

Stochastic Analysis of Amino Acid Substitution in Protein Synthesis*

D. Bošnački¹, H.M.M. ten Eikelder¹, M.N. Steijaert¹, and E.P. de Vink²

¹ Dept. of Biomedical Engineering, Eindhoven University of Technology

² Dept. of Mathematics and Computer Science, Eindhoven University of Technology

Abstract. We present a formal analysis of amino acid replacement during mRNA translation. Building on an abstract stochastic model of arrival of tRNAs and their processing at the ribosome, we compute probabilities of the insertion of amino acids into the nascent polypeptide chain. To this end, we integrate the probabilistic model checker Prism in the Matlab environment. We construct the substitution matrix containing the probabilities of an amino acid replacing another. The resulting matrix depends on various parameters, including availability and concentration of tRNA species, as well as their assignment to individual codons. We draw a parallel with the standard mutation matrices like Dayhoff and PET91, and analyze the mutual replacement of biologically similar amino acids.

1 Introduction

The transfer of genetic information from DNA to mRNA to protein happens with very high precision. Errors can have dramatic consequences for the organism as a whole. In this paper we analyze the second stage of this information pathway—the translation from mRNA to protein, i.e., the protein biosynthesis, in the light of translation errors and the factors of potential influence.

An mRNA molecule can be considered as a string of codons, each of which codes for a specific amino acid. The codons of an mRNA molecule are sequentially read by a ribosome, where each codon is translated using an amino acid specific transfer-RNA (aa-tRNA). This way, one-by-one, a chain of amino acids, i.e. a protein is built. In this setting, aa-tRNA can be seen as molecules containing a so-called anticodon, and carrying a specific amino acid. Arriving by Brownian motion, an aa-tRNA, docks into the ribosome and may succeed in adding its amino acid to the chain under construction, or alternatively dissociates in an early or later stage of the translation. This depends on the pairing of the codon under translation with the anticodon of the aa-tRNA, as well as on the stochastic influences such as the changes in the conformation of the ribosome.

Thanks to the vast amount of research during the last thirty years, the overall process of translation is reasonably well understood from a qualitative perspective. The process can be divided into around twenty small steps/reactions, a

* Partially supported by EU FP6-project ESIGNET.

number of them being reversible. Relatively little is known exactly about the kinetics of the translation. Over the past several years, Rodnina and collaborators have measured kinetic rates for various steps in the translation process for a small number of specific codons and anticodons [1–4]. Using various advanced techniques, they were able to show that in several of those steps the rates strongly depend on the degree of matching between the codon and the anticodon. Additionally, in [5] the average concentrations (amounts) of aa-tRNAs per cell have been collected for the model organism *Escherichia coli*. Based on these results, Viljoen and co-workers started from the assumption that the rates found by Rodnina et al. can be used in general, for all codon-anticodon pairs as estimates for the reaction dynamics. In [6], a complete detailed model is presented for all 64 codons and all 48 aa-tRNA classes for *E. coli*, on which extensive Monte Carlo experiments are conducted. In particular, using the model, codon insertion times and frequencies of erroneous elongations are established. A strong correlation of the translation error and the ratio of the concentrations of the so-called near-cognate and cognate aa-tRNA species was observed. Consequently, one can argue that the competition of aa-tRNAs, rather than their availability, decides both speed and fidelity of codon translation.

In the present paper, we model the translation kinetics via the modelchecking of continuous-time Markov chains (CTMCs) using the tool Prism [7, 8]. The tool provides built-in performance analysis algorithms and a formalism (Computational Stochastic Logic, CSL) to reason about various properties of the CTMCs, removing the burden of extensive mathematical calculations from the user. Additionally, in our case, the Prism tool provides much shorter response times compared to Gillespie simulation.

We present an improvement of the stochastic model from [9], integrated in a Matlab environment. We use our model in the context of an original case study. To this end, we define the notion of a *translation substitution matrix*. The columns and rows of this matrix are labeled with amino acids. The element of the matrix indexed by amino acids a_1 and a_2 is the probability that a_1 is substituted by a_2 in the polypeptide chain. The translation substitution matrix can be used to check the error resistance capabilities of the translation process and the genetic code in general. It can also be used as an alternative similarity measure between amino acids from a point of view of translation.

The flexibility of our integrated Matlab-Prism model makes it possible to relatively easily investigate possible factors that can influence the probabilities in the substitution matrix. Here, we consider two of them:

- *Concentrations of the tRNAs*. To check this, we modify our model by assuming artificial conditions where all tRNAs have concentrations that deviate from the realistic *E. coli* model.
- *An alternative set of tRNAs*. Instead of the standard tRNAs that are confirmed by the experiments in *E. coli* we use ‘synthetic’ tRNAs assuming different number of tRNA species and their distributions over codons.

The obtained results indicate that the overall translation error is dependent on the tRNA set.

We also use the matrix to check the hypothesis that similar amino acids substitute for one another with higher probability than dissimilar ones. As a measure of similarity we use mutation data matrices, like Dayhoff [10] and PET91 [11] which are used for a similar purpose in sequence alignment tools, like BLAST. Our results confirm this hypothesis by showing that to a great extent the similarity patterns implied by the mutation data matrices emerge also in the translation substitution matrix.

Related work We did not find other work on translation substitution matrices and investigating the hypothesis that similar amino acids tend to substitute each other. The model that is used in this paper builds upon [9], which was inspired by the simulation experiments of mRNA translation reported in [6]. There, only insertion errors per codon are considered, rather than amino acid-amino acid substitution probabilities.¹ A similar model, based on ordinary differential equations, was developed in [12]. Although probabilistic, it is used to compute insertion times, but no translation errors. The model of mRNA translation in [13] assumes insertion rates that are directly proportional to the mRNA concentrations, but assigns the same probability of translation error to all codons.

There exist numerous applications of formal methods to biological systems. A selection of recent papers from modelchecking and process algebra includes [14–16]. More specifically pertaining to the current paper, [17] applies the Prism modelchecker to analyze stochastic models of signaling pathways. Their methodology is presented as a more efficient alternative to ordinary differential equations models, including properties that are not of probabilistic nature. Also, [8] employs Prism on various types of biological pathways, showing how the advanced features of the tool can be exploited to tackle large models.

Acknowledgements. We are indebted to Timo Breit, Christiaan Henkel, Erik Luit, Jasen Markovski, Tessa Pronk and Hendrik Viljoen for fruitful discussions and constructive feedback. We gratefully acknowledge the contribution of the students of the 8P135 Bioinformatics project at TU/e.

2 Biological background

Proteins are essential building blocks for the production and regulation in cellular life. In fact, proteins take care of the major part of the functioning of the cell. Proteins are produced in two stages from the genetic information carried by the DNA: From a gene at the DNA, several copies of mRNA can be generated involving RNA-polymerases. Subsequently, from each mRNA, many identical proteins can be produced with the help of a ribosome. In the present case study, we will focus on the latter aspect of expression, the generation of proteins from mRNA, the process generally referred to as translation.

