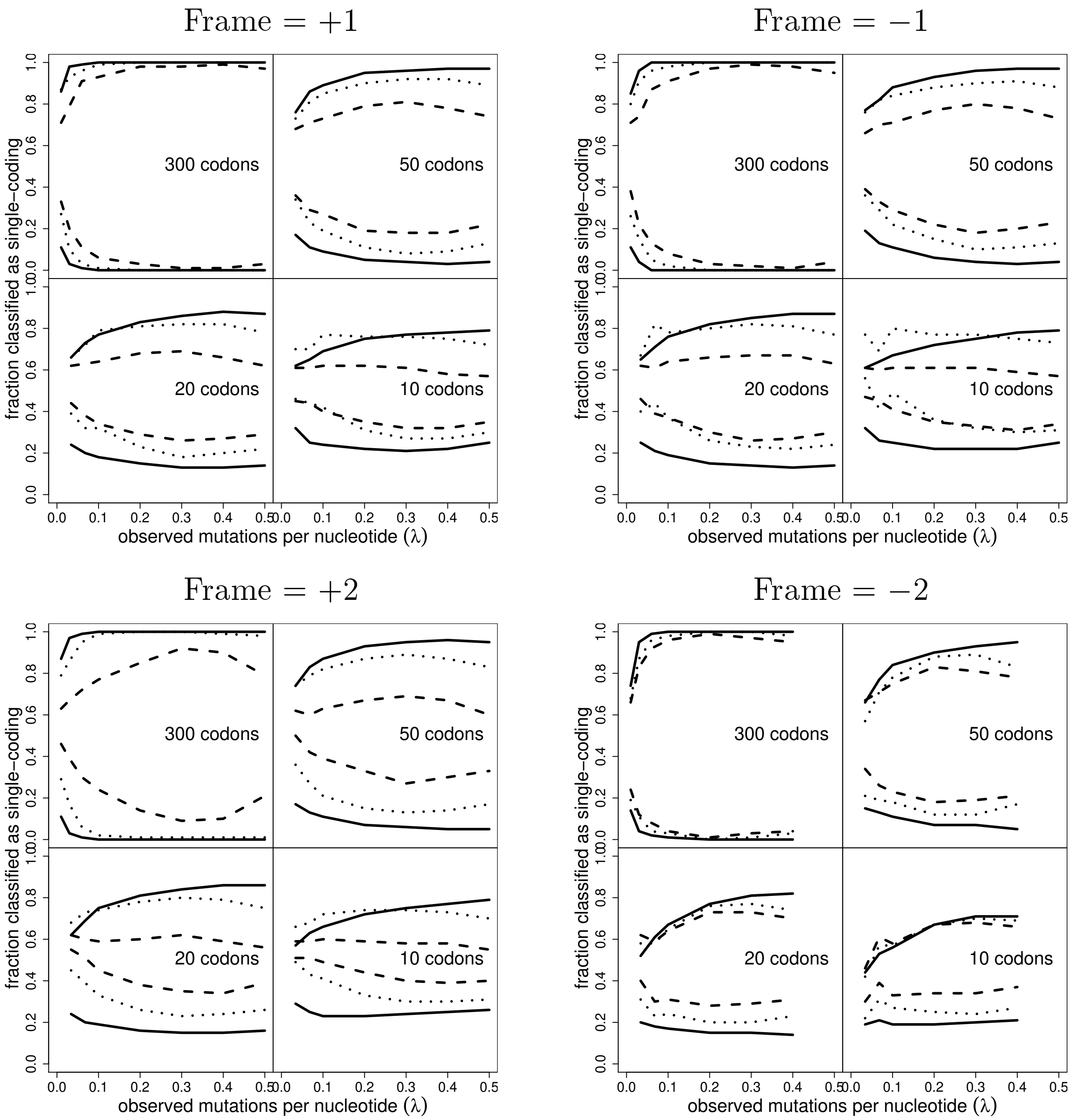


Figure 4: Loci of the synonymous and nonsynonymous mutation rates, $f_{\rm syn}$ and $f_{\rm non}$, as a function of mean mutation rate λ, for simulated sequences. The thick lines represents the median values while the thin lines enclose the central 68% of values. Sequence divergence is measured on the x-axis. The twelve panels show the loci of $f_{\rm syn}$ and $f_{\rm non}$ for single-coding sequences and the five possible read-frames