Stochastic Modeling of Codon Bias with PRISM

T.E. Pronk[a,b], E.P. de Vink[1,c], D. Bošnački[d] & T.M. Breit[a]

[a] *Integrative Bioinformatics Unit, Universiteit van Amsterdam*

[b] *Computer Systems Architecture Group, Universiteit van Amsterdam*

[c] *Department of Mathematics and Computer Science, TU/e, Eindhoven*

[d] *Department of Biomedical Engineering, TU/e, Eindhoven*

Abstract We present a biological case study of codon bias with the probabilistic model checker PRISM with which we perform a quantitative analysis of expression speeds of genes. The variability in this setting concerns the matching of codons and anticodons. We distinguish between iso-acceptance (one codon matches exactly one anticodon) and wobble matching (multiple codons are matched by one anticodon). Our modeling confirms a recent result by Lavner and Kotlar, stating that at low and high expression speeds high codon bias prevails, whereas at moderate expression speeds low codon bias is more advantageous. Although the approach needs further validation, our preliminary investigation shows that probabilistic model checking with PRISM is promising in obtaining further insight in codon bias.

Key words: Systems biology, codon bias, probabilistic model checking, PRISM

1 Introduction

Differential equations play a predominant role in the numerical study of biological systems. For stochastic systems, a substantial amount of research is based on simulations using the well-known Gillespie algorithm. Alternative approaches have emerged in recent years that exploit the similarity of systems in biology and computer science. The analogy between the concept of a process in these fields has been exploited in applications of formal methods in a biological context. For instance, model checking as an automated formal technique, originally developed for the debugging and verification of software and hardware. Here we use probabilistic model checking, where besides qualitative properties also quantitative aspects of a system are considered, like performance and probability.

---

[1] Corresponding author: `evink@win.tue.nl`

In this paper, we tackle the problem of codon bias using the probabilistic model checker PRISM. Codons encode amino acids, which are the building blocks of proteins. Proteins are involved in most of the reactions in the cell. During the translation of a sequence of codons into a protein, the cellular machinery has to supply the appropriate amino acids that the protein is made of. This process is inherently stochastic. Interestingly, between organisms and even between proteins, there can be a large difference in the choice of particular codons to encode an amino acid. A specific preference for the use of particular codons is generally known as the codon bias. The underlying mechanism is not fully understood yet and is the topic of on-going debate.

A number of factors play a key role in the translation of the codon sequence. Therefore, a model should be in place that is appropriate to test assumptions on the relevant variables. Surprisingly, little work on executable models exists. Instead, biologists have relied on statistical methods. The model proposed here concerns the variability in the matching of codons and so-called anticodons. Here, we distinguish between iso-acceptance, when one codon matches exactly one anticodon, and wobble, when multiple codons are matched by one anticodon. Our modeling, although rather simplistic, confirms a recent insight by Lavner and Kotlar [13,10], stating that at low and high translation speeds high codon biases prevail, whereas at moderate translation speeds low codon biases are more advantageous. Moreover, although the differences were small, iso-acceptance fitted with high codon biases, while low codon biases matched with wobbles. The model that we build with PRISM has the same modular structure for all experiments we conduct: a module for the codons, that drives the computation and various reactive amino acid modules.

To summarize, our contributions are a validation of the use of quantitative model checking for biological systems. In particular, we provide a formal model of the codon bias problem. The experiments conducted with the model confirm biological results obtained with correlation analysis. Initial findings suggest that the distinction between iso and wobble acceptance is an important factor when comparing codon bias and expression speed. As further confirmation is needed, our hypothesis may serve as a starting point for biological experiments. In general, our case-study adds to the claim that methods and tools from formal methods such as the PRISM model checker are suitable for addressing biological problems.

Related work Formal methods and model checking have been applied earlier in the biological setting. For instance, the stochastic π-calculus has been proven successful as a modeling language for biological systems and their simulation [14,15,11]. Qualitative approaches in this context exploit e.g. CTL with NuSMV [4,5] and Rewrite Logic with Maude [7,18,17]. The Pathway Logic Assistant, built on top of the latter system, provides a visualization of pathways and an interface that is intuitive to the working biologist. Wolf pro-

2

poses a modular approach based on stochastic automata networks to model biochemical reactions as sparse diagonally organized matrices representing directly continuous-time Markov chains [19]. Closer to our approach is the work of Calder et al. In [3], a quantitative case-study on a signaling network of Raf1 is conducted using PRISM. Discrete concentration levels are used to abstract away from individual molecules. Heath et al. report in [9] on a model with PRISM of the fibroblast growth factor focusing on the timing of signaling. The strength of the approach is high-lighted by successive elimination of pathway elements and analysis of the subsequent effect. This is, in general, the main advantage of modeling of biological systems in silico, the flexibility to add and remove components in a clean way and measure the behavioral and numerical changes.

