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

March 16, 2005

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.

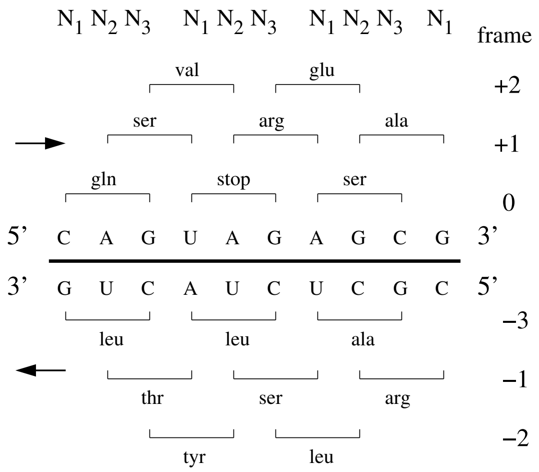


Figure 1: Naming convention for the different possible read-frames of a secondary ORF relative to a frame = 0 primary ORF. In the paper we use the terms 'primary' and 'secondary' to clarify which read-frame we are referring to; it does not imply that one ORF is more important than the other. We use 'frame = 0' to refer to a single-coding region, and 'frame = $-3, \pm 1, \pm 2$' 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

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.

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

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

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

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

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

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, $P(\mathrm{GGU} \to \mathrm{GAU}) = \mathbf{P}(\mathrm{G} \to \mathrm{A}) \times \mathbf{A}(gly \to asp) \times \mathbf{A}(val \to ile)$, 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. $\mathrm{A} \to \mathrm{C} \to \mathrm{G}$ 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, $0 \leq \lambda \leq 0.75$ 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 $\kappa = 3$. For the simulations with random nucleotide sequences, these are the correct π_j to use. Nucleotide substitution matrices are often calculated from sequence alignments by studying substitutions in four-fold degenerate third codon positions. Due to the large number of overlapping CDSs, and other constraints, this is not straightforward for the HBV genome. However the HBV π_j are not strongly biased and so we continue to use equal π_j for simplicity. The value $\kappa = 3$ is similar to that for a human–mouse comparison (Chiaromonte *et al.* 2002).

Note that to convert the Chiaromonte *et al.* nucleotide substitution matrices to the form used in our software, one may proceed as follows. Chiaromonte *et al.* provide matrices for three different %GC values. These matrices are log odds scores

$$s(i,j) = A \log\left(\frac{f(i,j)}{f(i)f(j)}\right), \quad i,j = \mathrm{U,C,A,G}, \tag{10}$$

where $f(i,j)$ is the frequency of aligned nucleotides i and j, $f(j) = \pi_j$ is the frequency of nucleotide j, and A is some unspecified scaling factor. Using the π_j defined by the given %GC values and the constraint that $\sum f(i,j) = 1$, we can find A and hence the matrix $\mathbf{P}(t) = [p_{ij}(t)]$ where $p_{ij}(t) = f(i,j)/f(i) = P(j|i)$ is essentially the probability of nucleotide i mutating to j after some time t representing the human–mouse divergence. Diagonalizing $\mathbf{P}(t)$ allows us to find $\mathbf{Q}t = \log \mathbf{P}(t)$. The resulting matrix is essentially equivalent to that described in equation 2. The parameters a, b, c, d, e, f may be recovered by factoring out the π_j. The resulting three sets of a, c, d, f values are similar for the three Chiaromonte *et al.* matrices, while the b, e values vary by 60–70%. The a, b, c, d, e, f values may then be remultiplied by π_j for a particular sequence alignment. Finally the diagonals are set and the matrix is normalized, as in equations 1 and 3 respectively, to give the instantaneous nucleotide mutation matrix $\mathbf{Q}$.

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. $\mathrm{A} \to \mathrm{A}$) or non-null (e.g. $\mathrm{A} \to \mathrm{C}$) 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. For the simulations with random nucleotide sequences, this is the correct CUT to use. For HBV, due to the large number of overlapping CDSs, and other constraints, it is not clear that a CUT generated directly from the HBV genome will be representative of mutation probabilities. A human CUT may be more appropriate. However the human CUT is not strongly biased (cf. certain prokaryotes) and so we use a null CUT for simplicity. The fact that the Monte Carlo simulations agree well with the observed statistics (e.g. Figure 9 in the paper) suggests that this approach is justified. 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 $\rightarrow$ CUG (*leu*) will be a factor of six less probable than ?UG $\rightarrow$ 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 $X_i \rightarrow X_j$ relative to $X_i \rightarrow X_i$ 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

$$\begin{aligned} b(N_1^k \to i; t, \mathrm{s}) \;=\; & \mathbf{P}(N_1^k \to i; t) \times \mathbf{C}(X_2) \\ & \times \mathbf{A}(X_1 \to X_2) \end{aligned} \tag{11}$$

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

$$\begin{aligned} b(N_1^k \to i; t, \mathrm{d}) \;=\; & \mathbf{P}(N_1^k \to i; t) \times \mathbf{C}(X_2) \\ & \times \mathbf{A}(X_1 \to X_2) \times \mathbf{C}(X_2') \\ & \times \mathbf{A}(X_1' \to X_2') \end{aligned} \tag{12}$$

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

$$P(N_1^k \to N_2^k; t, m) = \frac{b(N_1^k \to N_2^k; t, m)}{\sum_{i=\mathrm{U,C,A,G}} b(N_1^k \to i; t, m)}. \tag{13}$$

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

$$\log P(S_1 \to S_2; t, m) = \sum_{k=1}^{N_{\mathrm{nucleotides}}} \log P(N_1^k \to N_2^k; t, m). \tag{14}$$

We maximize equation 14 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

$$\Delta_{\mathrm{s,d}} = \log P(S_1 \to S_2; t^{\mathrm{d}}, \mathrm{d}) - \log P(S_1 \to S_2; t^{\mathrm{s}}, \mathrm{s}). \tag{15}$$

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 14), 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

$$\Delta_{\mathrm{s,d}}^{\mathrm{N123}} = \log\left[\frac{P(f_{\mathrm{N1}}, f_{\mathrm{N2}}, f_{\mathrm{N3}}|\mathrm{d}, \lambda)}{P(f_{\mathrm{N1}}, f_{\mathrm{N2}}, f_{\mathrm{N3}}|\mathrm{s}, \lambda)}\right] \tag{16}$$

and

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

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

$$\Delta_{\mathrm{s,d}}^{\mathrm{MLOGD}} = \log\left[\frac{P(\Delta_{\mathrm{s,d}}|\mathrm{d}, \lambda)}{P(\Delta_{\mathrm{s,d}}|\mathrm{s}, \lambda)}\right]. \tag{18}$$

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 $z_x^m = \frac{x - \mu_x^m}{\sigma_x^m}$,

$$P(x|m, \lambda) = \frac{1}{\sqrt{2\pi}\sigma_x^m} \exp\left[-\frac{1}{2}(z_x^m)^2\right] \tag{19}$$

giving the log likelihood ratio

$$\log\left[\frac{P(x|\mathrm{d}, \lambda)}{P(x|\mathrm{s}, \lambda)}\right] = \log\left(\frac{\sigma_x^{\mathrm{s}}}{\sigma_x^{\mathrm{d}}}\right) + \frac{1}{2}\left((z_x^{\mathrm{s}})^2 - (z_x^{\mathrm{d}})^2\right). \tag{20}$$

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

$$\Delta_{\mathrm{s,d}}^{\mathrm{N123}} = \log\left[\frac{P(f_{\mathrm{N1}}|\mathrm{d}, \lambda)}{P(f_{\mathrm{N1}}|\mathrm{s}, \lambda)}\right] + \log\left[\frac{P(f_{\mathrm{N2}}|\mathrm{d}, \lambda)}{P(f_{\mathrm{N2}}|\mathrm{s}, \lambda)}\right] \tag{21}$$

and

$$\Delta_{\mathrm{s,d}}^{\mathrm{NsNn}} = \log\left[\frac{P(f_{\mathrm{syn}}|\mathrm{d}, \lambda)}{P(f_{\mathrm{syn}}|\mathrm{s}, \lambda)}\right] + \log\left[\frac{P(f_{\mathrm{non}}|\mathrm{d}, \lambda)}{P(f_{\mathrm{non}}|\mathrm{s}, \lambda)}\right]. \tag{22}$$

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 $\Delta_{\mathrm{s,d}}^{\mathrm{NsNn}} > 0$.

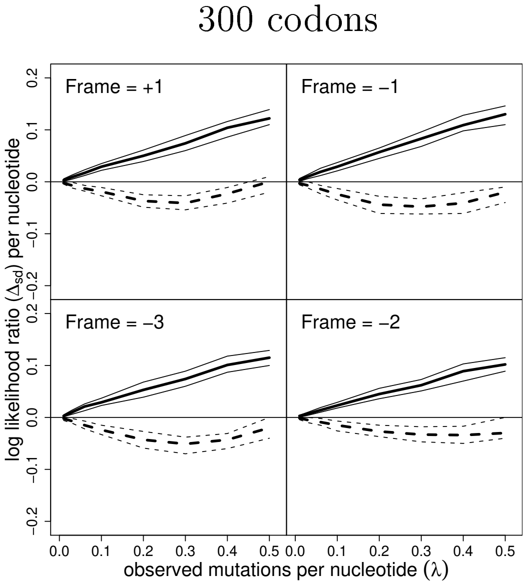
This classification scheme is simple to implement but ignores such things as non-independence of f_{N1}, f_{N2} and of f_{syn}, f_{non}, and deviations from normal distributions (e.g. f_{N1}, f_{N2}, f_{syn}, f_{non} are constrained to be non-negative), so $\Delta_{\mathrm{s,d}}^{\mathrm{MLOGD}}$, $\Delta_{\mathrm{s,d}}^{\mathrm{N123}}$ 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.