of an overlapping secondary ORF relative to a primary ORF (solid lines). The dashed line – marking the median locus for single-coding sequences – is included in each plot for reference. Note that the above graphs (for each sequence length) are for a particular starting sequence. Using a different starting sequence (different sequence composition) can lead to different, albeit similar, graphs.



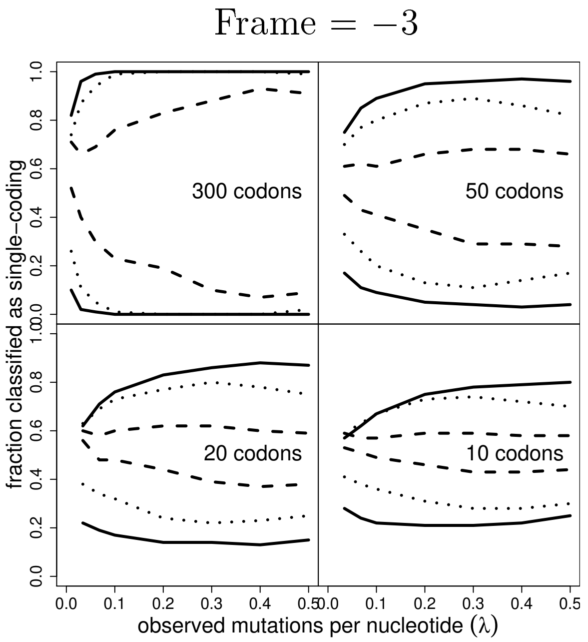
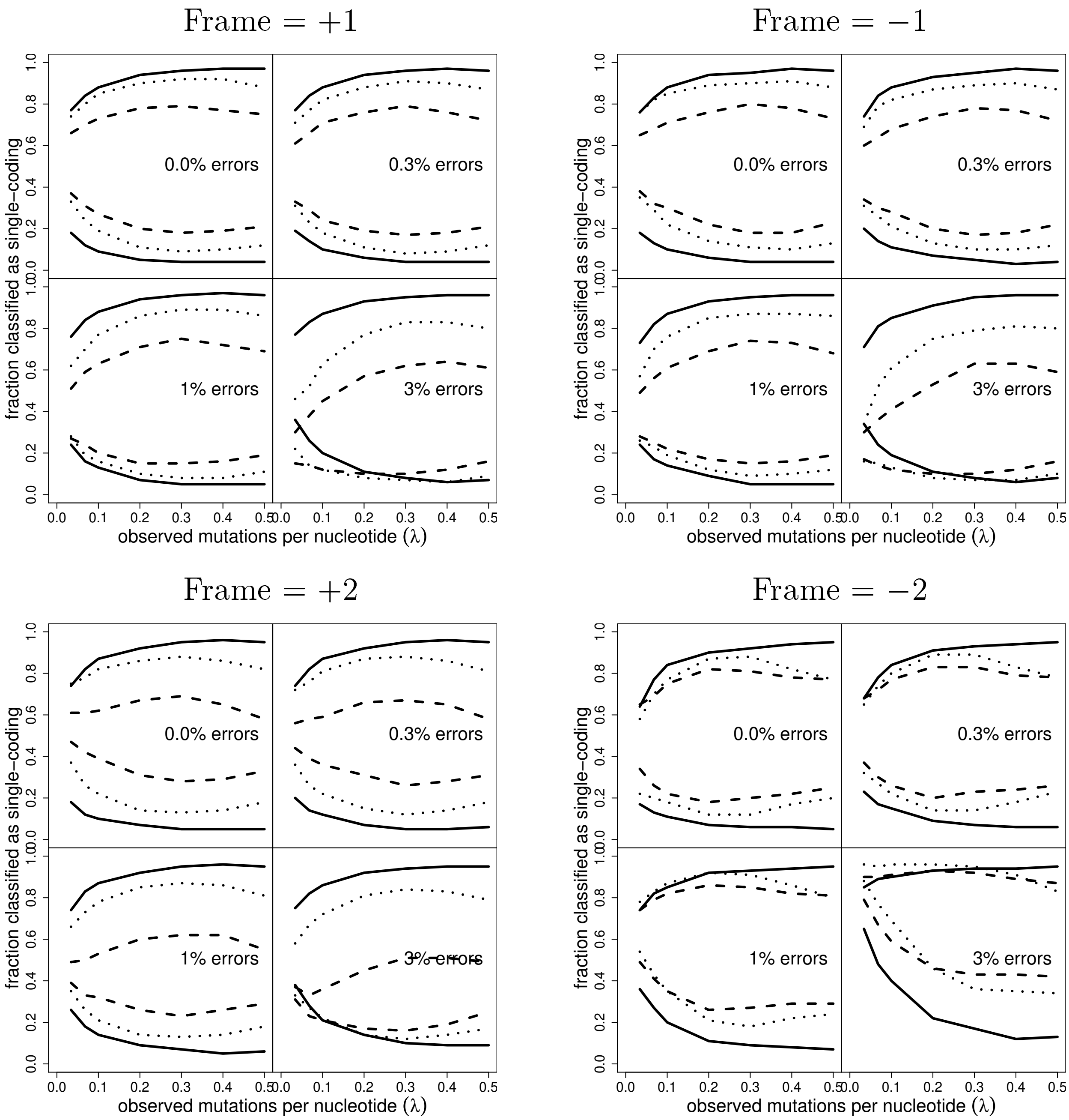