In effect, proteins are long, typically folded, strains composed from amino acids. Grossly, there are only twenty different types of amino acid present in living material. On the other hand, a string of mRNA can be interpreted as a

¹ The implementation of [6] was not available to us, impeding a direct comparison with our implementation. However, the overall insertion error per codon is comparable.

sequence of nucleotides, molecules out of four types only, A, C, G and U, short for adenine, cytosine, guanine and uracil. Each three consecutive nucleotides form a codon. So, codons are essentially triplets of nucleotides. One type of codon prescribes exactly one type of amino acid, but not vice versa. Generally, in an organism, several codons may code for the same amino acid. The correspondence of codons and amino acids is the same for all organisms but for a few exceptions, and is called the genetic code. So, an mRNA, as sequence of codons, specifies precisely a protein, as sequence of amino acids.

Basically, the translation of an mRNA into a protein takes effect as follows: A ribosome attaches to the mRNA. Next, the codons of the mRNA are processed one-by-one, stepwise building up a chain of amino acids, the protein in nascent. The amino acids used are brought to the mRNA-ribosome complex by aa-tRNA, tRNA charged with an amino acid. Characteristic for an aa-tRNA is a specific triplet of nucleotides, called the anticodon. It turns out that the anticodon decides which amino acid can be charged at the tRNA. As for codons, there is exactly one amino acid that corresponds to an anticodon. Now, an aa-tRNA arriving at the mRNA-ribosome complex, docks into the A-site of the ribosome. The aa-tRNA may either (i) immediately dissociate during the initial binding or codon recognition phase, (ii) be rejected after reconfiguration of the elongation factor *Tu* (*EF-Tu*) or (iii) successfully finish the second translocation phase and have its amino acid added to the protein under construction. The particular codon of the mRNA is then considered to be processed; the mRNA-ribosome complex shifts one position with a new codon for translation, if available.

The binding of a codon and an anticodon differs from pair to pair. Given a codon, we distinguish between cognate, near-cognate and non-cognate aa-tRNA, dependent on the match of the codon and the anticodon of the particular aa-tRNA. For a cognate aa-tRNA the binding of the codon and anticodon is strong, for near-cognate the binding is less strong, for a non-cognate the binding is weak. In fact, the codon-anticodon binding influences the success (going through none, one or two phases) and the actual speed of a translation attempt.

Next, we give a brief overview of the individual steps of the translation mechanism. Our explanation is based on [2, 18, 19].

An aa-tRNA arrives at the ribosome-mRNA complex in a ternary complexation with *EF-Tu* and *GTP* with a rate determined by the interaction of *EF-Tu* and the ribosome [18]. The initial binding is relatively weak. Codon recognition comprises (i) the establishing of contact between the anticodon of the aa-tRNA and the current codon in the ribosome-mRNA complex, and (ii) subsequent conformational changes of the ribosome, that are different for cognate and near-cognates. The overall rates are similar for cognates and near-cognates. Note that, non-cognates are not selected during codon recognition. *GTPase*-activation of the elongation factor is largely favored by the conformational changes in the ribosome induced by a cognate aa-tRNA, while for near-cognate aa-tRNA *GTPase*-activation is lessened [20, 4]. During *GTP*-hydrolysis that takes place next, inorganic phosphate P_i and *GDP* are produced. It is assumed that P_i is released,

EF-Tu reconfigures and that *aa-tRNA* dissociates from the complex with *EF-Tu* and *GDP*, see [21]. The accommodation step that follows happens rapidly for cognate *aa-tRNA*, whereas for near-cognate *aa-tRNA* this proceeds slower and the *aa-tRNA* is likely to be rejected, also because of the instability of the binding to the ribosome-*mRNA* complex.

The translocation phase that follows, is unidirectional except for its reversible first step involving the elongation factor *EF-G*. In short, during the translocation phase, another GTP-hydrolysis catalyzed by *EF-G*, produces *GDP* and P_i and results in unlocking and movement of the aa-tRNA to the P-site of the ribosome. The latter step is preceded or followed by P_i -release. Translocation of the ribosome, with dramatic shifts in the positioning of ribosomal subunits, and release of *EF-G* moves the tRNA, that has transferred its amino acid to the polypeptide chain, into the E-site of the ribosome. Further rotation eventually leads to dissociation of the used tRNA.

3 Abstract model

In this section, we present an abstract model of the translation mechanism sketched in the previous section. Our aim is to capture various combined steps by a probabilistic automaton. The grouping of multiple steps results in a limited number of states and, subsequently, in a smoother analysis and quicker response times for the Prism experiments. Figure 1 depicts the probabilistic automaton obtained.

Given a particular codon under translation at the ribosome, we distinguish the following states:

- State 1, initial binding. An aa-tRNA binds, in an arrival process, at the mRNA-ribosome complex.
- State 2, recognition. The weak binding of aa-tRNA at the complex either stabilizes (transition to state 3), or the binding breaks and the tRNA dissociates (transition to state 0).
- State 3, conformation. Again, one out of two options may happen. A number of steps related to the processing of GTP may take place (modeled by the transition to state 4). Alternatively, the binding of the aa-tRNA at the mRNA-ribosome complex may lose strength (modeled by the transition back from state 3 to state 2).
- State 4, proofreading. The aa-tRNA can either be rejected, resulting in a dissociation from the mRNA-ribosome complex (transition to state 5), or the aa-tRNA shifts to P-site of the ribosome.
- State 6, accommodation. A reversible reaction involving the elongation factor *EF-G*, prepares for translocation (transition to state 7).
- State 7, translocation. Translocation may take place, as well as a number of other unidirectional steps in the translation process, and the amino acid is successfully added to the peptidyl chain (transition to state 8), or the binding of *EF-G* does not lead to reconfiguration of the ribosome (transition back from state 7 to state 6).

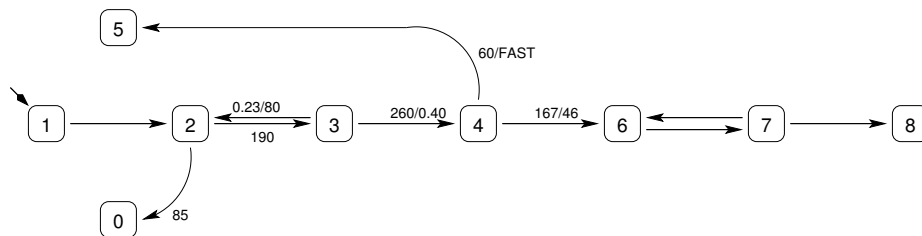


Fig. 1. Abstract automaton representing translation.

Further, we have a number of auxiliary states, that do not have a concrete, biological counterpart. However, the states are introduced for modelchecking purposes, to discriminate between the various ‘exit modes’ of an aa-tRNA.