2 Codon bias

The translation of genes to proteins is one of the core processes in cellular life. Proteins are the elements that actually perform the majority of activities in cellular processes. Proteins are produced from genes in two stages: From one of the strands of a double stranded stretch of DNA nucleotide sequence, i.e. a gene, single stranded mRNA copies are made with help of a RNA polymerase in a process called transcription. Subsequently, from each mRNA copy, proteins can be produced with help of ribosomes in a process called translation. In the present case study, we will consider only translation.

Proteins are effectively long, typically folded, strains of amino acids. There are only twenty standard different amino acids present in living material. An mRNA is a sequence of nucleotides, which contains four types of base A, C, G and U, short for adenine, cytosine, guanine and uracil. Each triplet of nucleotides forms a codon. A codon codes for an amino acid in a unique way. This is called the genetic code (see Table 1). So, an mRNA, as sequence of codons, codes for a sequence of amino acids, i.e. a protein. Four nucleotides organized in triplets results in 64 codons. Thus, multiple codons code for an amino acid. The extent to which cells use the variety of codon types is called the codon bias. With high codon bias, a particular codon occurs more frequently than others that code for the same amino acid. With low codon bias, codon usage is more equally spread over the various codon types. Translation of a codon in an mRNA into an amino acid in a protein is established via an intermediate transfer RNA (tRNA). tRNA are non-coding RNAs with a strict tertiary structure. Each tRNA is present in a genome with a range from 0 (only possible in theory) to over 30 copies, which will have a major impact on its cellular concentration level [13]. Within the tRNA sequence an anticodon that is complementary to a given codon is present and this gives the tRNA it specificity. At the same time, each specific tRNA carries a specific amino

3

acid and may result in translation of the codon via the corresponding anti-codon into the amino acid. So, while the ribosome reads the mRNA codon by codon, tRNAs bring and couple successively the appropriate amino acids for the protein coded by the mRNA.

Ala	GCU, GCC, GCA, GCG	Leu	UUA, UUG, CUU, CUC, CUA,CUG
Arg	CGU, CGC, CGA, CGG, AGA, AGG	Lys	AAA, AAG
Asn	AAU, AAC	Met	AUG
Asp	GAU, GAC	Phe	UUU, UUC
Cys	UGU, UGC	Pro	CCU, CCC, CCA, CCG
Gln	CAA, CAG	Ser	UCU, UCC, UCA, UCG, AGU,AGC
Glu	GAA, GAG	Thr	ACU, ACC, ACA, ACG
Gly	GGU, GGC, GGA, GGG	Trp	UGG
His	CAU, CAC	Tyr	UAU, UAC
Ile	AUU, AUC, AUA	Val	GUU, GUC, GUA, GUG

Table 1: The genetic code

The codon-anticodon matching is largely governed by base pairing of the first two nucleotides of the codon. Complementarity of the third nucleotide is less significant. The base pairing is sequence specific in that A–U base pairs have two hydrogen bonds and G–C base pairs three. Although we know that other factors influences base pairing strength, for this study we defined 4 levels of binding strength, as there are may be 6 up to 9 hydrogen bonds between the codon and anticodon. Dependent on the codons in the mRNA sequence, there can be two situations. 1) An anticodon only matches one codon. In this case the anticodon is referred to as iso-accepting. 2) An anticodon matches more than one codon. Then the anticodon is said to wobble. The same terminology is used for tRNA with respect to the anticodon that is carried.

In the experiments with PRISM discussed in the sequel, we will distinguish between types of codon/anticodon acceptance, deal with codon bias at the mRNA and handle different concentration levels for tRNAs.

3   Probabilistic Model Checking and the PRISM tool

Probabilistic model checking is a formal technique that is used for the analysis of systems that exhibit stochastic behavior. In probabilistic model checking, we check a probabilistic model of the system under consideration against a formal specification of a required or undesired property. The model is usually described in a formal language, quite similar to a programming language. The property is typically specified as a formula in a temporal logic extended with special operators that capture probabilistic and timing aspects of the system [8,2]. The analysis is done automatically by a tool that explores the state space of the model. Often, the model checking tool yields true or false as output, expressing whether the specified property holds on the model. Al-

ternatively, the output of the tool can be a numerical value.

PRISM [12] is one of the most popular probabilistic model checkers. It has been used in various applications, including biological systems [3,9]. PRISM supports three kinds of probabilistic models: discrete-time and continuous-time Markov chains, as well as Markov decision processes. In this paper we use continuous-time Markov chains (CTMCs) only.