Figure 5: Classification success as a function of mutation rate λ and sequence length, for different double-coding frames. In each panel the upper three lines show the fraction of single-coding simulations classified as single-coding ($1 \equiv$ perfect classification) while the lower three lines show the fraction of double-coding simulations classified as single-coding ($0 \equiv$ perfect classification). The classifications use the MLOGD (solid lines), NsNn (dashed lines), and N123 (dotted lines) methods. Plotted points are averages for $\sim \frac{2000}{N_{\text{codons}}}$ random initial sequences. The rms error is of the order 0.01-0.04 (depending on λ, sequence length etc.). Note that the plot is terminated at $\lambda = 0.4$ in the -2 frame because the strong constraints on N_1, N_2 mutations in this frame inhibit high sequence divergences in the simulations.



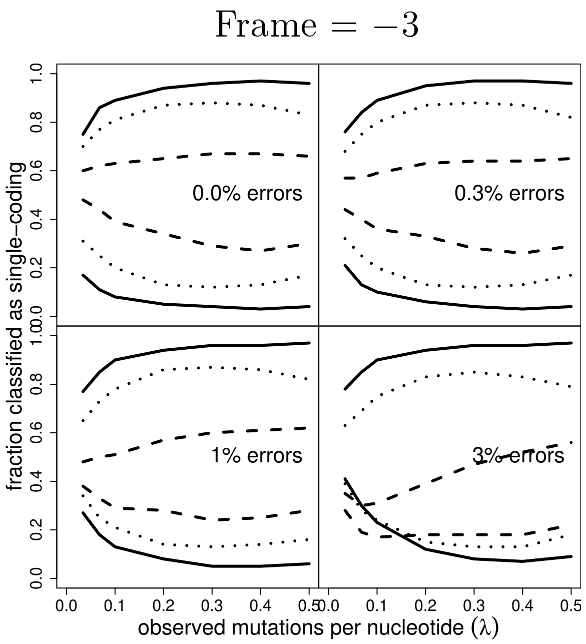
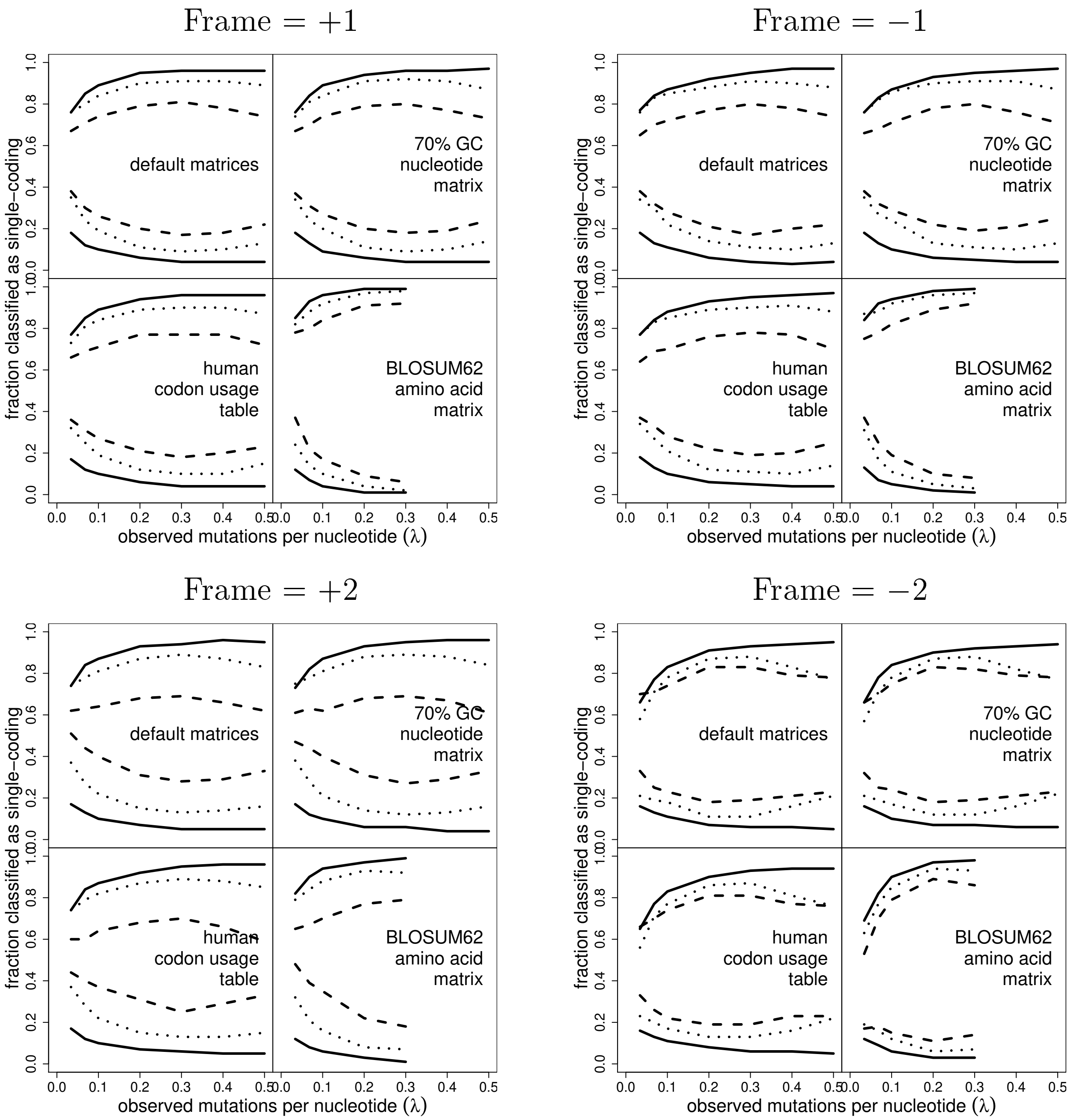


Figure 6: Reduction in classification success as a result of sequencing errors. The error rates per nucleotide are indicated in each panel. Sequence length = 50 codons. See caption to Figure 5 for further details.