- State 0, dissociation. The initial binding of the aa-tRNA and the mRNA-ribosome complex does not stabilize and the aa-tRNA floats away.
- State 5, rejection. The aa-tRNA dissociates from the mRNA-ribosome complex while being transferred from the A-site to the P-site of the ribosome.
- State 8, elongation. The amino acid carried by the aa-tRNA is added to the peptide chain. The uncharged tRNA flows back into the cytosol. (aa-tRNA synthesis, the recharging of a tRNA with an amino acid is not modeled here.)

The choices in the abstract automaton of Figure 1 can all be considered to be probabilistic. Of relevance are those in states 2, 3 and 4. In states 1 and 6 the choice is degenerated, in auxiliary states 0, 5 and 8 it does not occur. The probabilistic choice in state 7 does not influence the eventual exit state. In the past decade, Rodnina and co-workers have collected various kinetic parameters of a number of steps in the peptidyl transfer phase of the translation mechanism [1–4]. Additionally, using advanced fluorescent and crystallographic techniques, they showed that the binding of the codon under translation, on the one hand, and the anticodon of the aa-tRNA that has attached to the mRNA-ribosome complex, on the other hand, decisively influences the success or failure of some of the steps. Their rates are incorporated in the automaton, that can be interpreted as a continuous-time Markov chain.²

In state 2, the transition to state 3 has rate 190 whereas the transition to state 0 is taken with rate 85. These two rates are equal for cognate and near-cognate aa-tRNA alike. The transition back from state 3 to state 2 has rate 0.23 for cognate aa-tRNA against 80 for near-cognate aa-tRNA. Conversely, the transition forward from state 3 to state 4 is of rate 260 for cognate aa-tRNA and of rate 0.40 for near-cognate aa-tRNA. Note the difference between the two classes of aa-tRNA. A similar phenomenon can be observed for state 4. A transition from state 4 to state 6 of rate 167 and to state 5 of rate 60 for cognate aa-tRNA, compared to a rate 46 from state 4 to state 6 for near-cognate aa-tRNA and a fast rate (chosen to be 1000 in our experiments) for the alternative transition to state 5.

² All rates in this paper are of dimension s^{-1} .

Clearly, based on the rates provided, the probabilities of ending up in state 8 differ significantly for cognate aa-tRNA compared to near-cognate aa-tRNA. Non-cognate aa-tRNA probabilistic choices do not apply, as stable association to the mRNA-ribosome complex is considered to be negligible.

The substitution error for a codon is the probability that another amino acid is added to the nascent protein than is coded for by the codon under translation. The average insertion time is the expected time it takes for any amino acid to be added for the particular codon. The model given above has been used previously in [9] to analyze both the substitution error for a codon and its average insertion time. To this end, the automaton of Figure 1 is combined with an arrival process of cognate, near-cognates and non-cognate aa-tRNAs, the separation in classes of aa-tRNA depending on the active codon. In the current paper, we refine the analysis of the substitution error, by considering the actual amino acid that is inserted.

We define $P(\mathbf{aa}|\mathbf{cd})$ as the probability for an elongation of the peptidyl chain with an amino acid \mathbf{aa} , given a codon \mathbf{cd} . Let \mathbf{cd} code for amino acid \mathbf{bb} . Then, we distinguish five categories of aa-tRNA: \mathbf{xx} -cognate, \mathbf{yy} -cognate, \mathbf{xx} -near-cognate, \mathbf{yy} -near-cognate and non-cognate aa-tRNA. A cognate aa-tRNA is classified as an \mathbf{xx} -cognate if it carries the amino acid \mathbf{aa} of interest (i.e. if \mathbf{aa} equals \mathbf{bb}); the cognate aa-tRNA is considered an \mathbf{yy} -cognate if it carries an amino acid different from the amino acid \mathbf{aa} (i.e. if we are interested in an amino acid \mathbf{aa} that is different from the amino acid \mathbf{bb} coded for by \mathbf{cd}). Similarly, a near-cognate aa-tRNA is referred to as an \mathbf{xx} -near-cognate if its amino acid is \mathbf{aa} , and is called an \mathbf{yy} -near-cognate if its amino acid is different from \mathbf{aa} . All aa-tRNA that do not belong to any of the other classes are considered non-cognates. Thus, the selected amino acid determines the \mathbf{xx} -type or \mathbf{yy} -type for a cognate or near-cognate aa-tRNA; the active codon decides whether an aa-tRNA is cognate, near-cognate or non-cognate. In our model, non-cognates never lead to insertion of an amino acid. The binding of a non-cognate anticodon and the codon under translation is considered too weak for the tRNA to move to state 3. Non-cognate aa-tRNA will always exit at state 0.

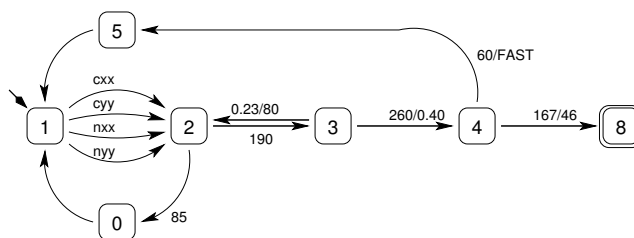


Fig. 2. Adapted automaton.

Given the above definitions of aa-tRNA, we end up with the automaton as given in Figure 2. The rates \mathbf{cxx} , \mathbf{cyy} , \mathbf{nxx} and \mathbf{nyy} depend on the amount of the aa-tRNAs that are classified as \mathbf{xx} -cognate, \mathbf{yy} -cognate, \mathbf{xx} -near-cognate and

yy-near-cognate, respectively, for the considered codon and amino acid. The transitions from state 0 and state 5 to state 1 are added, since the process continues as long as state 8 has not been reached, i.e. until an amino acid has been transferred successfully. States 6 to 8 have been identified, as from state 6 the final state 8 will always be reached eventually. The rates have been kept deliberately, although they could have been replaced by probabilities. In the set-up of our experiments, it comes in handy to deal with CTMCs to which we can feed numbers of aa-tRNAs, avoiding to calculate their relative fractions. An overview of the reactions involved are collected in Table 1.

$C + R1$	$\xrightleftharpoons[85]{c_{xx}/c_{yy}}$	$CR2$	$\xrightleftharpoons[190]{0.23}$	$CR3$	$\xrightarrow{260}$	$CR4$
$CR4$	$\xrightarrow{167}$	$CR8$		$CR4$	$\xrightarrow{60}$	$C + R1$
$N + R1$	$\xrightleftharpoons[85]{n_{xx}/n_{yy}}$	$NR2$	$\xrightleftharpoons[190]{80}$	$NR3$	$\xrightarrow{0.40}$	$NR4$
$NR4$	$\xrightarrow{46}$	$NR8$		$NR4$	\xrightarrow{FAST}	$NN + R1$

Table 1. Molecular reactions underlying the adapted model.

4 Amino acid substitution

In this section, we discuss how the amino acid-codon insertion probability matrix AC and the amino acid-amino acid translation substitution matrix TS are computed for the *Escherichia coli* bacterium. The abstract model of the previous section is, per codon, supplemented with relevant concentrations of the various types of aa-tRNAs, to establish the probability for an amino acid to be inserted with the particular codon being active. When this information is available for all codons, by proper grouping, the probability for an amino acid to replace the amino acid that is actually coded for, can be obtained.