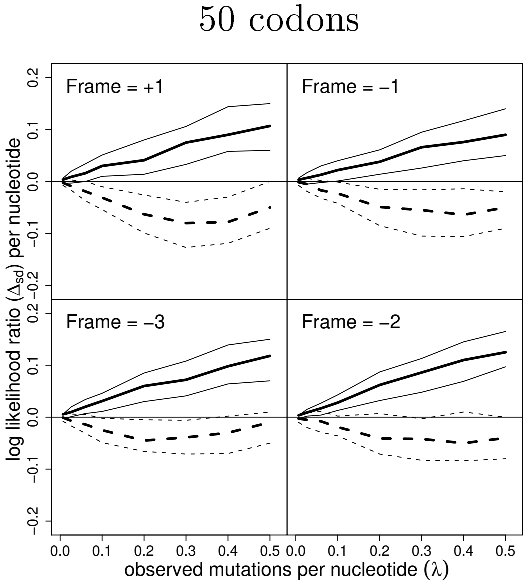
For reasons mentioned above, none of $\Delta^x = \Delta_{\mathrm{s,d}}$, $\Delta_{\mathrm{s,d}}^{\mathrm{MLOGD}}$, $\Delta_{\mathrm{s,d}}^{\mathrm{N123}}$ or $\Delta_{\mathrm{s,d}}^{\mathrm{NsNn}}$ are exactly likelihood ratios. If they were, then the relative probability of a given sequence being double-coding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

$$\begin{aligned} \frac{P(\mathrm{d}|\mathrm{S}_1+\mathrm{S}_2)}{P(\mathrm{s}|\mathrm{S}_1+\mathrm{S}_2)} &= \frac{P(\mathrm{d})}{P(\mathrm{s})}\frac{P(\mathrm{S}_1+\mathrm{S}_2|\mathrm{d})}{P(\mathrm{S}_1+\mathrm{S}_2|\mathrm{s})} \\ &= \frac{P(\mathrm{d})}{P(\mathrm{s})}\exp(\Delta^x) \end{aligned} \tag{23}$$

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





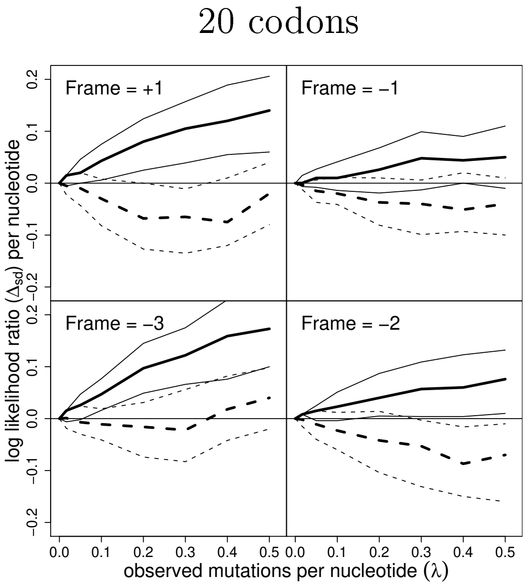
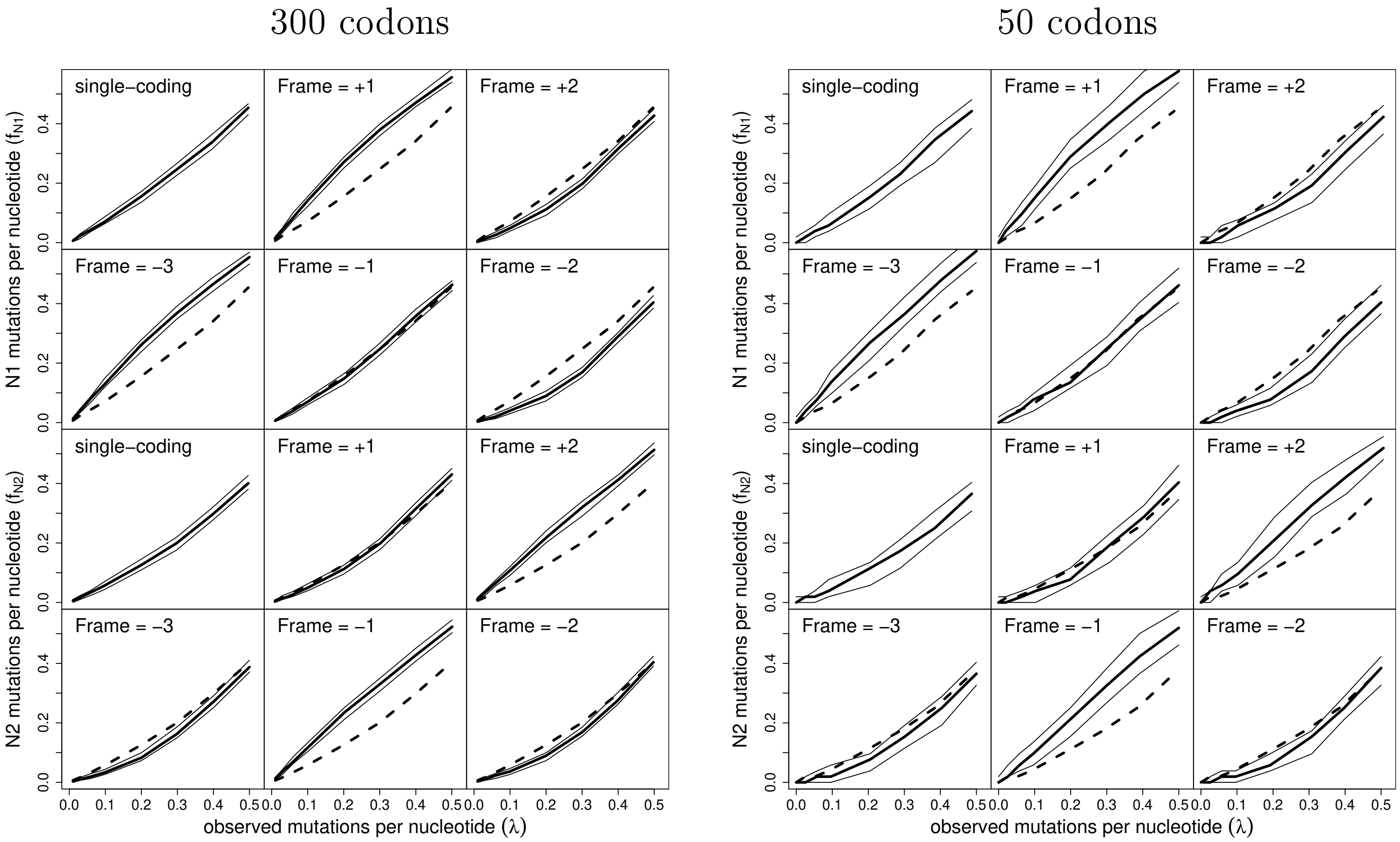


Figure 2: Loci of the MLOGD statistic $\Delta_{\mathrm{s,d}}$ (per nucleotide) for simulated single-coding (dashed lines) and double-coding (solid lines) sequences. The thick lines represents the median values while the thin lines enclose the central 68% of values. Sequence divergence is measured on 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.