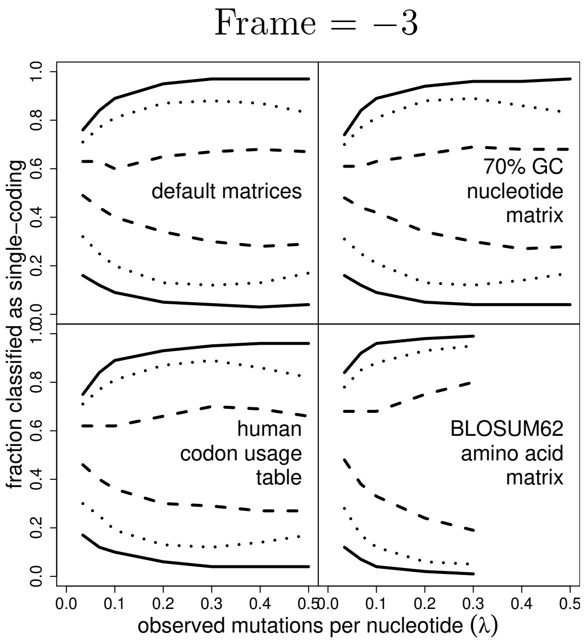


Figure 7: Reduction in classification success as a result of incorrect model nucleotide, codon or amino acid matrices. The default matrices ($\kappa = 3$ Kimura nucleotide matrix, null CUT – i.e. equal codon frequencies, and BLOSUM40 amino acid matrix) are used for classification and the different matrices used to generate the simulated sequences to be classified are indicated in each panel. Sequence length = 50 codons. See caption to Figure 5 for further details. Note that the BLOSUM62 plots are terminated at $\lambda = 0.3$ because the stronger constraints on mutations represented by the BLOSUM62 matrix relative to the BLOSUM40 matrix inhibit high sequence divergences in the simulations.

4 Triple coding

Overlapping coding sequences are common in compact viral genomes. When testing for new overlapping CDSs, one may find oneself testing an ORF that overlaps two known CDSs (which if really coding would be an example of triple coding). This is difficult using the N123 method because of the extra frame-dependent constraint on N1, N2, N3. MLOGD is fairly robust with respect to this problem, as we show below. In principle, it is straightforward to add a triple-coding option to our Monte Carlo simulations, however we decided not to do this because it would not be of very general use.

To assess the effect of two overlapping real coding sequences on the MLOGD signal in a third overlapping – but non-coding – sequence, double-coding sequences were simulated for each of the five possible frames of a secondary coding sequence relative to a primary coding sequence. Then the MLOGD $\Delta_{\mathrm{s,d}}$ score was evaluated for each of the five possible frames of a tertiary ORF relative to the primary sequence. The distribution of these scores, as a function of sequence divergence λ, are illustrated in Figure 8. In addition, Figure 9 shows the number of stop codons introduced into the tertiary ORF. A random string of 900 nucleotides was used as the initial sequence and, for each frame combination, 200 sequences were simulated with λ randomly chosen from the range (0,0.5). A null codon usage table, $\kappa = 3$ Kimura nucleotide matrix, and BLOSUM40 amino acid matrix were used for the simulations and to calculate $\Delta_{\mathrm{s,d}}$.

Most frame combinations give a negative MLOGD signal in the tertiary frame – indicating, as desired, that the tertiary frame is non-coding. Out of the 20 frame combinations where the primary, secondary and tertiary frames are all distinct, only four give a false positive signal in the tertiary frame. Specifically: if there is a coding ORF2 in frame -3, $+1$, $+2$ or -1 relative to a coding ORF1, then there is liable to be a false positive double-coding signal in a non-coding ORF3 in frame $+1$, -3, -1 or $+2$, respectively, relative to ORF1.