PRISM provides a state-based modeling language which is a probabilistic version of Reactive Modules [1]. A PRISM model description is compiled into a Markov chain that can be analyzed by the tool. Each description consists of a collection of modules. A description contains declarations of constants and variables, and a set of transitions specified in a guarded commands style, i.e., a condition followed by an action. The modules can communicate with one another via shared memory (global variables) or by synchronizing on their labels. An example of a PRISM model is given next.

```
stochastic

const int N = 10;
const double r1 = 1.0;
const double r2 = 0.1;
const double r3 = 0.6;

module A
  x: [0..N];
  [a] x < N  ->  r1 : x'=x+1;
  [b] x > 0  ->  r1 : x'=x-1;
  [ ] x = 0  ->  r3 : x'=x+1;
endmodule

module B
  y : [0..5];
  [a] y > 0  ->  r2 : y'=y-1;
  [ ] y < 5  ->  r3 : y'=y+1;
endmodule
```

The transition pattern in the model description is `[label] condition -> rate: action`. The interpretation, of such a command on its own, reads: if the `condition` holds, the `action` is executed after some delay. The delay is modelled as an exponentially distributed arrival process of `rate` arrivals on the average per unit of time. In PRISM, it is characteristic for the CTMC model that the joint rate of each synchronous transition is a product of the rates of the component transitions. For instance, when the transitions labeled with `a` in modules `A` and `B` synchronize, the synchronized transition has rate `r1*r2` which is the product of the two component rates `r1` and `r2`.[1] The PRISM

---

[1] Such a concept of product of rates is often disputed by the probabilistic model checking community. However, in our setting it does not make sense to consider the component transitions as stand-alone actions with separate rates. They are just components of a

description also can have transitions that are not synchronized, such as the transitions with rate `r3` in the example above. They do not have labels and they can be considered as independent, spontaneous actions of the module.

PRISM can be used for the verification of different kinds of qualitative and quantitative properties. The properties are specified in the logics PCTL [8] – for models that are discrete time Markov chains and Markov decision processes– and in CSL [2] –for continuous time Markov chains. The following informally specified examples illustrate the kind of properties that can be analyzed by PRISM:

- "The program successfully terminates with probability 1."
  ($\mathcal{P}_{\geq 1}[\,\texttt{true}\;\mathcal{U}\;\texttt{terminated}\,]$);

- "In the long-run (steady state) the probability that the temperature drops to 0 is less than 0.3." ($\mathcal{S}_{<0.3}[\,\texttt{true}\;\mathcal{U}\;\texttt{temperature = 0}\,]$);

- "What is the probability that if the number of molecules of type A is greater than 10000, then between 23 and 25 hours the number of molecules of type B will become greater than 3000?"
  ($\mathcal{P}_{=?}[\,(\texttt{nrA > 10000}) \Rightarrow (\texttt{true}\;\mathcal{U}_{[23,25]}\;(\texttt{nrB > 3000}))]$).

Besides verification, PRISM can also perform simulation of the model. The tool can be used from the command line or via a graphical user interface. More details about PRISM are available form the web page of the tool [16].

4 Modeling

The PRISM model for the codon bias problem consists of modules that correspond to the mRNA and the amino acids. The other elements of the translational machinery are implicit. In this section we discuss the general form of the PRISM modules that are use in the experiments.

The mRNA module The mRNA module, interpreted as sequence of codons, is driving the computation. Control steps along the codons one-by-one. Each codon of the mRNA requires synchronization with the corresponding amino acid module. In general, there are several codons for the amino acid. By the handshake of the mRNA module on the one hand, and, the amino acid module on the other hand, one of the codons for the amino acid is selected. Because of the linear form of automaton underlying the mRNA module, the selection of amino acid modules is deterministic.

Next, the mRNA module waits to synchronize with the particular amino acid module again. This handshake expresses that the amino acid was successfully added to the protein in nascent. These two steps comprise the translation

---

joint transition and the distribution of the synchronized rate over its components is only a convenience in the model description. Therefore, the product rate feature of PRISM to model chemical reactions comes quite natural. See, e.g., the 'molecular reactions' example at [16].

of a single codon of the mRNA; the computation proceeds as long as there are codons to process. A possible repetitive structure present in the experiment mRNA, is captured by counter controlled iteration.

```
module mRNA
  s : [0..6] init 1;
  cnt : [0..N] init 0;
  ready : bool init false;

  // Nx 3 codon mRNA CGA-GGG-AAG for protein Arg-Gly-Lys
  [cga] s=1  ->  ONE : s'=2; // CGA codon for Arg
  [arg] s=2  ->  ONE : s'=3; // Arg added to AA-chain
  [ggg] s=3  ->  ONE : s'=4; // GGG codon for Gly
  [gly] s=4  ->  ONE : s'=5; // Gly added to AA-chain
  [aag] s=5  ->  ONE : s'=6; // AAG codon for Lys
  [lys] s=6  ->  ONE : s'=0; // Lys added to AA-chain
   [ ]   s=0 & cnt<N  ->  FAST : s'=1 & cnt'=cnt+1;  // one more
   [ ]   s=0 & cnt=N  ->  FAST : ready'=true;        // stop
endmodule
```