To calculate $P(\mathbf{aa}|\mathbf{cd})$, the probability of elongation of the growing protein chain with the amino acid \mathbf{aa} , when codon \mathbf{cd} is under translation, we proceed as follows: We provide the Prism modelchecker with four input parameters, viz. `xx_cogn`, `yy_cogn`, `xx_near` and `yy_near`, each representing the amount of available cognate and near cognate tRNAs, carrying either the amino acid of interest (indicated by `xx`) or an other amino acid (indicated by `yy`). This instantiates the arrival process of an aa-tRNA at state 1 of the abstract automaton. The probabilities for success, i.e. reaching state 8, and failure, dissociation via state 0 or rejection via state 5 followed by re-activation of the arrival process at state 1, are not independent from the arrival process itself. Therefore, we have to consider the CTMC simultaneous representing the arrival process and translation

mechanism. With this done in Prism, we only need to establish the CSL formula

$$P=? [(s!=0 \ \& \ s!=5) \ U \ (s=8)] .$$

See Appendix B for the complete Prism code.

In order to facilitate the construction of the insertion probability matrix, we have combined the Prism modelchecker with Matlab. We make use of Matlab’s extern call mechanism and Prism’s command line option for parameter instantiation. In two nested loops, the outer loop iterating over codons, the inner loop iterating over amino acids, we call from within Matlab the Prism tool, for example by

```
prism ourmodel.sm ourformula.csl -const xx_cogn=1037, ...
```

communicating the values of parameter `xx_cogn` and others to the modelchecker. The negligible value $1.0e-6$ is used instead of 0, as only positive values are allowed as rates of an exponential distribution. Apart from directory management, the interaction of the two tools quite conveniently suited our purposes.

For the usual *E. coli* aa-tRNA set with standard concentrations, referred to as *real*, we illustrate how the parameter values are obtained for the codon UUU and amino acids Phe, Leu, and Ile.

codon	amino acid	xx_cogn	yy_cogn	xx_near	yy_near
UUU	Phe	1037	0	0	2944
UUU	Leu	0	1037	2944	0
UUU	Ile	0	1037	0	2944

Table 2. Some input values for the realistic model.

In case of calculating $P(\text{Phe}|\text{UUU})$, where the amino acid under consideration coincides with the amino acid of the codon, we assign to `xx_cogn` the total amount of cognate tRNAs. From Table 7 in Appendix A, we see that tRNA 28 is the only cognate, so `xx_cogn` is set to 1037, the number of tRNA 28 in Table 8. In this case, there are no cognates coding for an amino acid other than Phe, hence `yy_cogn` = 0. Next, we check the near-cognates of UUU. According to Table 7, only tRNAs 22 and 23 act as near-cognates. Both code for Leu, the amino acid leucine. Therefore, we put `xx_near` = 0, as no near-cognate codes for Phe and `yy_near` = 1913+1031 = 2944 the sum of all counts of near-cognates coding for an amino acid different from Phe. See the first row of Table 2.

To establish $P(\text{Leu}|\text{UUU})$, we put `xx_cogn` to zero (or rather 10^{-6}), as there are no cognates of the codon UUU carrying Leu, and `yy_cogn` to 1037, the number of molecules of the non-Leu cognate aa-tRNA 28. Since, the near-cognates aa-tRNAs 22 and 23 both carry Leu, `xx_near` is set to 2944, the sum of their counts, and `yy_near` is zero as there are no near-cognate with other amino acids.

To compute $P(\text{Ile}|\text{UUU})$, the substitution probability for the amino acid isoleucine with respect to the codon UUU, we have that `xx_cogn` and `xx_near`

are both nihil. No cognate or near-cognate aa-tRNAs will insert Ile, hence the substitution probability will be zero. Because of this, no further calculation is required. The Matlab script automatically puts $P(\text{Ile}|\text{UUU}) = 0$.

Having the amino acid-codon insertion matrix $AC = (P(\text{aa}|\text{cd}))_{\text{aa},\text{cd}}$ available, we derive the amino acid-amino acid translation substitution matrix TS . Each item $TS_{\text{aa},\text{bb}}$ of TS denotes the probability that amino acid bb is inserted while the current translation codes for the amino acid aa . To compensate for the differences of occurrence in the *E. coli* genome of the various codons of an amino acid, we balance the sum of amino acid-codon insertion probabilities, by the relative frequencies of the codons:

$$TS_{\text{aa},\text{bb}} = \sum_{\text{cd} \in \text{codons}(\text{aa})} rf(\text{cd}, \text{aa}) \cdot P(\text{aa}|\text{cd})$$

with $\text{codons}(\text{aa})$ the set of codons coding for the amino acid aa according to the genetic code, and $rf(\text{cd}, \text{aa})$ the relative frequency of the codon cd with respect to $\text{codons}(\text{aa})$.

5 Alternative aa-tRNA sets

We investigate how substitution probabilities are affected by the aa-tRNA population. We consider both the effect of the concentration of aa-tRNA molecules of a specific species, and the influence of the composition of the aa-tRNA set. Our starting point is the *E. coli* model from the previous sections with an aa-tRNA set containing 48 different tRNA species and numbers of molecules from each species as reported in [5]. Alternative models, with an aa-tRNA set different from usual and with concentrations deviating from standard values, are examined as well. The experimental set-up is rather flexible in this respect, only different aa-tRNA populations with other amounts of available molecules need to be supplied. We consider eight different models, listed in Table 3.

Name	species	aa-tRNA counts
Model <i>48R</i>	48	based on Table 8
Model <i>48F</i>	48	1000 per aa-tRNA
Model <i>48C</i>	48	1000 for per codon recognized
Model <i>64R</i>	64	based on Table 8
Model <i>64FC</i>	64	1000 per aa-tRNA / codon
Model <i>25R</i>	25	based on Table 8
Model <i>25F</i>	25	1000 per aa-tRNA
Model <i>25C</i>	25	1000 per codon recognized

Table 3. Alternative models: aa-tRNA sets of 48, 64 or 25 species, aa-tRNA concentrations based on real measurements, flat, or proportional to codon matching.

48 species aa-tRNA set

Model $48R$, the so-called realistic model with parameters based on [6, 5], has 48 aa-tRNA species and molecule counts based on physical measurements. In order to assay the stability of amino acid substitution, we have run similar experiments with varying parameters. In one model, we use for each tRNA species the same amount of molecules, arbitrarily chosen as 1000. This model with flat aa-tRNA concentrations is referred to as $48F$, with F for flat. Another variation is a model in which we assign the equal amounts (again 1000 molecules) to each codon. If the same codon is recognized by multiple aa-tRNAs, each of them is assigned a proportional part. So, the count of 1000 is equally split over the number of cognate tRNA species. If an aa-tRNA is cognate to several codons, it will be allotted accordingly. We refer this model as $48C$, with C for codon. Note that, for models $48F$ and $48C$, the arbitrarily chosen value of 1000 does not influence the outcome, as the error probabilities are determined by the fractions of cognate and near-cognate species and not by the values themselves.