In many cases introduced stop codons will result in there being no significant conserved tertiary ORF (Figure 9). Aside from that, while these particular four three-ORF frame combinations may give a false positive signal when the tertiary ORF is tested against the primary ORF, they are all expected to give a negative signal when the tertiary ORF is tested against the secondary ORF, and thus may be selected out in this way.

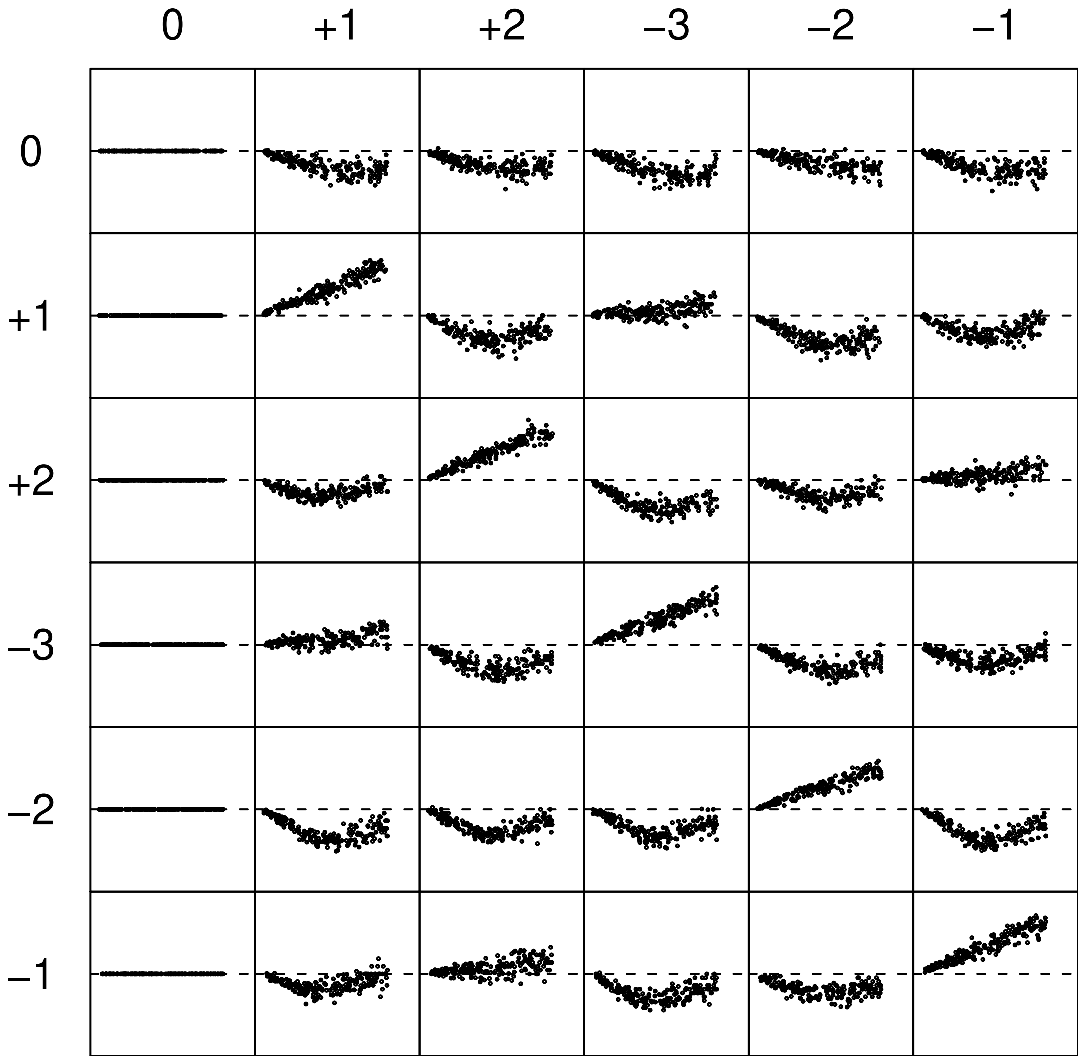


Figure 8: Panels showing the particular combinations of frames that can lead to a false double-coding signal in a third non-coding ORF that overlaps two overlapping coding ORFs. Double-coding sequences were simulated for each of the five possible frames of a secondary ORF relative to a primary ORF, and the MLOGD $\Delta_{\mathrm{s,d}}$ score was evaluated for each of the five possible frames of a tertiary ORF relative to the primary ORF. The frame of the secondary ORF in which double-coding was actually simulated is indicated down the left-hand side, while the frame of the tertiary ORF in which the MLOGD signal was tested is indicated along the top. The horizontal axis (observed mutations per nucleotide, λ) in each panel ranges from 0 to 0.6, while the vertical axis ($\Delta_{\mathrm{s,d}}$) ranges from -0.2 to 0.2. Ideally we would like a negative signal except where the tertiary ORF coincides with the secondary ORF (i.e. down the main diagonal). In fact four other frame combinations also give a positive signal. (Note the scores in the first column are identically zero since here the tertiary ORF is in the same frame as the primary ORF, so MLOGD is just calculating the single-coding versus single-coding likelihood ratio.) In these plots the initial sequence length was 300 codons and the sequence divergence λ ranges from 0 to 0.5.