The mRNA represented by the module above, consists of a repetition of three codons, viz. CGA, GGG and AAG, coding for the protein Arg-Gly-Lys composed, in that order, from the amino acids Arginine, Glycine and Lysine, respectively. The state transitions, starting from state 1, describe codon selections and protein elongation, alternatingly, requiring synchronization on the labels cga, arg, and so on, with the corresponding amino acid modules. After the counter cnt has been incremented sufficiently, the computation stops in state 0 with the boolean ready set to true. There is no synchronization demand for the latter two commands. Due to their rate FAST, their time is negligible.

Iso-acceptance The simplest scheme of codon-anticodon matching is iso-acceptance: there is exactly one type of anticodon –in the tRNA pool– that fits with the codon, and there is exactly one type of codon at the mRNA that fits with the anticodon. Below an example for the amino acid Arginine: the two codons CGC and CGG correspond to, in our terminology, the two iso-accepting anti-codons GCG and GCU, respectively. (Note that anticodons are denoted in 3' to 5' orientation.)

```
// amino acid Arginine iso-acceptance: CGC-GCG, CGG-GCU

[cgc]  s_arg=0  ->  FAST  : s_arg'=1;
 [ ]   s_arg=1  ->  r_gcg : s_arg'=2;
 [ ]   s_arg=2  ->  STR4*FAST : (s_arg'=3) +
               (1-STR4)*FAST : (s_arg'=4);
[arg]  s_arg=3  ->  SUCC  : s_arg'=0;
 [ ]   s_arg=4  ->  FAIL  : s_arg'=1;

[cgg]  s_arg=0  ->  FAST  : s_arg'=5;
```

7

```
[ ]    s_arg=5  ->  r_gcu : s_arg'=6;
[ ]    s_arg=6  ->  STR2*FAST : (s_arg'=7) +
                    (1-STR2)*FAST : (s_arg'=8);
[arg]  s_arg=7  ->  SUCC  : s_arg'=0;
[ ]    s_arg=8  ->  FAIL  : s_arg'=5;
```

Starting from the state 0 a transition is made while synchronizing on the label cgc (or cgg) shared with the mRNA module. Note, according to the parallel construct of PRISM, the rate of the synchronized commands is the product of the rates of the individual commands. Hence, the handshake occurs at speed ONE*FAST and the time spent can be ignored. The non-deterministic choice for leaving state 0 is resolved by the mRNA module, as the transition to state 1 and 5 are labeled differently. From state 1, reached after a cgc-transition from state 0, the waiting time for a tRNA with anticodon GCG is captured by the rate r_gcg. In the next state 2, a probabilistic choice is made. There is a probability of STR4 that the tRNA with anticodon attaches successfully, and a probability of 1-STR4 that it will not. Here, STR4 represents strength level 4. As we consider the probabilistic choice not to take time, a multiplication of the probabilities STR4 and 1-STR4 with the large constant FAST is incorporated, rendering the expected sojourn negligible. In case of success, probability STR4, control proceeds to state 3, and apparently after some processing at the ribosome, continues to the initial state 0, ready to respond to the next codon for Arginine. The successful elongation of the protein with the Arginine amino acid is signaled to the mRNA module by synchronization on the arg label. In case of failure, probability 1-STR4, control returns to state 1 after residing in state 4, to take the time taken by the failing interaction into account. In state 1 the computation will wait again for a tRNA with anticodon GCG to arrive. The PRISM commands for the codon CGG are similar and disjoint from those for the codon CGC.

Wobble acceptance For a wobble anticodon for an amino acid in the tRNA pool there are multiple codons at the mRNA that match. However, as the mRNA module is driving the computation, the amino acid module is not essentially different from that of an iso-accepting anticodon. Below PRISM code that implements this for the amino acid Glycine where both codons GGC and GGU match the wobble CCG.

```
// amino acid Glycine wobble-acceptance: GGC-CCG and GGU-CCG

[ggc]  s_gly=0  ->  FAST  : s_gly'=1;
[ ]    s_gly=1  ->  r_ccg : s_gly'=2;
[ ]    s_gly=2  ->  STR4*FAST : (s_gly'=3) +
                    (1-STR4)*FAST : (s_gly'=4);
[gly]  s_gly=3  ->  SUCC  : s_gly'=0;
[ ]    s_gly=4  ->  FAIL  : s_gly'=1;

[ggu]  s_gly=0  ->  FAST  : s_gly'=5;
```

```
[ ]    s_gly=5  ->  r_ccg : s_gly'=6;
[ ]    s_gly=6  ->  STR2*FAST : (s_gly'=7) +
                    (1-STR2)*FAST : (s_gly'=8);
[gly]  s_gly=7  ->  SUCC  : s_gly'=0;
[ ]    s_gly=8  ->  FAIL  : s_gly'=5;
```

Note that, the probabilities for success in the state 2 and 5 are different as this depends on the binding strength of the codon and anticodon involved. Apparently, the binding strength is strength 4 for GGC and CCG and strength 2 for GGU and CCG, cf. Table 2.