64 species aa-tRNA set

Apart from variations in the concentrations of aa-tRNAs, one can also modify, in silico, the sets of tRNA species. An obvious choice, is the model with the maximal number of 64 aa-tRNA species. In this model, each of aa-tRNA is considered to recognize exactly one codon. Thus, under this assumption, each codon has exactly one cognate tRNA and nine near-cognate aa-tRNA.

For the model $64R$, the count for each aa-tRNA species is equal to the sum of tRNA species in the original model that recognize the corresponding codon. For aa-tRNAs in the original model that recognize more than one codon, the new aa-tRNAs get an equal share of the original amount. Analogously to the models with a 48 species aa-tRNA set, we also define flat and cognate models for the 64 aa-tRNA case, with equal amounts of molecules per tRNA species and per codon. However, for this specific case these two models coincide, each with an amount of 1000 molecules for each of the 64 aa-tRNA species. Consequently, the model is named $64FC$.

25 species aa-tRNA set

The other obvious choice of aa-tRNA set, is the opposite case, where the number of species is minimal. However, instead of the theoretical minimum of aa-tRNA species, where the choice of cognates seems arbitrarily, we decide to have exactly one tRNA for each ‘genomic box or block’, i.e. a group of codons that codes for the same amino acid and only differ in the third nucleotide. This model contains 25 aa-tRNA species, reminiscent to the fine-tuned genetic code in eukaryote mitochondria. As before, $25F$ denotes the flat model variant, each tRNA species having 1000 molecules. For $25R$, the amount aa-tRNA is set to the total of the tRNAs in the $48R$ model that belong to the block. The one exception being release factor RF1, for which we assign the full amount to the block UAA, UAG

although it recognizes both UAA and UAG (which belong to separate blocks). Finally, for the model *64C*, the amount of molecules for each aa-tRNA species is calculated as 1000 times the number of codons in the corresponding block. See Table 4.

aa-tRNA	recognized codons	aa-tRNA	recognized codons
1 (Phe)	UUU, UUC	14 (Gln)	CAA, CAG
2 (Leu)	UUA, UUG	15 (Asn)	AAU, AAC
3 (Leu)	CUU, CUC, CUA, CUG	16 (Lys)	AAA, AAG
4 (Ile)	AUU, AUC, AUA	17 (Asp)	GAU, GAC
5 (Met)	AUG	18 (Glu)	GAA, GAG
6 (Val)	GUU, GUC, GUA, GUG	19 (Cys)	UGU, UGC
7 (Ser)	UCU, UCC, UCA, UCG	20 (End)	UGA
8 (Pro)	CCU, CCC, CCA, CCG	21 (Trp)	UGG
9 (Thr)	ACU, ACC, ACA, ACG	22 (Arg)	CGU, CGC, CGA, CGG
10 (Ala)	GCU, GCC, GCA, GCG	23 (Ser)	AGU, AGC
11 (Tyr)	UAU, UAC	24 (Arg)	AGA, AGG
12 (End)	UAA, UAG	25 (Gly)	GGU, GGC, GGA, GGG
13 (His)	CAU, CAC		

Table 4. The 25 species aa-tRNA set.

For all the above models we have computed the substitution probability matrix as sketched in Section 4. The diagonal of the matrices denotes the probability for peptide elongation with the amino acid that was coded for. The average substitution error for the model, shown in Table 5, is obtained by taking the average over all amino acid for the probability of a substitution by a non-coded amino acid. The table displays in parentheses a weighted average of errors, obtained by scaling the errors for individual amino acids with the relative occurrence of their codons in the *E. coli* genome. Remarkably, the errors for the 48 tRNA models are always smaller than those of the synthesized 25 and 64 tRNA models. The errors for individual amino-acids are shown in Figure 3.

model	‘real’	‘flat’	‘codon’
48 aa-tRNA set	0.48% (0.45%)	0.45% (0.45%)	0.39% (0.36%)
64 aa-tRNA set	0.93% (0.85%)	0.73% (0.69%)	0.73% (0.69%)
25 aa-tRNA set	1.16% (0.83%)	0.66% (0.64%)	0.76% (0.68%)

Table 5. Mean substitution error and occurrence-weighted substitution error (within parentheses) for all eight models (model *64RF* twice).

Notably, the ‘real’ *48R* model has a striking low probability for errors for stop-codons. None of the other models reaches such low value. Moreover, the probability for other codons to accidentally act as a stop-codon is lowest for *48R* model as well. See the red/grey bars in Figure 4.

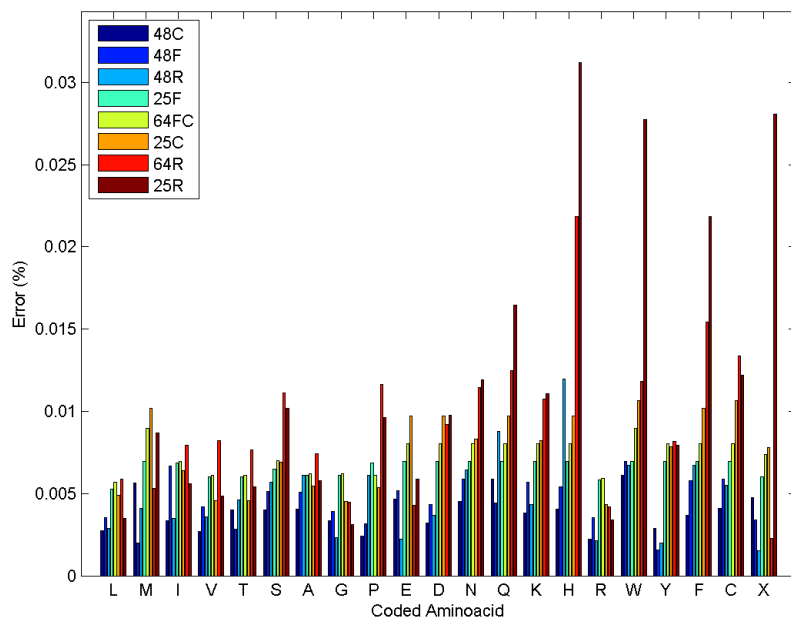


Fig. 3. Substitution errors for individual amino acids and the for stop-codon.

6 Groups of related amino acids

In this section, we check the hypothesis that biologically similar amino acids substitute for each other during translation. Its rationale is that such substitutions would lower the chance that the resulting protein is non-functional. Namely, it has been observed that if in a polypeptide chain an amino acid is substituted by another biochemically similar amino acid, then it is likely that the modified protein has biochemical properties similar to the original one. Therefore, it seems plausible to assume that, under evolutionary pressure, an error robustness mechanism has been developed also exploited at translational level.