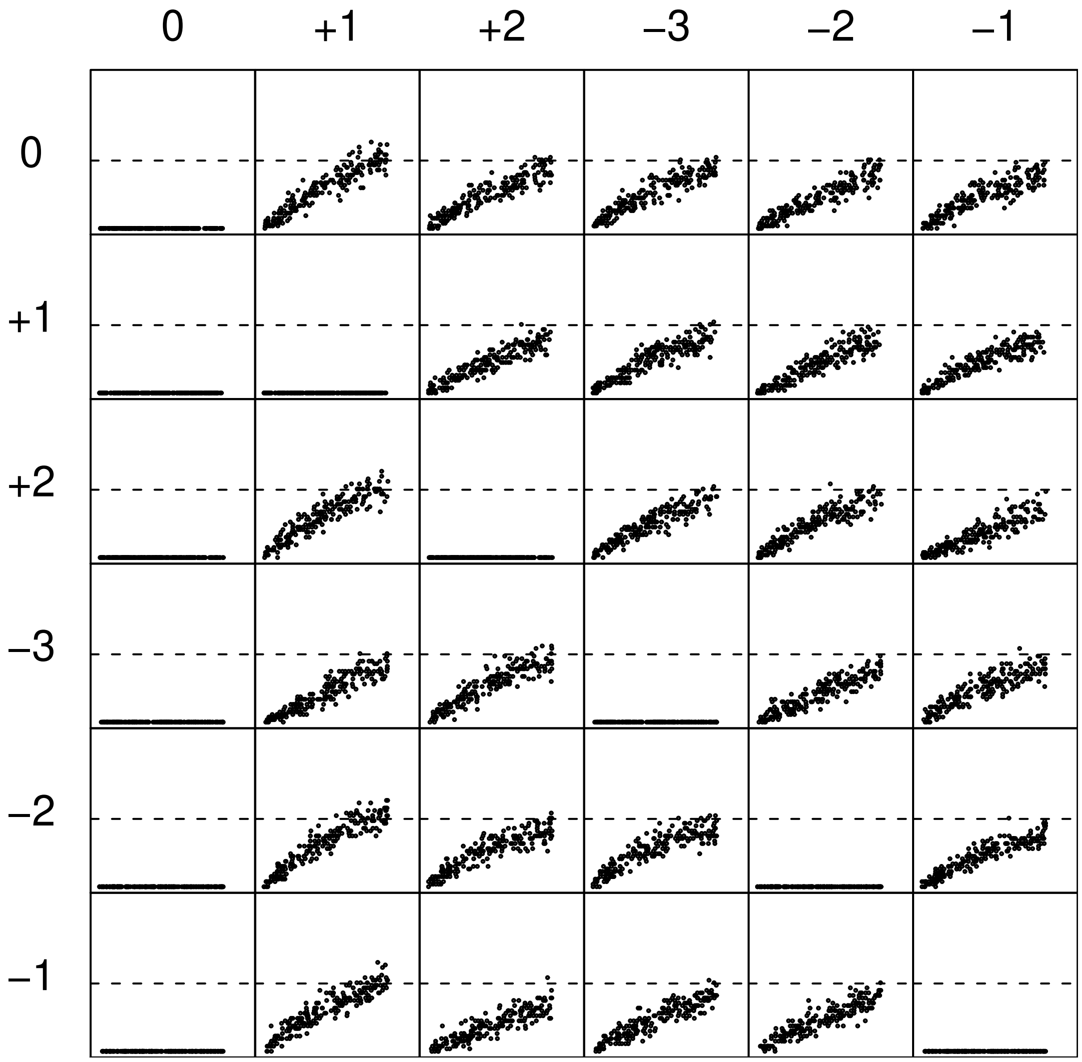


Figure 9: Panels showing the number of stop ↔ non-stop transitions introduced into a third non-coding ORF that overlaps two overlapping coding ORFs, for simulated sequences. See caption to Figure 8 for details. The horizontal axis (observed mutations per nucleotide, λ) in each panel ranges from 0 to 0.6, while the vertical axis (number of stop ↔ non-stop transitions per codon) ranges from 0 to 0.2. For two completely unrelated sequences, the probability of a stop codon aligning with a non-stop codon is $\left(\frac{3}{64} \times \frac{61}{64}\right) + \left(\frac{61}{64} \times \frac{3}{64}\right) = 0.089$, which is indicated by the horizontal dashed lines.

References

Chiaromonte F., Yap V. B., Miller W., 2002, Scoring pairwise genomic sequence alignments, *Pac. Symp. Biocomput. 2002*, 115–126

Dayhoff M. O., Schwartz R. M., Orcutt B. C., 1978, A model of evolutionary change in proteins, *Atlas of Protein Sequence and Structure*, 5, Suppl. 3, 345–358

Ewens W. J., Grant G. R., 2001, Statistical Methods in Bioinformatics, Springer-Verlag, New York

Goldman N., Yang Z., 1994, A codon-based model of nucleotide substitution for protein-coding DNA sequences, *Mol. Biol. Evol.*, 11, 725–736