Mixed acceptance Above, in the tRNA pool a single type of anticodon was available to accommodate two different codons. The complementary situation is theoretically possible as well. Below an adaptation of the Glycine module is given for the case where the codon GGG associates in the tRNA pool with the anticodon CCC, as above, but also with the anticodon CCU.

```
// Glycine mixed acceptance: GGG-CCC and GGG-CCU

[ggg]  s_gly=0  ->  FAST : s_gly'=1;

[ ]    s_gly=1  ->  r_ccc  : s_gly'=2;
[ ]    s_gly=2  ->  STR4*FAST : (s_gly'=3) +
                    (1-STR4)*FAST : (s_gly'=4);
[gly]  s_gly=3  ->  SUCC : s_gly'=0;
[ ]    s_gly=4  ->  FAIL : s_gly'=1;

[ ]    s_gly=1  ->  r_ccu  : s_gly'=5;
[ ]    s_gly=5  ->  STR2*FAST : (s_gly'=6) +
                    (1-STR2)*FAST : (s_gly'=7);
[gly]  s_gly=6  ->  SUCC : s_gly'=0;
[ ]    s_gly=7  ->  FAIL : s_gly'=1;
```

Main difference, compared to the previous PRISM snippets, is that state 1 is shared for continuation after synchronization on ggg. Indeed, as both anticodon CCC and anticodon CCU fit with the codon GGG, the selection of the anticodon is decided on the race-condition for the arrival of the anticodons. In the situation for iso-acceptance for Arginine discussed above, the choice for the anticodon was, in fact, implied by the codon selection. Success and failure for attachment have different probabilities, viz. of strength 4 and 2. Also note that the concentrations of the two types of tRNA, represented by the rates r_ccc and r_ccu, may differ.

In the experiments presented in the next section we have used modules of the above type in different combinations. Both the mRNAs and the tRNA pool –not modeled explictly but represented by the collective of the amino acid modules together– are varied. Casted in the high-level language of PRISM with its parallel construct of synchronization on labels and product of rates,

	codon	anticodon	binding		codon	anticodon	binding
Arginine	CGC	GCG	4	Glycine	GGC	CCG	4
	CGU	GCG	2		GGU	CCG	2
	CGG	GCC	4		GGG	CCC	4
	CGG	GCU	2		GGG	CCU	2

Table 2: Example codon-anticodon pairings with associated binding strength

this can be achieved rather conveniently.

5   Experiments

As a first set-up illustrating the approach, we analyze the efficiency of different types of codon/anticodon combinations for the translation of a protein. We want to produce proteins each consisting of the amino acids Arginine and Glycine alternatingly, in total 40 amino acids long. We distinguish three types of codon/anticodon correspondence: iso-acceptance, wobble acceptance as well as a mixed variant. With each experiment, we change the codons encoding the amino acids on the mRNA and adapt the tRNA pool regarding the precense of anticodons that recognize the codons. The availability of tRNAs is held equal among all experiments, i.e., the total amount of tRNAs available for the translation of the mRNA is fixed. For the moment, we do not take into account the difference in energy for building an Arginine or Glycine into the protein. In our modeling with PRISM this implies that the rates for the arrival process of an anticodon to the ribosome remains unchanged during the experiment. We focus at the production times for varying tRNA pools of iso-accepting, mixed or wobble anticodons. We choose the codons and anticodons in such a way that the total binding strengths are the same across experiments.

Table 2 displays for the two amino acids Arginine and Glycine each, two codon-anticodon pairs of strength 4 and two pairs of strength 2 that we consider. By selecting different codons at the mRNA and varying the anticodons available in the tRNA pool, different types of matching can be emulated. For the iso-acceptance experiment we chose the translation of an mRNA of 10 blocks of `CGC-GGC-CGG-GGG` with codon-anticodon pairs `CGC-GCG` and `CGG-GCU` for Arginine, and, `GGC-CCG` and `GGG-CCU` for Glycine. So, for each type of codon there is exactly one type of anticodon that matches. In the mixed situation we have for the translation of a 20 block `CGG-GGG` with the pairs `CGG-GCC` and `CGG-GCU` for Arginine, and, `GGG-CCC` and `GGG-CCU` for Glycine. Finally, in the wobble experiment, there are `CGC-GCG` and `CGU-GCG` for Arginine, and, `GGC-CCG` and `GGU-CCG` for Glycine available for the translation of an mRNA of 10 blocks `CGC-GGC-CGU-GGU`. In all cases, for both amino acids there is a pair of strength 4 and a pair of strength 2.