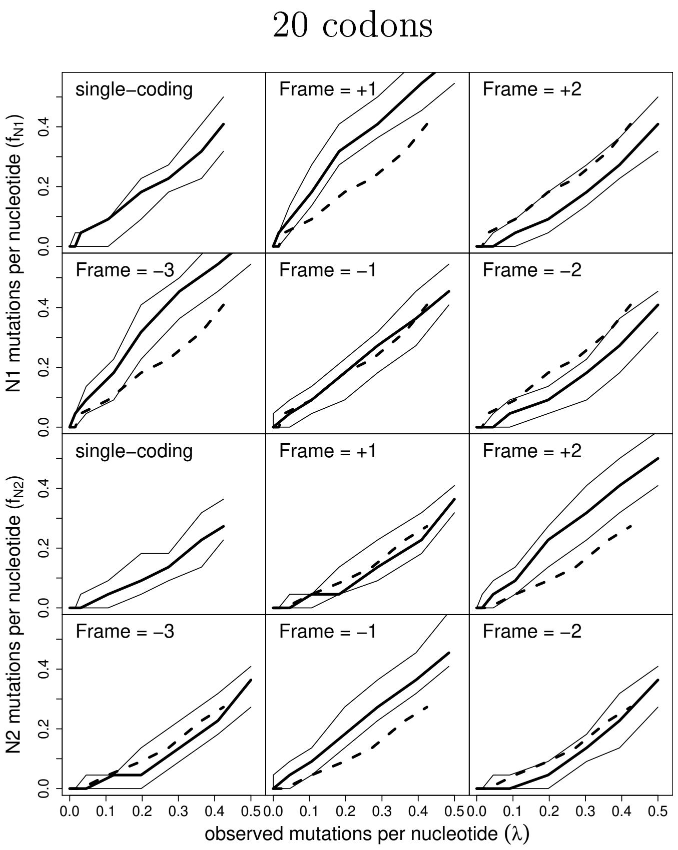
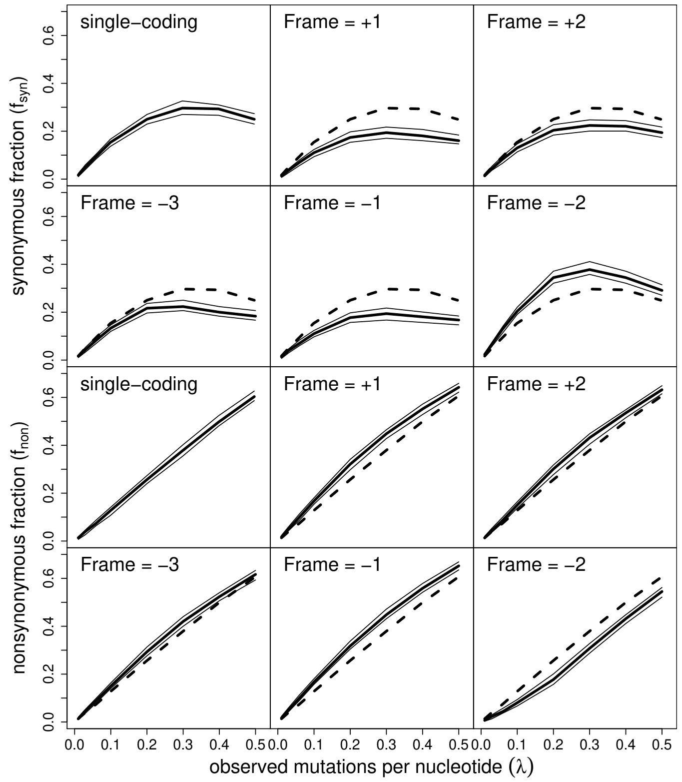
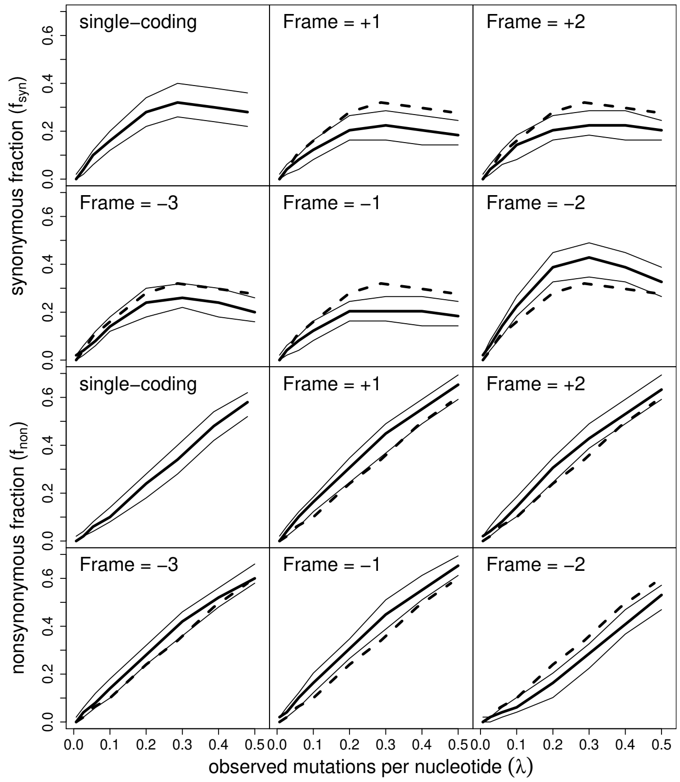


Figure 3: Loci of the 1st and 2nd codon position mutation rates, f_{N1} and f_{N2}, as a function of mean mutation rate λ, for simulated sequences. The thick lines represents the median values while the thin lines enclose the central 68% of values. Sequence divergence is measured on the x-axis. The twelve 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.

300 codons



50 codons



20 codons

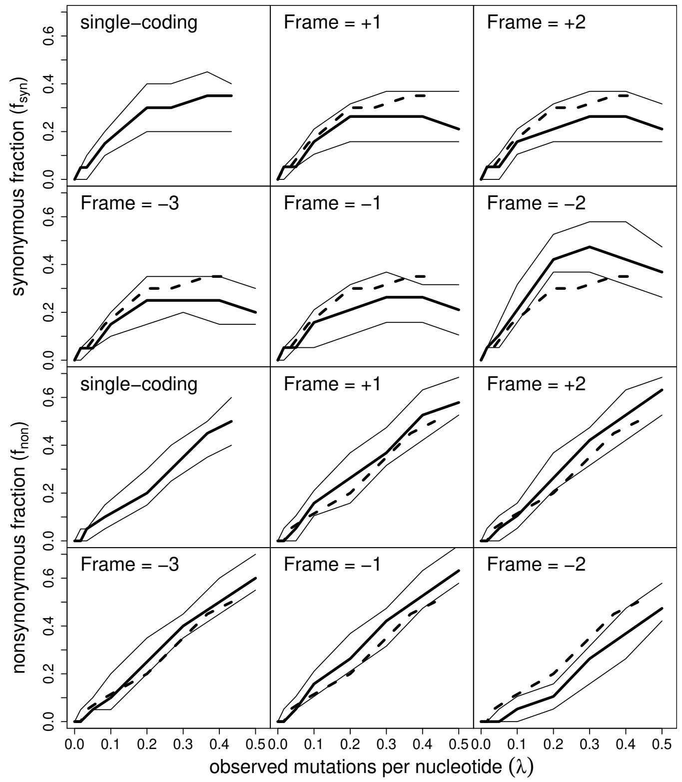
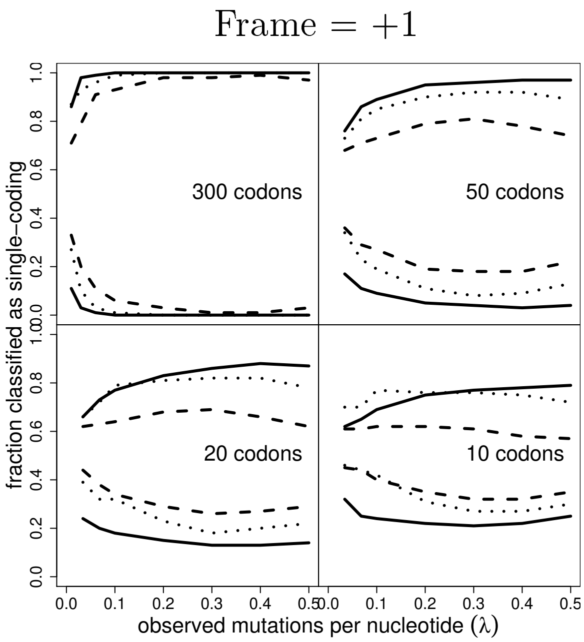
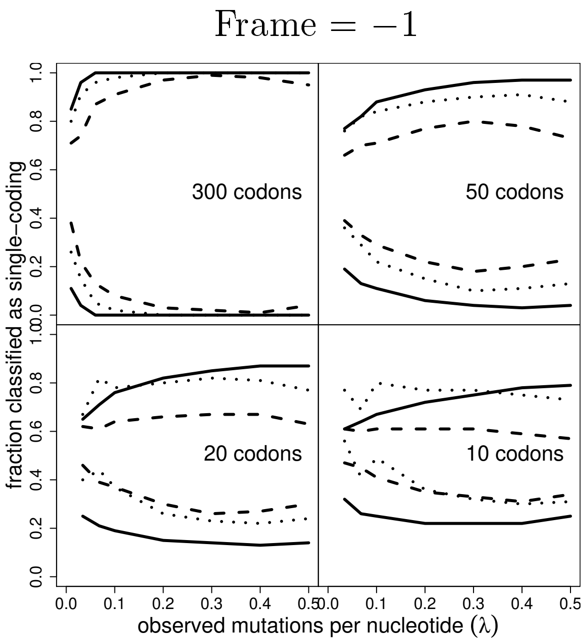
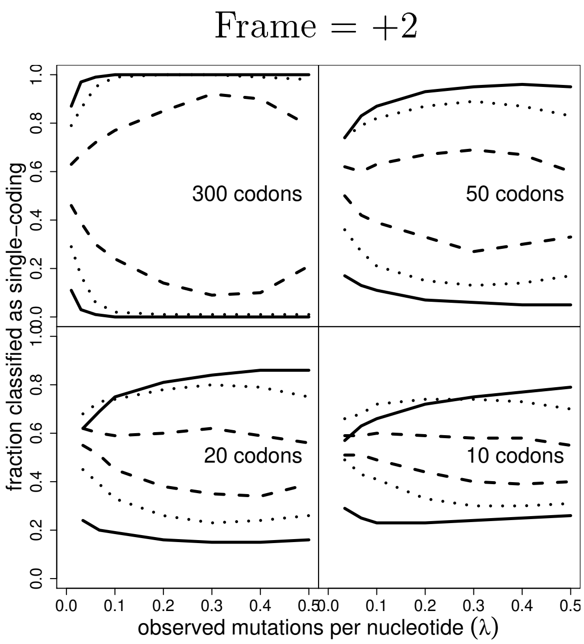
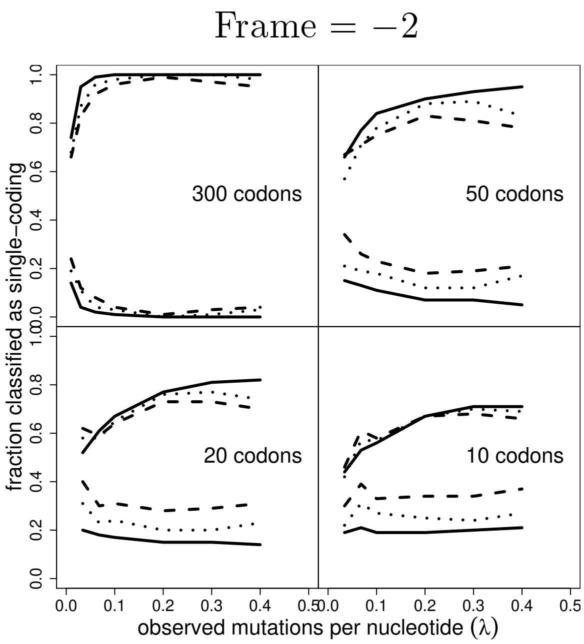