Grantham R., 1974, Amino acid difference formula to help explain protein evolution, *Science*, 185, 862–864

Henikoff S., Henikoff J. G., 1992, Amino acid substitution matrices from protein blocks, *Proc. Natl. Acad. Sci. USA*, 89, 10915–10919

Kimura M., 1980, A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences, *J. Mol. Evol.*, 16, 111–120

Krakauer D. C., 2000, Stability and evolution of overlapping genes, *Evolution*, 54, 731–739

Pedersen A. K., Wiuf C., Christiansen F. B., 1998, A codon-based model designed to describe lentiviral evolution, *Mol. Biol. Evol.*, 15, 1069–1081

Pedersen A. K., Jensen J. L., 2001, A dependent-rates model and an MCMC-based methodology for the maximum-likelihood analysis of sequences with overlapping reading frames, *Mol. Biol. Evol.*, 18, 763–776

Rogozin I. B. *et al.*, 2002, Purifying and directional selection in overlapping prokaryotic genes, *Trends Genet.*, 18, 228–232

Smith T. F., Waterman M. S., 1980, Protein constraints induced by multiframe encoding, *Math. Biosci.*, 49, 17–26

Yang Z., 1994, Estimating the pattern of nucleotide substitution, *J. Mol. Evol.*, 39, 105–111

Yang Z., Nielsen R., Hasegawa M., 1998, Models of amino acid substitution and applications to mitochondrial protein evolution, *Mol. Biol. Evol.*, 15, 1600–1611