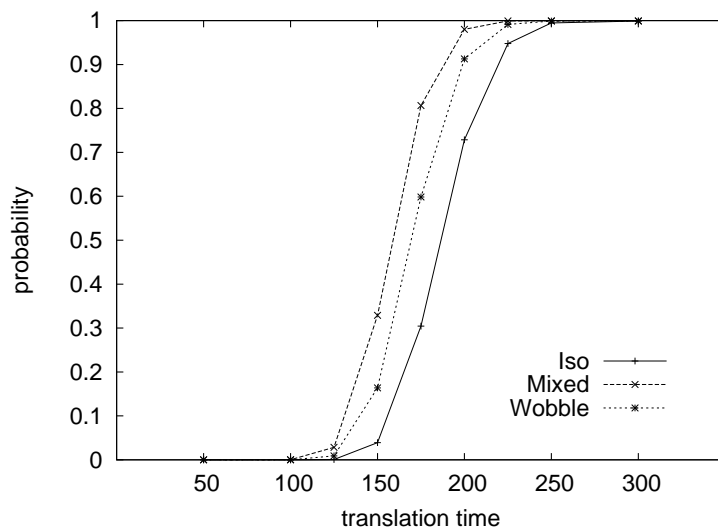The PRISM model consists, for each of the three cases, of a module for

Figure 3: Translation time for various mRNA and tRNA combinations

the mRNA, two modules for the amino acids Arginine and Glycine, the form being dependent on the type of acceptance as sketched in Section 4. Expression speeds are obtained by model checking the induced continuous-time Markov chain against the CSL formulae $\mathcal{P}_{\texttt{=?}}\left[\,\texttt{true}\;\mathcal{U}_{\leq\texttt{n}}\;\texttt{ready}\,\right]$ where $\texttt{n}$ ranges from 50 to 300. For fixed $\texttt{n}$, the formula expresses the total probability of reaching from the initial state of the underlying Markov chain, within $\texttt{n}$ time units, a state satisfying $\texttt{ready}$. For example, for the case of the $\texttt{CGC-GGC-CGU-GGU}$ mRNA with wobble tRNA pool with $\texttt{GGG}$ and $\texttt{CCG}$ anticodons for Arginine and Glycine, respectively, we have that

$$\mathcal{P}_{\texttt{=?}}\left[\,\texttt{true}\;\mathcal{U}_{\leq 175}\;\texttt{ready}\,\right] = 0.598.$$

The built-in numerics algorithms of PRISM do the calculations in seconds.

Figure 3 summarizes the results. For the particular experiments, iso-acceptance tends to be slower than wobble acceptance, which in turn is slower than mixed anticodons in the tRNA pool. The case for iso-acceptance is clear: each of the codons per amino acid has to be bound by a tRNA of a particular type. Because the total abundance of tRNA is kept constant, the concentration per particular tRNA type is less. This prolongs the time for connecting (captured by the rates $\texttt{r\_gcg}$, $\texttt{r\_gcu}$, $\texttt{r\_ccgt}$, and $\texttt{r\_ccu}$) and slows the translation down. For the codons in the wobble and mixed case there is more flexibility, which –in a sense– doubles the chance of a quick encounter.

In the experiment for the wobble, there is only one type of tRNA and all individual tRNA's of that type can bind the codons of one amino acid. In the mixed experiment both types of tRNA match the codon, which in both cases effectively doubles the chance of a quick encounter between tRNA and codon. The recorded difference between wobble and mixed experiment

11

is more subtle and depends on the values of successful binding of tRNA and ribosome (captured by STR2 and STR4). Namely, if a weak binding fails in the wobble case, the binding is renewed with another weak binding because the binding strength is determined by the codon. In the mixed case, the strength is determined by the tRNA anticodon and after failure the tRNA with a strong binding may attach to the ribosome. This will speed up the translation. To summarize, with equal binding strengths (STR2 and STR4) between experiments, we find that a protein can be translated most effectively by either a codon that is recognized by multiple anticodons, or codons that are recognized by a single wobble anticodon.