Figure 4: Loci of the synonymous and nonsynonymous mutation rates, f_{syn} and f_{non}, as a function of mean mutation rate λ, for simulated sequences. The thick lines represents the median values while the thin lines enclose the central 68% of values. Sequence divergence is measured on the x-axis. The twelve panels show the loci of f_{syn} and f_{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.









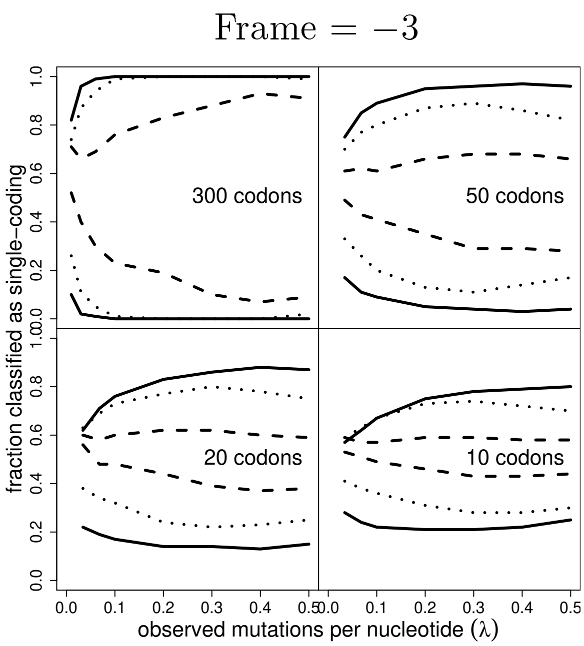
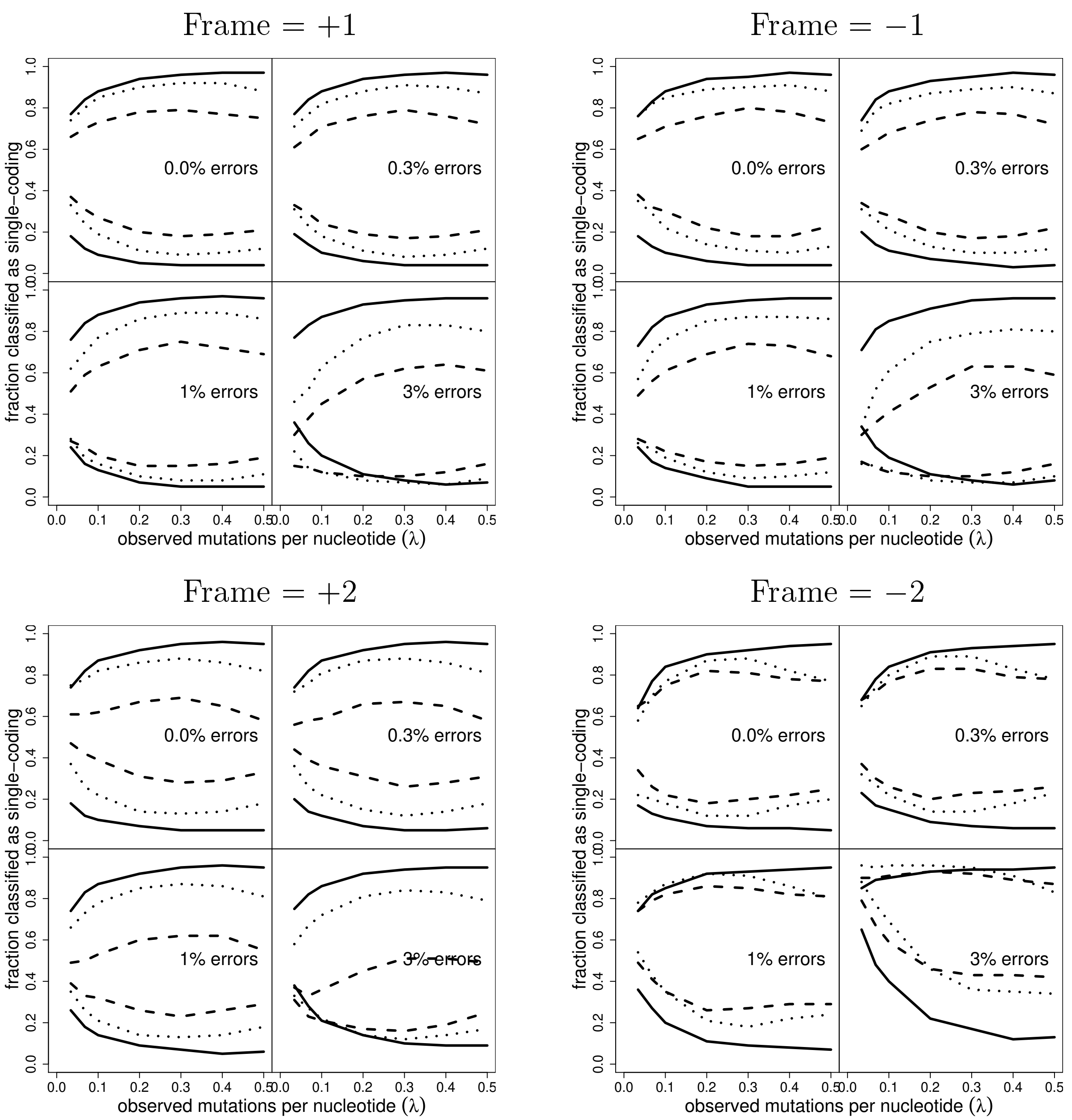


Figure 5: Classification success as a function of mutation rate λ and sequence length, for different double-coding frames. In each panel the upper three lines show the fraction of single-coding simulations classified as single-coding ($1 \equiv$ perfect classification) while the lower three lines show 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 $\sim\frac{2000}{N_{\mathrm{codons}}}$ 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.