Our measure of similarity of amino acids is based on so-called mutation data matrices, like Dayhoff [10] and PET91 [11]. Essentially, these matrices give for each pair of amino acids an evolutionary substitution probability. Mutation data matrices and their derivatives are widely used as an amino acid similarity measure, e.g. in sequence alignment tools like BLAST. At first sight it may look strange that we compare mutation errors, which introduce protein changes that are permanent for the organism, with translation errors which only affect one copy of the protein and are not inherited. Thus, it is worth emphasizing that we use the matrices only as a similarity measure ‘approved’ by the evolution without trying to draw any further analogy between these two phenomena.

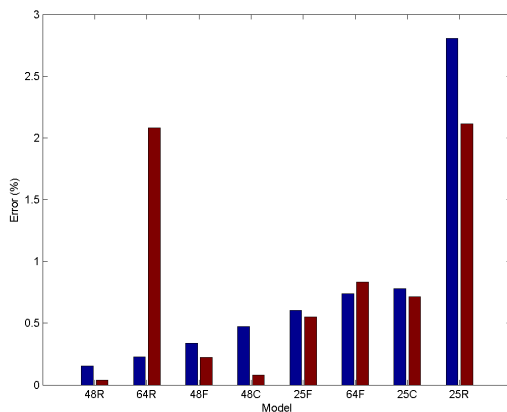


Fig. 4. Percentage of erroneous substitution of a stop-codon for an amino acid (blue/dark), and percentage of erroneous substitution of any other codon by a stop-codon (red/gray).

Based on the mutation data matrices, we divide amino acids in four groups. There are different groupings of amino acids in the literature usually based on their biochemical properties [22]. In this paper, we use the partitioning proposed by Swanson [23] which is based on mutation data matrices, but also rather well in agreement with the classifications based purely on biochemical properties.

The two main criteria for Swanson’s partitioning are the size (small vs. large) of the amino acids and their positioning in the proteins based on their affinity for water (hydrophobic/inner vs. hydrophilic/outer) The four groups are 1) the *small* group: Pro, Gly, Ala, Ser, Thr, 2) the *outer* group: Glu, Asp, Asn, Gln, Lys, His, 3) the *large* group: Arg, Trp, Tyr, Phe, and 4) the *inner* group: Cys, Val, Ile, Met, Leu. A natural cyclic arrangement of those groups can be made such that neighboring groups are small-outer, outer-large, large-inner and inner-small. Thus, the groups small and large are considered as opposite groups and the same holds for the groups inner and outer. See [23].

During translation three types of errors can be made. The least serious mis-reading results in adding to the protein an amino acid of the same group as the intended amino acid. A greater error would be a substitution with an amino acid of a neighboring group. Finally, potentially greatest consequences have substitutions with an amino acid of the opposite group. A completely different error is the misinterpretation of a codon as the stop codon, which means that the forming of the protein is terminated.

The probability GSP_{G_1, G_2} that an amino acid in group G_1 will be substituted with an amino acid from group G_2 is given by

$$GSP_{G_1, G_2} = \sum_{aa \in G_1, bb \in G_2} rf(aa, G_1) \cdot TS_{aa, bb}$$

where **aa** and **bb** are amino acids in groups G_1 and G_2 , respectively, $rf(\mathbf{aa}, G_1)$ is the relative occurrence frequency of amino acids **aa** within G_1 , and $TS_{\mathbf{aa}, \mathbf{bb}}$ is the probability that **bb** substitutes **aa**, i.e., the corresponding element of the substitution matrix TS as defined in Section 4. The relative frequency of amino acid **aa** within its group G_1 is obtained as a ratio between the sum of the occurrence frequencies of all codons of **aa** and the sum of the occurrence frequencies of all codons of the amino acids in G_1 . (Again we deal with the so-called codon bias, i.e. within the genome of an organism not all codons and amino acids are used with equal frequency.)

In Table 6 we give the above defined GSP probabilities for the realistic model. Each row gives the error probabilities for amino acids of the corresponding group. For all groups, the probability of generating the correct amino acid is more than 0.995. The probability of generating a wrong amino acid from the *correct* group is indicated in bold. Note that for groups 1, 2 and 4 this probability is larger than the probability of generating a wrong amino acid of another group. Somewhat surprisingly, this does not hold for group 3. Furthermore, the probability of an unexpected termination (stop codon) is extremely small.

	to group 1	to group 2	to group 3	to group 4	stop-codon	correct
group 1	0.22 %	0.077 %	0.047 %	0.11 %	0.0006 %	99.5 %
group 2	0.076 %	0.25 %	0.081 %	0.10 %	0 %	99.7 %
group 3	0.065 %	0.027 %	0.043 %	0.21 %	0.0019 %	99.7 %
group 4	0.069 %	0.029 %	0.055 %	0.17 %	0.0015 %	99.7 %

Table 6. Realistic model, percentage of erroneous amino acids, stop codon and the correct amino acid for amino acids from the four groups.

A graphical representation of the error probabilities is given in Figure 5. In this figure only the error probabilities, except the very small probability of an erroneous stop codon, are shown. The probabilities of generating an erroneous amino acid of the original group are indicated in white. The probabilities of indicating an amino acid of the opposite group are shown in black. Clearly, the probability of generating erroneous amino acids in the correct (white) group is higher than the probability of generating amino acids in the opposite (black) group, except for group 3.

We performed analogous analyses of the other models and in all cases the outcome was similar. This indicates that the relatively small probability of out-of-group amino acid substitutions is due to the distribution of the amino acids over codons, i.e., the genetic code, rather than to the tRNA species and their concentrations.

7 Conclusions and future work

We described a formal analysis of codon misreading errors during translation of mRNA to protein, caused by a mismatch between codons and tRNAs. To this

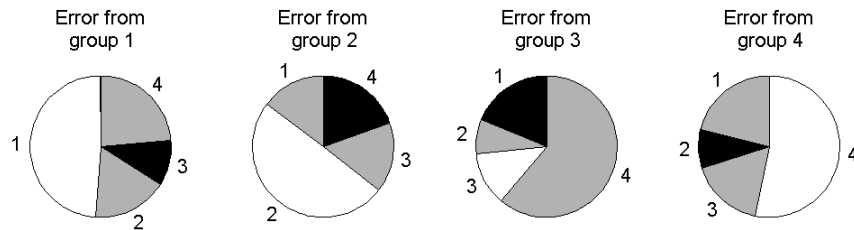


Fig. 5. Realistic model, probabilities of erroneous amino acid from another group (own group: white, opposite group: black).

end, we presented a method based on probabilistic modelchecking, in particular the modelchecker Prism in combination with Matlab.

Inspired by mutation data matrices, we introduced the notion of a translation substitution matrix. Using our model, we computed the elements of this matrix which are the probabilities that amino acids replace each other in the protein as a result of codon misreading. Further, we investigated the influence of some parameters, like tRNA concentrations and different tRNA species, to the misreading probabilities. It turned out that the translation mechanism is quite robust. The mean substitution error for the realistic model is in line with experimental findings, cf. [6]. Remarkably, for the realistic model it is smaller than for our synthesized models. We also showed that biologically similar amino acid replace each other with higher probabilities than dissimilar ones. Experiments as described in Section 5 can easily be done *in silico*, but will require substantial effort, if not impossible on such rigorous scale, in a wetlab. Additionally, our case studies confirm that probabilistic modelchecking has advantage over simulation regarding reliability and running times. Preliminary experiments indicate that our modelchecking approach is about 10 times faster than our Gillespie simulations.