Typically, however, wobble tRNAs have on average binding strengths that are less than for iso-accepting tRNAs. Therefore, the problem remains to find when it is best to use wobble and when iso-accepting tRNAs when different strengths are taken into account. We will tackle this question in the following set-up.

high-cost amino acids		low-cost amino acids	
Threonine	21.6	Glycine	1.0
Proline	31.8	Valine	12.3
Arginine	56.4	Leucine	16.0

Table 4: Duflon complexity of some amino acids

In our second round of experiments, we produce proteins consisting of three amino acids. Amino acids differ in the cost to incorporate them in the protein, as is reflected by the Dufton complexity score [6]. To keep the energy investment for codon translation equal over experiments, we assume the abundance and hence the arrival rates of tRNAs reciprocal to the Dufton complexity of amino acids involved. See Table 4. As high-costs amino acids we select Threonine, Proline and Arginine, as low-costs amino acids Glycine, Valine and Leucine are taken in the experiments. We also take into account the variety of codons used to code for a particular amino acid. This is referred to as codon bias. If the codon bias is high, there is a high preference for one codon over other codons that are available for an amino acid. If the codon bias is low, multiple codons are available in comparable amounts to code for an amino acid.

For the production of a high-cost protein we distinguish between an mRNA of 20 ACG-CCG-CGG groups vs. an mRNA of 10 ACG-CCU-CGG-ACU-CCG-CGU groups. Both mRNAs code for a protein of 20 repetitions of a Thr-Pro-Arg string of amino acids. In case of the low-cost protein, we have a GGG-GUG-CUU mRNA with a high codon bias on the one hand, and a GGG-GUG-CUG-GGU-GUU-CUU mRNA of a low codon bias on the other hand. Again, both mRNAs are 60 codons long and code for a Gly-Val-Leu protein. Thus, for the production of our proteins, we vary the amino acids according to cost as well as according to codon bias. Moreover, for each of the four experiments, we also calculate whether translation is more efficient using either
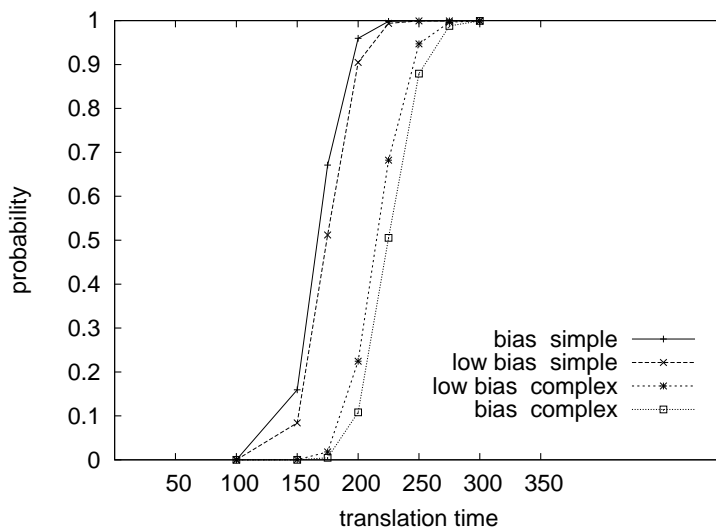
Figure 5: High codon bias (`bias`) vs. low codon bias (`low bias`), high-cost amino acid (`complex`) vs. low-cost amino acid (`simple`)

wobble or iso-accepting tRNA. As before, we provide varying tRNA pools for this. However, the results show that the differences for iso-accepting and wobble tRNAs are minute, in particular for the cases of low codon bias (not shown).

Figure 5 shows the efficiency for the outcomes of the various mRNAs. The highest translation speeds are obtained in case of low-cost proteins, where high codon bias is slightly quicker than low codon bias. The proteins consisting of expensive amino acids are translated consistently slower than the amino acids consisting of cheap amino acids because for the latter case less tRNA is available for the translation [6]. This is the direct consequence of the cost/tRNA level trade-off.

Our experiment is in line with Lavner and Kotlar [13]. Exploiting correlations and statistics on data of tRNA gene copy numbers (representing tRNA abundance), expression levels, codon bias and amino acid complexity, they found that slow and rapidly expressed genes are both coded with a high codon bias, whereas genes expressed at medium speed are coded with a low codon bias. In our executable model the same variables are taken into account but simplified by correlating tRNA abundance with amino acid complexity score, replacing expression level with translation time per protein, and regarding only two levels of codon bias and amino acid complexity.

Although in accordance with biological data, the empirical significance of our findings is limited. In addition, the results in the model are dependent on fitted parameters of binding strength and tRNA concentration. It is clear, that further investigations are needed. However, the main point here is that modeling with PRISM is feasible and follow-up research makes sense.

6   Concluding remarks

We presented a novel formal model in PRISM of the biological problem of codon bias. Using the quantitative features of probabilistic model checking, we obtained results that are in agreement with biological literature. Moreover we were able to raise new hypotheses (e.g., iso and wobble acceptance are an important factor when comparing codon bias and expression speed) that would be interesting to validate via biological experiments.

Compared to the existing approaches to tackle the codon bias problem, which are based on qualitative models and statistics, the main advantage of our model is that it is executable. As such the model lends itself naturally to simulation and model checking. This can further be exploited for efficient modifications of the model. Also, in our opinion, such modular models with a programming appeal, are much more intuitive than others in different formalisms, like, for instance, (stochastic) differential equations.