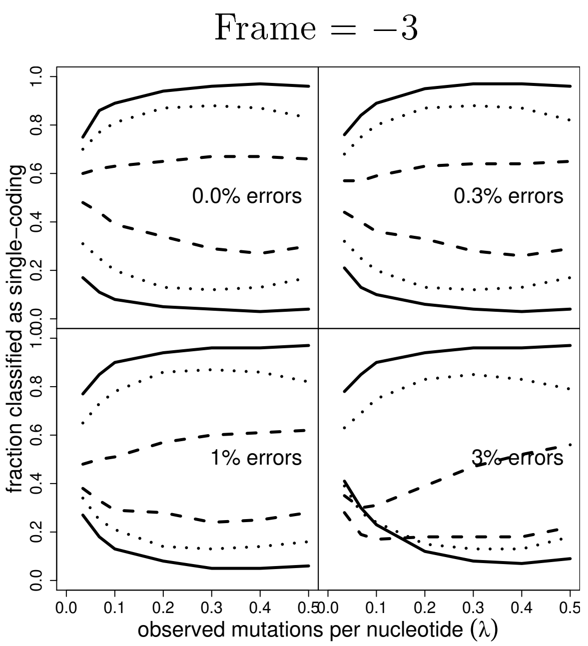
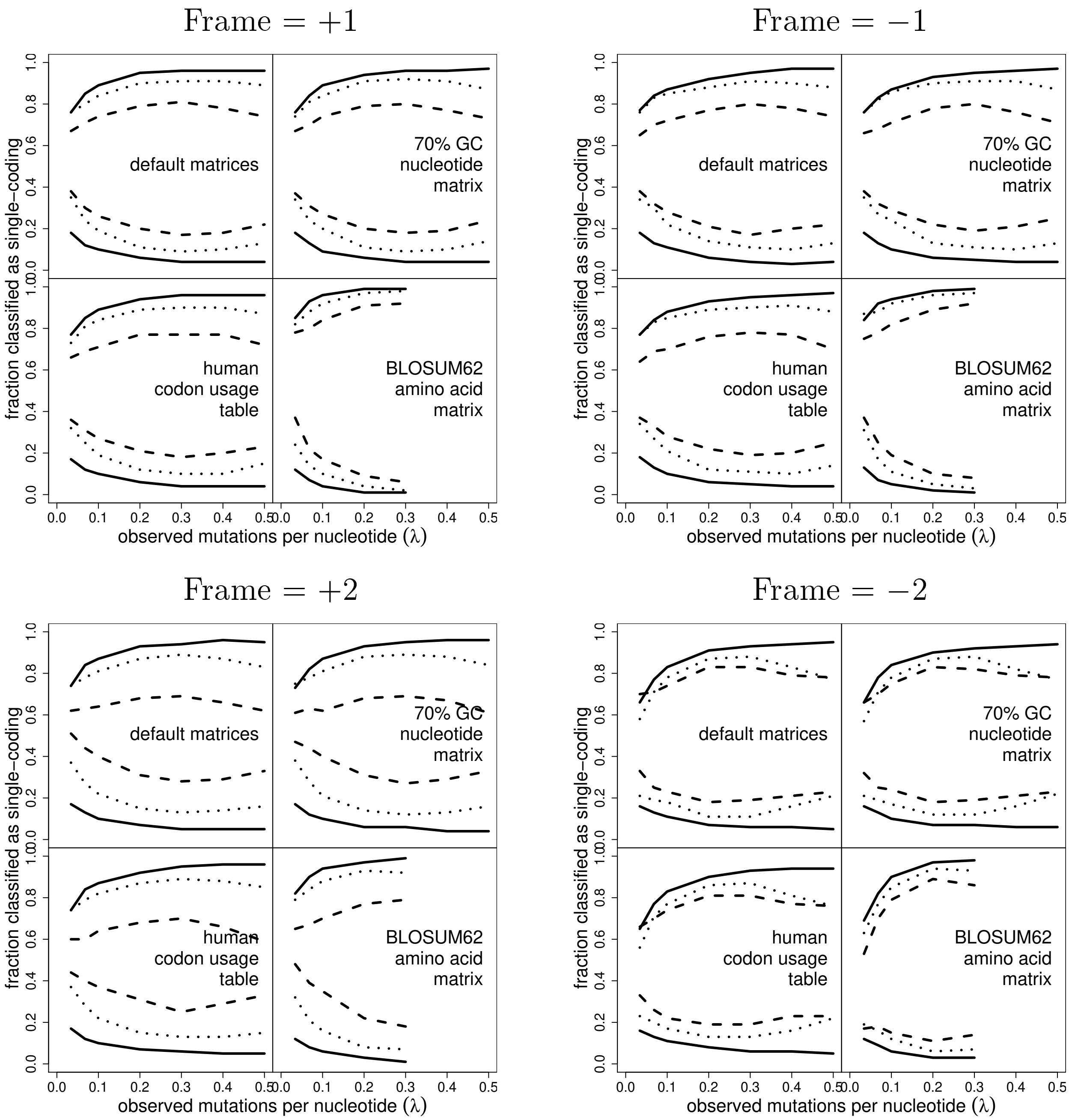


Figure 6: Reduction in classification success as a result of sequencing errors. The error rates per nucleotide are indicated in each panel. Sequence length = 50 codons. See caption to Figure 5 for further details.



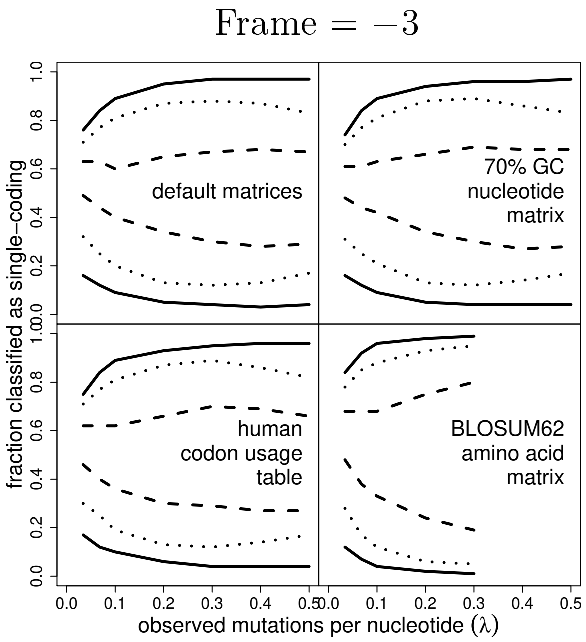


Figure 7: Reduction in classification success as a result of incorrect model nucleotide, codon or amino acid matrices. The default matrices ($\kappa = 3$ Kimura nucleotide matrix, null CUT – i.e. equal codon frequencies, and BLOSUM40 amino acid matrix) are used for classification and the different matrices 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

A large fraction of the HBV genome comprises overlapping coding sequences. 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.

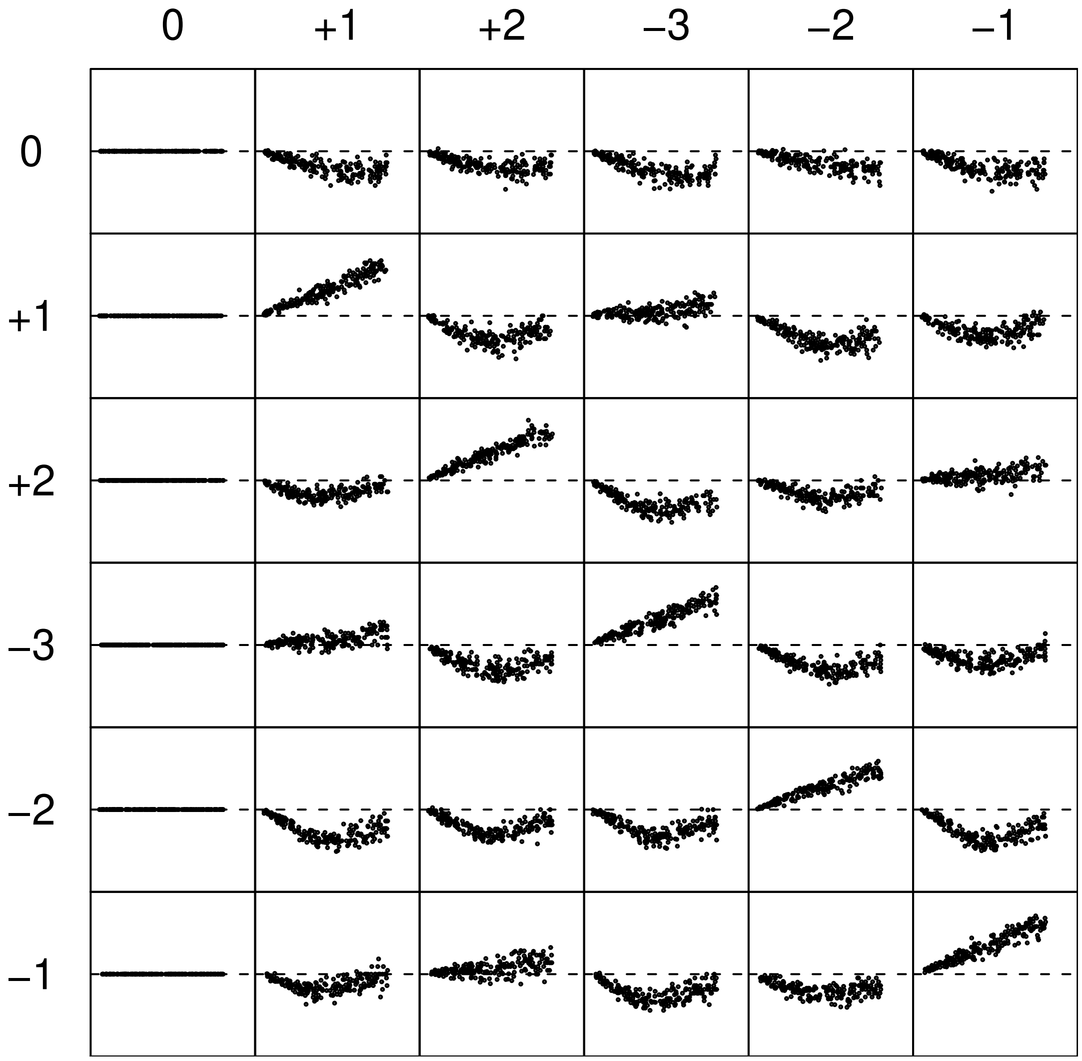


Figure 8: Panels showing the particular combinations of frames that can lead to a false double-coding signal in a third non-coding ORF that overlaps two overlapping coding ORFs. Double-coding sequences were simulated for each of the five possible frames of a secondary ORF relative to a primary ORF, and 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.