In the future, we plan to apply our translation model to further investigate the robustness of the translation mechanism and the genetic code. The translation substitution belongs to a class of case studies for which the essential properties are of a probabilistic nature. It would be interesting to employ the methodology of this paper to similar problems, like the precision of DNA repair and antibody recognition.

References

1. Pape et al., T.: Complete kinetic mechanism of elongation factor Tu-dependent binding of aa-tRNA to the A-site of *E. coli*. *EMBO Journal* **17** (1998) 7490–7497
2. Rodnina, M., Wintermeyer, W.: Ribosome fidelity: tRNA discrimination, proof-reading and induced fit. *TRENDS in Biochemical Sciences* **26**(2) (2001) 124–130
3. Savelsbergh et al., A.: An elongation factor G-induced ribosome rearrangement precedes tRNA–mRNA translocation. *Molecular Cell* **11** (2003) 1517–1523
4. Gromadski, K., Rodnina, M.: Kinetic determinants of high-fidelity tRNA discrimination on the ribosome. *Molecular Cell* **13**(2) (2004) 191–200
5. Dong et al., H.: Co-variation of tRNA abundance and codon usage in *Escherichia coli* at different growth rates. *Journal of Molecular Biology* **260** (1996) 649–663
6. Fluitt et al., A.: Ribosome kinetics and aa-tRNA competition determine rate and fidelity of peptide synthesis. *Computational Biology and Chemistry* **31** (2007) 335–346
7. Kwiatkowska et al., M.: Probabilistic symbolic model checking with Prism: a hybrid approach. *Journal on Software Tools for Technology Transfer* **6** (2004) 128–142 See also <http://www.prismmodelchecker.org/>.
8. Heath et al., J.: Probabilistic model checking of complex biological pathways. In: *Proc. CMSB 2006, LNBI 4210* (2006) 32–47
9. Bosnacki et al., D.: *In Silico* modelling and analysis of ribosome kinetics and aa-trna competition. In: *Proc. Computational Models for Cell Processes, Turku Centre for Computer Science, Åbo Academia, Turku* (2008) 16pp.
10. Dayhoff, M.: Supplements 1, 2 and 3. *Atlas of Protein Sequence and Structure* **5** (1978)
11. Jones et al., D.: The rapid generation of mutation data matrices from protein sequences. *CABIOS* **3** (1992) 275–282
12. Heyd, A., Drew, D.: A mathematical model for elongation of a peptide chain. *Bulletin of Mathematical Biology* **65** (2003) 1095–1109
13. Gilchrist, M., Wagner, A.: A model of protein translation including codon bias, nonsense errors, and ribosome recycling. *Journal of Theoretical Biology* **239** (2006) 417–434
14. Priami et al., C.: Application of a stochastic name-passing calculus to represent action and simulation of molecular processes. *Information Processing Letters* **80** (2001) 25–31
15. Chabrier, N., Fages, F.: Symbolic model checking of biochemical networks. In: *Proc. CMSB 2003, LNCS 2602* (2003) 149–162
16. Danos et al., V.: Rule-based modelling of cellular signalling. In: *Proc. CONCUR, LNCS 4703* (2007) 17–41
17. Calder et al., M.: Analysis of signalling pathways using continuous time Markov chains. In: *Transactions on Computational Systems Biology VI. LNBI 4220* (2006) 44–67
18. Rodnina et al., M.: Recognition and selection of tRNA in translation. *FEBS Letters* **579** (2005) 938–942
19. Karp, G.: *Cell and Molecular Biology*, 5th ed. Wiley (2008)
20. Rodnina et al., M.: Codon-dependent conformational change of elongation factor Tu preceding GTP hydrolysis on the ribosome. *EMBO Journal* **14** (1995) 2613–2619
21. Knudsen et al., C.: The importance of structural transitions of the switch II region for the functions of elongation factor Tu on the ribosome. *Journal of Biological Chemistry* **276** (2001) 22183–22190

22. Taylor, W.: The classification of amino acid conservation. *Journal of Theoretical Biology* **119** (1986) 205–218
23. Swanson, R.: A unifying concept for the amino acid code. *Bulletin of Mathematical Biology* **46**(2) (1984) 187–203

A Additional tables

Codon	Cognates	Near-Cognates	Codon	Cognates	Near-Cognates
UUU	28	22,23	GUU	44,45,46	
UUC	28	9,17,20,22,23,36,42,43,45,46	GUC	45,46	2,8,15,17,20,28,44
UUG	22,23	18,19,25,26,27,28,34,41	GUG	44	13,18,19,22,25,26,27,45,46
UUA	23	21,22,28,32,33,44	GUA	44	1,12,14,21,23,45,46
UCU	33,36	34	GCU	1	2
UCC	36	2,9,28,30,33,34,37,39,42,43	GCC	2	1,8,15,30,36,37,39,45,46
UCG	33,34	22,29,36,38,41	GCG	1	2,13,29,34,38
UCA	33	1,23,31,32,34,36,40	GCA	1	2,12,14,31,33,40,44
UGU	9	3,32,41	GGU	15	3,13,14
UGC	9	15,28,32,35,36,41,42,43	GGC	15	2,8,9,13,14,35,45,46
UGG	41	4,6,9,13,22,32,34	GGG	13,14	4,6,15,41
UGA	32,48	5,9,14,23,33,41	GGA	14	1,5,12,13,15,32,44
UAU	42,43		GAU	8	12
UAC	42,43	7,8,9,16,28,36	GAC	8	2,7,12,15,16,42,43,45,46
UAG	47	11,22,34,41,42,43	GAG	12	8,11,13
UAA	47,48	10,12,23,24,32,33,42,43	GAA	12	1,8,10,14,24,44
CUU	20	3,19,21	AUU	17	18,25,26,27
CUC	20	16,17,19,21,28,30,45,46	AUC	17	7,18,20,25,26,27,28,35,37,39,45,46
CUG	19,21	4,11,18,20,22,25,26,27,29	AUG	25,26,27	6,17,18,19,22,38
CUA	21	10,19,20,23,31,44	AUA	18	5,17,21,23,24,25,26,27,40,44
CCU	30,31	3,29	ACU	37,39,40	38
CCC	30	2,16,20,29,31,36,37,39	ACC	37,39	2,7,17,30,35,36,38,40
CCG	29,31	4,11,19,30,34,38	ACG	38,40	6,18,25,26,27,29,34,37,39
CCA	31	1,10,21,29,30,33,40	ACA	40	1,5,24,31,33,37,38,39
CGU	3	4	AGU	35	3,5,6
CGC	3	4,9,15,16,20,30,35	AGC	35	5,6,7,9,15,17,37,39
CGG	4	3,6,11,13,19,29,41	AGG	6	4,5,13,18,25,26,27,35,38,41
CGA	3	4,5,10,14,21,31,32	AGA	5	6,14,24,32,35,40
CAU	16	3,10,11	AAU	7	24
CAC	16	7,8,10,11,20,30,42,43	AAC	7	8,16,17,24,35,37,39,42,43
CAG	11	4,10,16,19,29	AAG	24	6,7,11,18,25,26,27,38
CAA	10	11,12,16,21,24,31	AAA	24	5,7,10,12,40