In general, our model supports the conclusion from several previous experiences of using PRISM in a life science context, in that the tool is suitable for the analysis of biological systems. Such an approach adds to the current collection of biological modeling ways by the principle of stochasticity and formality and high level.

In the future, we plan to extend the experiments, being realistic in size already regarding the number codons, to mRNAs of a richer variation in codons that encode for real proteins. Being optimistic regarding the scalability of the approach for the case of codon bias, it would be interesting to refine the model by including new parameters and try to understand their balance.

References

[1] R. Alur and T. Henzinger. Reactive modules. *Formal Methods in Systems Design*, 15:7–48, 1999.

[2] C. Baier, J.-P. Katoen, and H. Hermanns. Approximate symbolic model checking of continuous-time Markov chains. In J.C.M. Baeten and S. Mauw, editors, *Proc. Concur'99*, pages 146–161. LNCS 1664, 1999.

[3] M. Calder, V. Vyshemirsky, D. Gilbert, and R. Orton. Analysis of signalling pathways using continuous time Markov chains. In C. Priami and G. Plotkin, editors, *Transactions on Computational Systems Biology VI*, pages 44–67. LNBI 4220, 2006.

[4] N. Chabrier and F. Fages. Symbolic model checking of biochemical networks. In C. Priami, editor, *Proc. CMSB 2003*, pages 149–162. LNCS 2602, 2003.

[5] N. Charbrier-Rivier, M. Chiaverini, V. Danos, F. Fages, and V. Schächter. Modeling and querying biomolecular interaction networks. *Theoretical Computer Science*, 325:25–44, 2004.

[6] M.J. Dufton. Genetic code synonym quotas and amino acid complexity: cutting the cost of proteins? *Journal of Theoretical Biology*, 187:165–173, 1997.

[7] S. Eker, M. Knapp, K. Laderoute, P. Lincoln, J. Meseguer, and M.K. Sönmez. Pathway Logic: symbolic analysis of biological signaling. In R.B. Altman, A.K. Dunker, L. Hunter, and T.E. Klein, editors, *Proc. Biocomputing 2002*, pages 400–412, Lihue, 2002.

[8] H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6:512–535, 1994.

[9] J. Heath, M. Kwiatkowska, G. Norman, D. Parker, and O. Tymchyshyn. Probabilistic model checking of complex biological pathways. In C. Priami, editor, *Proc. CMSB 2006*, pages 32–47. LNBI 4210, 2006.

[10] D. Kotlar and Y. Lavner. The action of selection on codon bias in the human genome is related to frequency, complexity and chronology of amino acids. *BMC Genomics*, 7:67–77, 2006.

[11] C. Kuttler. Simulating bacterial transcription and translation in a stochastic π-calculus. In C. Priami and G. Plotkin, editors, *Transactions on Computational Systems Biology VI*, pages 113–149. LNBI 4220, 2006.

[12] M. Kwiatkowska, G. Norman, and D. Parker. Probabilistic symbolic model checking with PRISM: a hybrid approach. *Journal of Software Tools for Technology Transfer*, 6:128–142, 2004.

[13] Y. Lavner and D. Kotlar. Codon bias as a factor in regulating expression via translation rate in the human genome. *Gene*, 345:127–138, 2005.

[14] C. Priami, A. Regev, E. Shapiro, and W. Silverman. Application of a stochastic name-passing calculus to representation and simulation of molecular processes. *Information Processing Letters*, 80:25–31, 2001.

[15] A. Regev, E.M. Panina, W. Silverman, L. Cardelli, and E.Y. Shapiro. Bioambients: an abstraction for biological compartments. *Theoretical Computer Science*, 325:141–167, 2004.

[16] School of Computer Science, University of Birmingham. *PRISM Manual, Version 3.1*, 2006. http://www.cs.bham.ac.uk/~dxp/prism/.

[17] C.L. Talcott. Symbolic modeling of signal transduction in Pathway Logic. In L.F. Perrone, B. Lawson, J. Liu, and F.P. Wieland, editors, *Proc. WSC 2006*, pages 1656–1665, Monterey, 2006.

[18] C.L. Talcott, S. Eker, M. Knapp, P. Lincoln, and K. Laderoute. Pathway Logic modeling of protein functional domains in signal transduction. In R.B. Altman, A.K. Dunker, L. Hunter, T.A. Jung, and T.E. Klein, editors, *Proc. Biocomputing 2004*, pages 568–580, Hawaii, 2004.

[19] V. Wolf. Modelling of biochemical reactions by stochastic automata networks. In N. Busi and C. Zandron, editors, *Proc. MeCBIC 2006, Venice*, to appear in the ENTCS.