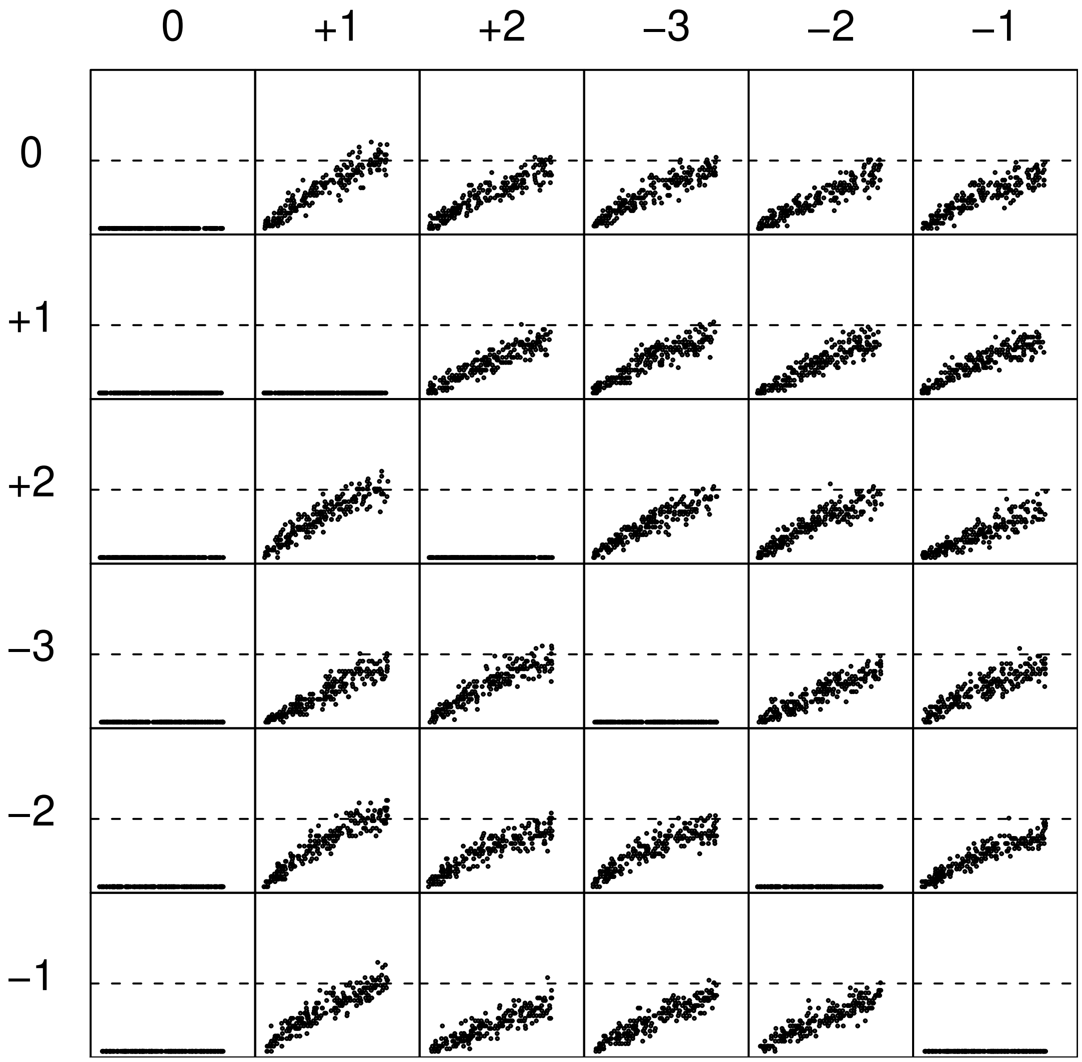


Figure 9: Panels showing the number of stop $\leftrightarrow$ non-stop transitions introduced into a third non-coding ORF that overlaps two overlapping coding ORFs, for simulated sequences. See caption to Figure 8 for details. The horizontal axis (observed mutations per nucleotide, λ) in each panel ranges from 0 to 0.6, while the vertical axis (number of stop $\leftrightarrow$ 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.

5 Hepatitis B results

Table 3: Hepatitis B virus (HBV) sequences used in the paper. (Genotypes from Huy *et al.* 2004.)

GenBank ID	version	genotype	notes
NC_003977.1	GI:21326584	C2	based on X04615; reference sequence
X70185.1	GI:59455	A	
D00329.1	GI:221497	B	
X75665.1	GI:452622	C	
AB074755.1	GI:21624227	C1	
D50520.1	GI:2627005	C2	
X02496.1	GI:62280	D	
X75664.1	GI:452627	E	
X75663.1	GI:452632	F	
AB056514.1	GI:15425692	G	
AF046996.1	GI:3150070	woolly monkey	
J02442.1	GI:336126	woodchuck	

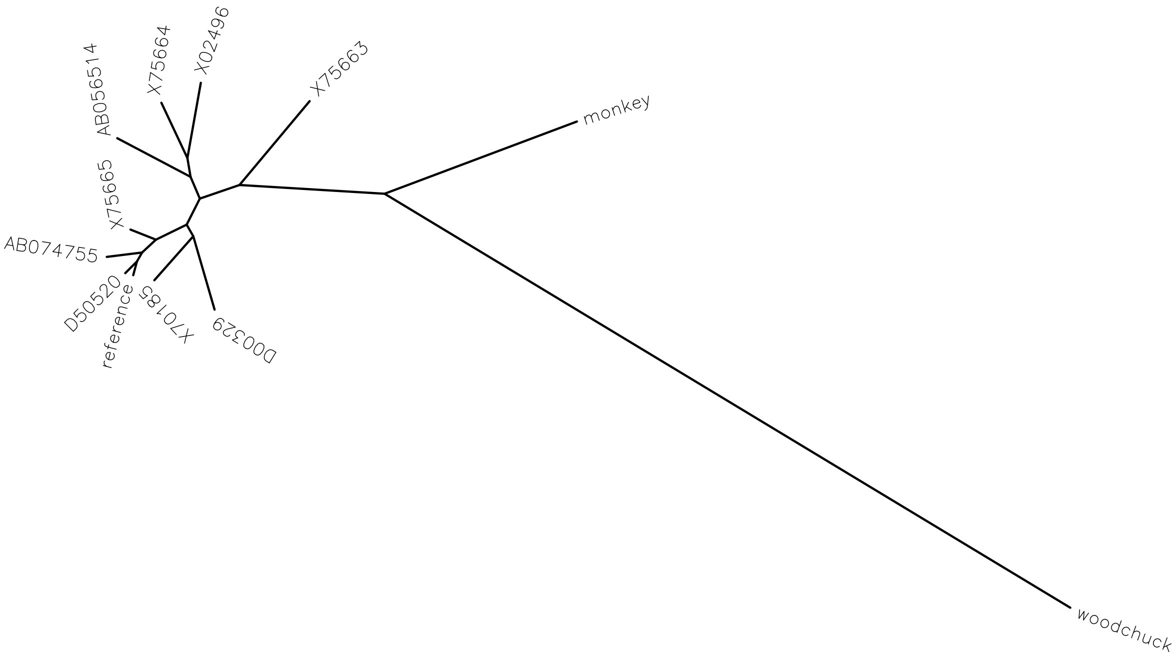


Figure 10: Phylogenetic tree for the P gene, generated with the programme `DNAML` in the `PHYLIP` package (Felsenstein 2004). Genotype labels are from Huy *et al.* 2004 (see Huy *et al.* for a phylogenetic tree of the human strains, based on the full genome).

Table 4: Table of distances. These are the mean number of observed mutations per nucleotide (which we call λ) in the P gene. (Note that these are not the distances used by `DNAML` in Figure 10. See `PHYLIP` documentation for details of those distances.)

	NC_003977	X70185	D00329	X75665	AB074755	D50520	X02496	X75664	X75663	AB056514	AF046996	J02442
NC_003977	0											
X70185	0.08	0										
D00329	0.09	0.08	0									
X75665	0.04	0.08	0.09	0								
AB074755	0.04	0.08	0.09	0.05	0							
D50520	0.02	0.08	0.09	0.04	0.04	0						
X02496	0.10	0.10	0.10	0.10	0.10	0.10	0					
X75664	0.10	0.11	0.11	0.10	0.11	0.10	0.09	0				
X75663	0.13	0.13	0.14	0.13	0.13	0.14	0.13	0.13	0			
AB056514	0.11	0.10	0.12	0.11	0.11	0.12	0.10	0.09	0.14	0		
AF046996	0.22	0.23	0.23	0.22	0.22	0.22	0.22	0.22	0.23	0.23	0	
J02442	0.39	0.39	0.40	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0

Table 5: Nucleotide range and length of all ORFs with length greater than or equal to 30 nucleotides in the reference sequence. The HBV reference sequence (NC_003977) has 3215 nucleotides. Nucleotide positions greater than 3215 are used for ORFs overlapping the sequence end (due to HBV having a circular genome). See Figure 11 for a diagram of the HBV genome with these ORFs marked.