Table 7. Codons and their cognate and near-conate tRNAs. Derived from Table 7.

	tRNA	Amino acid	Anticodon	Recognized Codons	Molecules/cell
1	Ala1	A	UGC	GCU, GCA, GCG	3250
2	Ala2	A	GGC	GCC	617
3	Arg2	R	ACG	CGU, CGC, CGA	4752
4	Arg3	R	CCG	CGG	639
5	Arg4	R	UCU	AGA	867
6	Arg5	R	CCU	AGG	420
7	Asn	N	GUU	AAC, AAU	1193
8	Asp1	D	GUC	GAC, GAU	2396
9	Cys	C	GCA	UGC, UGU	1587
10	Gln1	Q	UUG	CAA	764
11	Gln2	Q	CUG	CAG	881
12	Glu2	E	UUC	GAA, GAG	4717
13	Gly1	G	CCC	GGG	1068.5
14	Gly2	G	UCC	GGA, GGG	1068.5
15	Gly3	G	GCC	GGC, GGU	4359
16	His	H	GUG	CAC, CAU	639
17	Ile1	I	GAU	AUC, AUU	1737
18	Ile2	I	CAU	AUA	1737
19	Leu1	L	CAG	CUG	4470
20	Leu2	L	GAG	CUC, CUU	943
21	Leu3	L	UAG	CUA, CUG	666
22	Leu4	L	CAA	UUG	1913
23	Leu5	L	UAA	UUA, UUG	1031
24	Lys	K	UUU	AAA, AAG	1924
25	Met f1	M	CAU	AUG	1211
26	Met f2	M	CAU	AUG	715
27	Met m	M	CAU	AUG	706
28	Phe	F	GAA	UUC, UUU	1037
29	Pro1	P	CGG	CCG	900
30	Pro2	P	GGG	CCC, CCU	720
31	Pro3	P	UGG	CCA, CCU, CCG	581
32	Sec	X	UCA	UGA	219
33	Ser1	S	UGA	UCA, UCU, UCG	1296
34	Ser2	S	CGA	UCG	344
35	Ser3	S	GCU	AGC, AGU	1408
36	Ser5	S	GGA	UCC, UCU	764
37	Thr1	T	GGU	ACC, ACU	104
38	Thr2	T	CGU	ACG	541
39	Thr3	T	GGU	ACC, ACU	1095
40	Thr4	T	UGU	ACA, ACU, ACG	916
41	Trp	W	CCA	UGG	943
42	Tyr1	Y	GUA	UAC, UAU	769
43	Tyr2	Y	GUA	UAC, UAU	1261
44	Val1	V	UAC	GUA, GUG, GUU	3840
45	Val2A	V	GAC	GUC, GUU	630
46	Val2B	V	GAC	GUC, GUU	635
47	RF1	X		UAA, UAG	1200
48	RF2	X		UAA, UGA	6000

Table 8. tRNA species in *E. coli*, data from [5] and [6].

B Prism code

```

stochastic
// constants
const double ONE=1;
const double FAST=1000;

// tRNA rates, precalculated
const double c_xx_cogn ;
const double c_yy_cogn ;
const double c_xx_near ;
const double c_yy_near ;
const double c_nonc ;

const double k1f = 140;
const double k2b = 85;
const double k2bx=2000;
const double k2f = 190;
const double k3bc= 0.23;
const double k3bn= 80;
const double k3fc= 260;
const double k3fn= 0.40;
const double k4rc= 60;
const double k4rn=FAST;
const double k4fc= 166.7;
const double k4fn= 46.1;
const double k6f = 150;
const double k7b = 140;
const double k7f = 145.8;

module ribosome
s_rib : [0..8] init 1 ;
cogn : bool init false ;
near : bool init false ;
nonc : bool init false ;
xx : bool init false ;

// initial binding
[ ] (s_rib=1) -> k1f * c_xx_cogn : (s_rib'=2) & (xx'=true) & (cogn'=true) ;
[ ] (s_rib=1) -> k1f * c_yy_cogn : (s_rib'=2) & (cogn'=true) ;
[ ] (s_rib=1) -> k1f * c_xx_near : (s_rib'=2) & (xx'=true) & (near'=true) ;
[ ] (s_rib=1) -> k1f * c_yy_near : (s_rib'=2) & (near'=true) ;
[ ] (s_rib=1) -> k1f * c_nonc : (s_rib'=2) & (nonc'=true) ;
[ ] (s_rib=2) & ( cogn | near ) -> k2b :
    (s_rib'=0) & (cogn'=false) & (near'=false) & (xx'=false) ;
[ ] (s_rib=2) & nonc -> k2bx : (s_rib'=0) & (nonc'=false) ;

// codon recognition
[ ] (s_rib=2) & ( cogn | near ) -> k2f : (s_rib'=3) ;
[ ] (s_rib=3) & cogn -> k3bc : (s_rib'=2) ;
[ ] (s_rib=3) & near -> k3bn : (s_rib'=2) ;

// GTPase activation, GTP hydrolysis, EF-Tu conformation change
[ ] (s_rib=3) & cogn -> k3fc : (s_rib'=4) ;
[ ] (s_rib=3) & near -> k3fn : (s_rib'=4) ;

// rejection
[ ] (s_rib=4) & cogn -> k4rc : (s_rib'=5) & (cogn'=false) & (xx'=false);
[ ] (s_rib=4) & near -> k4rn : (s_rib'=5) & (near'=false) & (xx'=false);

// accommodation, peptidyl transfer
[ ] (s_rib=4) & cogn -> k4fc : (s_rib'=6) ;
[ ] (s_rib=4) & near -> k4fn : (s_rib'=6) ;

// EF-G binding
[ ] (s_rib=6) -> k6f : (s_rib'=7) ;
[ ] (s_rib=7) -> k7b : (s_rib'=6) ;

// GTP hydrolysis, unlocking, tRNA movement and Pi release,
// rearrangements of ribosome and EF-G, dissociation of GDP
[ ] (s_rib=7) -> k7f : (s_rib'=8) ;

// no entrance, re-entrance at state 1
[ ] (s_rib=0) -> FAST*FAST : (s_rib'=1) ;
// rejection, re-entrance at state 1
[ ] (s_rib=5) -> FAST*FAST : (s_rib'=1) ;
// elongation
[ ] (s_rib=8) -> FAST*FAST : (s_rib'=8) ;

endmodule

```