ORF ID	nucleotide range		length	notes
ORF_11	2307	4835	2529	P gene
ORF_10	2713	4047	1335	S gene
ORF_7	1814	2449	636	C gene
ORF_5	1374	1835	462	X gene
ORF_24	1609	1325	285	
ORF_18	2555	2370	186	
ORF_23	1566	1384	183	
ORF_34	3371	3201	171	
ORF_8	2446	2601	156	
ORF_28	831	697	135	
ORF_22	1704	1594	111	
ORF_17	2744	2634	111	
ORF_25	1269	1168	102	
ORF_2	987	1079	93	
ORF_26	1235	1146	90	
ORF_21	1842	1756	87	
ORF_12	3170	3093	78	
ORF_3	1064	1135	72	
ORF_33	170	102	69	
ORF_30	609	541	69	
ORF_31	494	429	66	
ORF_29	617	552	66	
ORF_19	2141	2076	66	
ORF_6	1846	1908	63	
ORF_35	3226	3164	63	
ORF_32	429	370	60	
ORF_13	3034	2984	51	
ORF_20	1908	1867	42	
ORF_27	1069	1031	39	
ORF_16	2692	2654	39	
ORF_14	2882	2844	39	
ORF_9	2579	2608	30	
ORF_15	2823	2794	30	

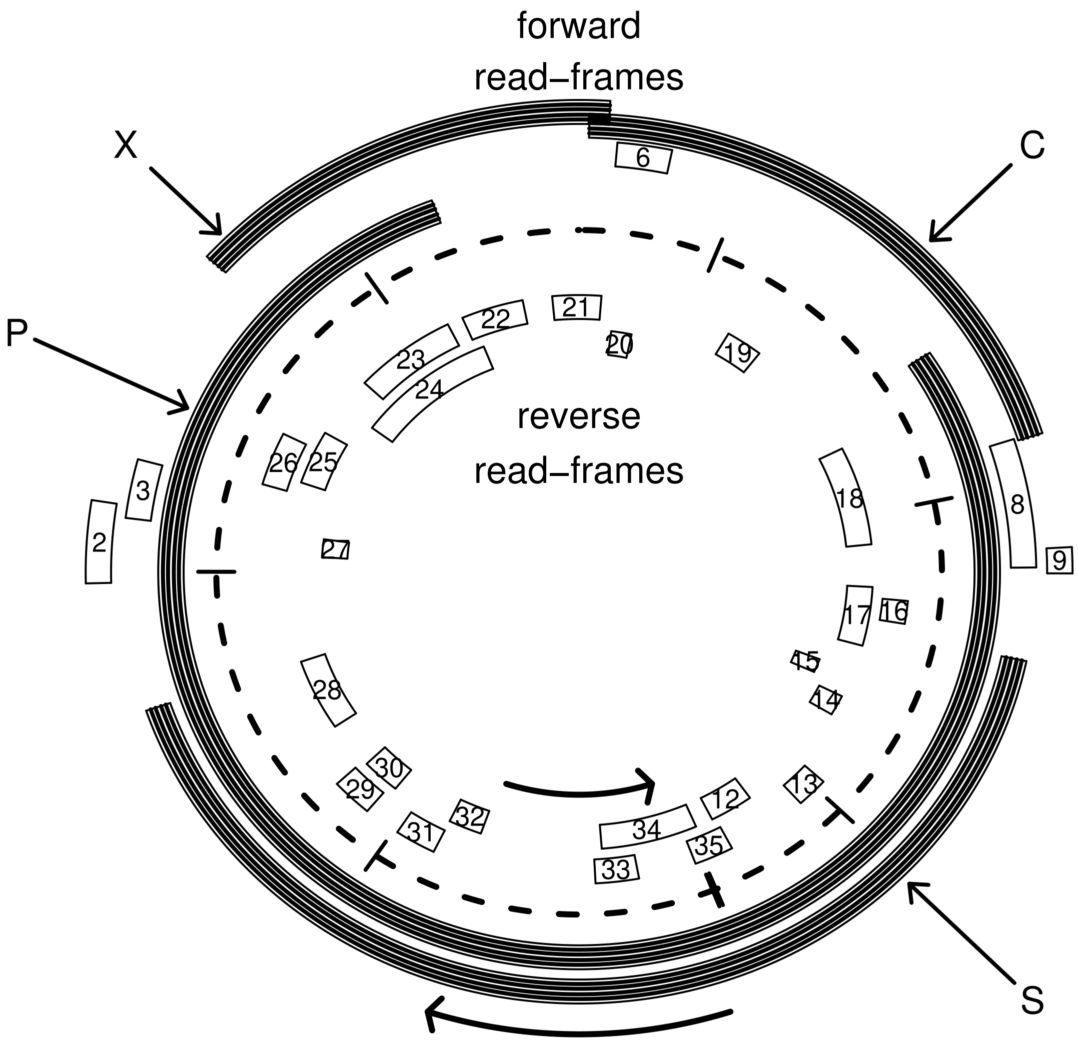


Figure 11: Diagram of the 3215-nucleotide circular HBV genome (reference sequence NC_003977). P, C, S and X are known protein-coding genes. Open boxes represent all other ORFs 30 nucleotides or longer in the reference sequence. The numbering corresponds to that in Table 5. The radial tickmarks on the dashed line are at 500-nucleotide intervals with the zero-point (in Table 5) represented by the thicker tickmark at bottom right.

Table 6: List of all overlapping pairs of ORFs, ORF1 and ORF2, in the reference sequence (NC_003977), with ORF1 at least 360 nucleotides and the overlap region at least 30 nucleotides in length. The nucleotide range is the range of ORF2 within (i.e. truncated in-frame if it overlaps an end) and relative to ORF1. The length is the length of the overlap region. The frame is the read-frame of ORF2 relative to ORF1.

overlap ID	ORF1 ID	ORF2 ID	nucleotide range		length	frame	notes
1	ORF_11	ORF_10	407	1741	1335	1	S in P
2	ORF_11	ORF_7	3	143	141	2	C in P
3	ORF_11	ORF_18	64	249	186	-3	
4	ORF_11	ORF_34	895	1065	171	-3	
5	ORF_11	ORF_8	140	295	156	1	
6	ORF_11	ORF_17	328	438	111	-3	
7	ORF_11	ORF_12	787	864	78	-3	
8	ORF_11	ORF_35	858	920	63	-2	
9	ORF_11	ORF_13	678	728	51	-2	
10	ORF_11	ORF_16	348	386	39	-2	
11	ORF_11	ORF_14	538	576	39	-3	
12	ORF_11	ORF_9	273	302	30	2	
13	ORF_11	ORF_15	488	517	30	-1	
14	ORF_11	ORF_5	2283	2528	246	2	X in P
15	ORF_11	ORF_24	2234	2518	285	-1	
16	ORF_11	ORF_23	2293	2475	183	-3	
17	ORF_11	ORF_28	1606	1740	135	-3	
18	ORF_11	ORF_25	2077	2178	102	-3	
19	ORF_11	ORF_2	1896	1988	93	2	
20	ORF_11	ORF_26	2055	2144	90	-2	
21	ORF_11	ORF_3	1973	2044	72	1	
22	ORF_11	ORF_33	1011	1079	69	-2	
23	ORF_11	ORF_30	1450	1518	69	-3	
24	ORF_11	ORF_31	1338	1403	66	-2	
25	ORF_11	ORF_29	1461	1526	66	-2	
26	ORF_11	ORF_32	1279	1338	60	-3	
27	ORF_11	ORF_27	1940	1978	39	-1	
28	ORF_10	ORF_11	3	1334	1332	2	P in S
29	ORF_10	ORF_34	489	659	171	-2	
30	ORF_10	ORF_17	3	32	30	-2	
31	ORF_10	ORF_12	381	458	78	-2	
32	ORF_10	ORF_35	452	514	63	-1	
33	ORF_10	ORF_13	272	322	51	-1	
34	ORF_10	ORF_14	132	170	39	-2	
35	ORF_10	ORF_15	82	111	30	-3	
36	ORF_10	ORF_28	1200	1334	135	-2	
37	ORF_10	ORF_33	605	673	69	-1	
38	ORF_10	ORF_30	1044	1112	69	-2	
39	ORF_10	ORF_31	932	997	66	-1	
40	ORF_10	ORF_29	1055	1120	66	-1	
41	ORF_10	ORF_32	873	932	60	-2	
42	ORF_7	ORF_11	494	634	141	1	P in C
43	ORF_7	ORF_18	557	634	78	-1	
44	ORF_7	ORF_19	263	328	66	-1	
45	ORF_7	ORF_6	33	95	63	2	
46	ORF_7	ORF_20	54	95	42	-2	
47	ORF_5	ORF_24	3	236	234	-2	
48	ORF_5	ORF_23	11	193	183	-1	
49	ORF_5	ORF_22	221	331	111	-1	
50	ORF_5	ORF_21	383	460	78	-1	
51	ORF_5	ORF_11	2	247	246	1	P in X

Table 7: Table of MLOGD, NsNn and N123 likelihood scores for the overlap regions listed in Table 6. Overlap IDs 30 and 35 are omitted due to a lack of successfully aligned sequences. Here, length is the mean number of nucleotides used to calculate the scores (i.e. averaged over sequence pairs and not including nucleotides aligned to gaps or in primary or secondary frame codons that overlap gaps). $\Delta_{\mathrm{s,d}}$ is the score calculated directly by the MLOGD method (§2.4.3). The MLOGD, NsNn and N123 likelihood scores are calculated by comparing the observed MLOGD, NsNn and N123 scores with the results from 100 single-coding and 100 double-coding simulated sequences at the same sequence divergence λ as the observed sequence pair (§2.5). Negative scores support single-coding; positive scores support double-coding. All scores have been summed over all eleven reference/non-reference species pairs, so – due to non-independence – they can not be converted to probabilities in the usual way. A + in the last column indicates ORFs that overlap two known genes (e.g. P and S) and that, due to the particular combination of the three frames, may be expected to show a false double-coding signal in the MLOGD method (§4). A + in parentheses is used where the double overlap is only partial.

				likelihood				
overlap ID	ORF1	length	$\Delta_{\mathrm{s,d}}$	$\sum \Delta_{\mathrm{s,d}}^{\mathrm{MLOGD}}$	$\sum \Delta_{\mathrm{s,d}}^{\mathrm{NsNn}}$	$\sum \Delta_{\mathrm{s,d}}^{\mathrm{N123}}$	notes	+ve
1	P	1308	516.4	356.0	149.0	195.1	S in P	
2	P	132	43.4	27.2	19.9	33.2	C in P	
3	P	180	-25.7	-24.5	2.0	-22.5	partly in C	
4	P	164	31.5	23.6	21.9	41.3	also in S	+
5	P	150	-19.4	-17.7	-27.8	-22.4		
6	P	105	-64.0	-54.4	-5.7	-12.3	partly in S	(+)
7	P	72	30.1	23.5	24.4	10.7	also in S	+
8	P	57	-22.1	-47.4	-48.8	-47.5	also in S	
9	P	42	-48.4	-69.0	-55.6	-98.7	also in S	
10	P	33	3.0	0.2	-3.7	7.5		
11	P	30	15.4	9.7	1.7	-6.1	also in S	+
12	P	24	-14.6	-10.4	-5.8	-7.2		
13	P	24	-18.7	-25.2	-2.7	-0.1	also in S	
14	P	240	36.6	21.5	-14.7	13.0	X in P	
15	P	279	-65.1	-58.5	-81.2	-21.8	also in X	+
16	P	177	-44.7	-41.0	-22.0	-51.0	also in X	
17	P	129	-7.5	-8.0	-4.8	4.2	also in S	+
18	P	96	-35.7	-35.3	-2.3	-18.5		
19	P	87	-1.5	-3.0	-25.7	-24.7		
20	P	84	-0.2	-4.0	-7.8	-2.1		
21	P	66	-1.4	-0.9	-14.1	7.5		
22	P	63	-15.6	-34.3	-17.3	-22.2	also in S	
23	P	63	13.2	6.3	-0.5	12.6	also in S	+
24	P	60	-18.8	-25.1	-30.8	-44.2	also in S	
25	P	60	-12.8	-22.2	-18.6	-27.4	also in S	
26	P	54	-2.4	-2.3	2.2	7.0	also in S	+
27	P	33	-14.0	-7.0	-19.6	-7.0		
28	S	1198	398.0	232.1	34.3	93.3	P in S	
29	S	161	-17.4	-35.6	-51.1	-13.5	also in P	
31	S	72	-12.1	-12.6	-17.6	-23.0	also in P	
32	S	57	3.1	-6.0	-8.0	-8.3	also in P	+
33	S	45	-6.1	-5.2	-10.2	-13.6	also in P	+
34	S	25	-15.2	-27.2	-6.1	-6.4	also in P	
36	S	129	-32.2	-36.7	-28.7	-9.0	also in P	
37	S	63	20.4	7.8	10.7	28.9	also in P	+
38	S	63	-7.1	-8.0	-3.4	-1.1	also in P	
39	S	60	-1.3	-0.3	-0.2	-6.5	also in P	+
40	S	60	9.7	8.4	-8.9	-1.1	also in P	+
41	S	54	-6.8	-6.8	1.0	4.1	also in P	
42	C	132	-1.3	3.3	-19.5	-4.3	P in C	
43	C	72	-7.0	-4.5	-7.5	-3.2	also in P	
44	C	60	-18.0	-12.9	-33.5	-16.2		
45	C	48	-5.5	-1.6	-0.2	2.5		
46	C	33	-0.5	0.8	-2.6	1.5		
47	X	227	-121.2	-221.3	-179.8	-207.7	also in P	
48	X	177	-38.4	-31.0	53.6	-38.5	also in P	
49	X	103	-55.4	-63.0	0.9	-8.2	partly in P	
50	X	70	7.0	6.5	9.0	-9.0	partly in C	(+)
51	X	238	95.8	79.1	62.0	99.6	P in X	

6 E. coli – S. typhimurium results

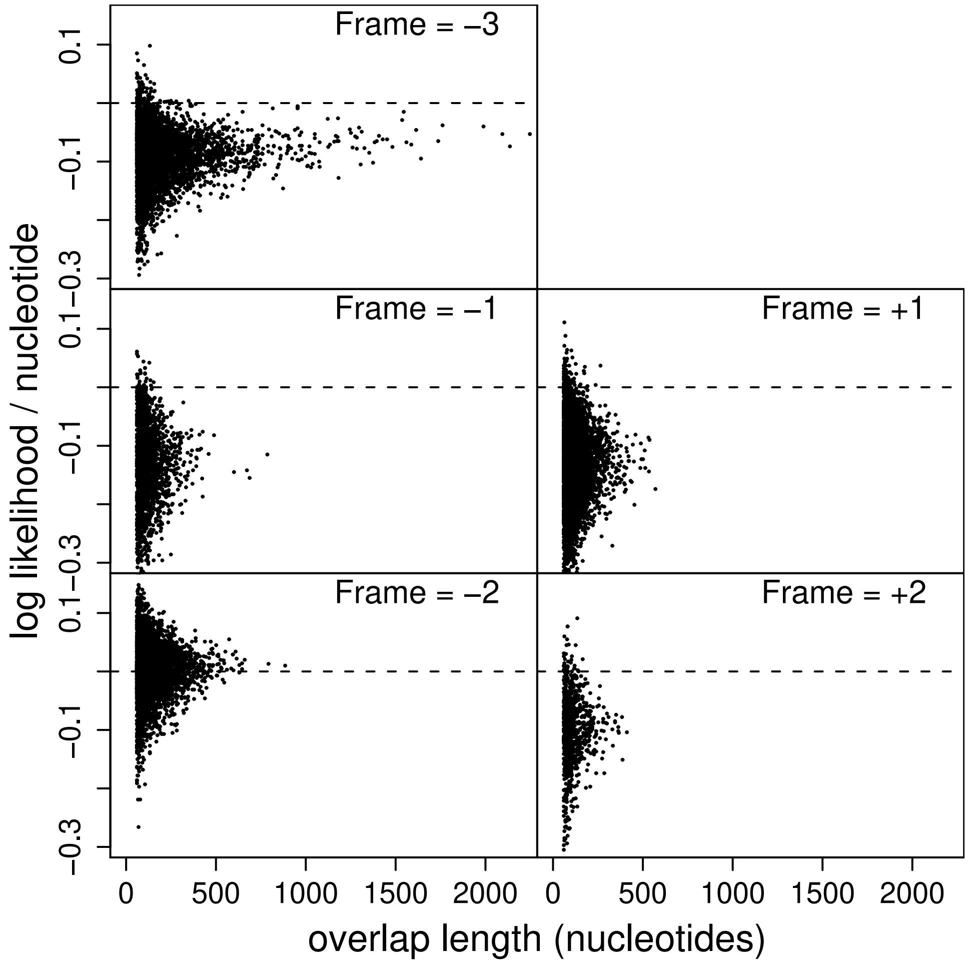


Figure 12: Plot of MLOGD scores (per nucleotide) as a function of overlap length, for the different frames, in an *E. coli* (K12, NC_000913.2, GI:49175990) – *S. typhimurium* (LT2, NC_003197.1, GI:16763390) comparison. Scores less than zero imply single-coding and are expected for the vast majority of overlapping ORFs. Except for the −2 frame, the distributions of MLOGD scores are as expected – with small tails above zero, consistent with random scatter plus a handful of true double-coding sequences. In the −2 frame, primary read-frame codon positions N_1, N_2 and N_3 align with secondary read-frame positions N_2, N_1 and N_3 respectively, constraining mutations in positions N_1 and N_2, while leaving position N_3 relatively free to mutate. This mimics the mutation constraints in single-coding regions. Hence double-coding in the −2 frame is more difficult to distinguish from single-coding, and a higher threshold needs to be set to avoid a high false positive rate. More generally, a varying threshold may be set as a function of frame and overlap length (based on the scatter present in the relevant plot) to select outlying points in the upper region of each plot for further bioinformatic and/or experimental follow-up. Note that the number of overlapping ORFs varies greatly between the different frames and, in particular, long overlapping (non-coding) ORFs occur mainly in the −3 frame.

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

Felsenstein, J., 2004, PHYLIP (Phylogeny Inference Package) version 3.6, `http://evolution.genetics.washington.edu/phylip.html`

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

Huy T. T-T. *et al.*, 2004, Genotype C of hepatitis B virus can be classified into at least two subgroups, *J. Gen. Virol.*, 85, 283–92

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