BMC Bioinformatics

Software

Sequence similarity is more relevant than species specificity in probabilistic backtranslation

Alfredo Ferro*[1,2], Rosalba Giugno[1], Giuseppe Pigola[1], Alfredo Pulvirenti[1], Cinzia Di Pietro[2], Michele Purrello[2] and Marco Ragusa[2]

Address: [1]Dipartimento di Matematica e Informatica, Università di Catania, Viale A. Doria 6, I-95125 Catania, Italy and [2]Dipartimento di Scienze Biomediche, Università di Catania, Via S. Sofia 87, I-95125 Catania, Italy

Email: Alfredo Ferro* - ferro@dmi.unict.it; Rosalba Giugno - giugno@dmi.unict.it; Giuseppe Pigola - pigola@dmi.unict.it; Alfredo Pulvirenti - apulvirenti@dmi.unict.it; Cinzia Di Pietro - dipietro@unict.it; Michele Purrello - purrello@unict.it; Marco Ragusa - mragusa@unict.it

* Corresponding author

Published: 21 February 2007

BMC Bioinformatics 2007, **8**:58 doi:10.1186/1471-2105-8-58

Received: 28 July 2006
Accepted: 21 February 2007

This article is available from: http://www.biomedcentral.com/1471-2105/8/58

Abstract

Background: Backtranslation is the process of decoding a sequence of amino acids into the corresponding codons. All synthetic gene design systems include a backtranslation module. The degeneracy of the genetic code makes backtranslation potentially ambiguous since most amino acids are encoded by multiple codons. The common approach to overcome this difficulty is based on imitation of codon usage within the target species.

Results: This paper describes EasyBack, a new parameter-free, fully-automated software for backtranslation using Hidden Markov Models. EasyBack is not based on imitation of codon usage within the target species, but instead uses a sequence-similarity criterion. The model is trained with a set of proteins with known cDNA coding sequences, constructed from the input protein by querying the NCBI databases with BLAST. Unlike existing software, the proposed method allows the quality of prediction to be estimated. When tested on a group of proteins that show different degrees of sequence conservation, EasyBack outperforms other published methods in terms of precision.

Conclusion: The prediction quality of a protein backtranslation methis markedly increased by replacing the criterion of most used codon in the same species with a Hidden Markov Model trained with a set of most similar sequences from all species. Moreover, the proposed method allows the quality of prediction to be estimated probabilistically.

Background

In natural systems, proteins are synthesized using template mRNA derived from molecules transcribed from the encoding genes. Backtranslation (reverse translation) reverses the normal flow of information, exploiting the primary structure of a protein to deduce the nucleotide sequence of the encoding mRNA. Backtranslation tools are basic to the construction of synthetic DNA segments (gene design systems) [1]. Such systems use suitable modules to optimize backtranslated segments to be used for expression by a host organism, or to be changed completely to accommodate various constraints [2-4].

The degeneracy of the genetic code makes backtranslation potentially ambiguous since most amino acids are encoded by multiple codons. Extensive studies have been conducted on synonymous codon usage in different species and its influence in biological processes such as structure prediction [5-9].

The approach to backtranslation common to all commercial and non-commercial software (BBOCUS [10], BACKTRANSEQ of the EMBOSS software suite [11]) is based on imitation of codon usage within the target species. For some of these methods, expert supervision is required to construct the codon usage tables. Several methods are based on the hypothesis that specific genomic contexts may influence codon usage (TIP [12,13], LBT [14]). The genetic algorithm TIP uses a set of "coding statistics", whereas LBT exploits Multiple Sequence Alignment (MSA) of the class of proteins under analysis. Both software packages give high-precision results. However, their users must set a number of parameters if the results are to be reliable.

In this paper, a parameter-free and fully-automated software called EasyBack is proposed. Given an amino acid sequence as input, EasyBack tries to reconstruct the codon usage of the gene under analysis using a Hidden Markov Model (HMM) [15]. The model is trained with an "input-driven" training set. This set of proteins is constructed from the input protein by querying the NCBI [16] databases with BLAST. The training set will be the "smallest" subset of the query output needed for HMM to make a prediction. The prediction is made by classical Viterbi or posterior decoding algorithms [15]. Prediction quality can be estimated by analyzing the posterior and forward probabilities. Experiments on eukaryotic and prokaryotic proteins showing different degrees of conservation demonstrate that EasyBack outperforms TIP and BACKTRANSEQ in terms of precision (i.e. number of codons properly decoded). Consequently, sequence similarity applied to all species yields better results than imitation of codon usage within the target species.

Implementation

EasyBack is an Open-Source backtranlsation tool implemented as a Java application. The Java package JFreeChart [17] has been used to depict graphs (see Figure 1 and Figure 2 for EasyBack application interface). EasyBack system is based on a Hidden Markov Model (briefly described below).

Hidden Markov Models overview

A Hidden Markov Model (HMM) is composed of:

1. A set $S = \{S_1, S_2,\ldots,S_N\}$ of hidden states. The state at time t is denoted by q_t;

2. A set $V = \{V_1, V_2,\ldots,V_M\}$ of observation symbols;

3. A state transition probability distribution A, represented as an $N \times N$ matrix where the generic element is $a_{ij} = P[q_{t+1} = S_j | q_t = S_i]$, the probability that S_j is the state at time $t + 1$ if S_i is the state at a previous time t. Notice that $a_{ij} \geq 0$ and $\sum_{j=1}^{N} a_{ij} = 1$;

4. An observation symbol probability distribution B, represented as an $N \times M$ matrix where the generic element is $b_j(k) = P[V_k \text{ at } t | q_t = S_j]$, the probability that V_k is observed at time t in the hidden state S_j;

5. An initial state distribution π represented as a vector of which the generic element is $\pi_i = P[q_1 = S_i]$, probability that the initial state is S_i.

Given a HMM, $\lambda = (A, B, \pi)$, three basic problems arise in real applications (see [15,18] for details).

1. Given an observation sequence $O = O_1 O_2 \ldots O_T$ (where each O_t is a symbol in V), compute the most likely corresponding hidden state sequence $Q = q_1 q_2 \ldots q_T$. In this paper we deal with this problem. It can be solved by a classical Viterbi algorithm or a posterior decoding technique based on a forward-backword algorithm. Both methods are used to make prediction.

2. Given an observation sequence $O = O_1 O_2 \ldots O_T$, compute the probability $P(O|\lambda)$ of the observation O in the model λ. Together with the posterior probability, this will be used to determine the reliability of back translation.

3. Given an observation sequence $O = O_1 O_2 \ldots O_T$, tune the model's parameters in order to maximize $P(O|\lambda)$.

EasyBack

Let q be an input sequence with unknown backtranslation, and let T be the training set of sequences. The set of states of the HMM will be $S = \{s_1, s_2,\ldots,s_{64}\}$ of all possible codons. A transition from state s_i to state s_j corresponds to a pair of consecutive amino acids coded by s_i and s_j, respectively. The alphabet of the HMM comprises the 20 amino acids. The transition probability of two codons s_i and s_j is the number of occurrences of the pair of consecutive codons "$s_i s_j$" in the training set divided by the number of occurrences of s_i not followed by a stop codon. The probability that a codon s_i generates an amino acid a (emission probability) is the number of times a is decoded by s_i in the training set divided by the number of occurrences of a in such a set. Since stop codons do not encode an amino acid, then their emission probability is zero.

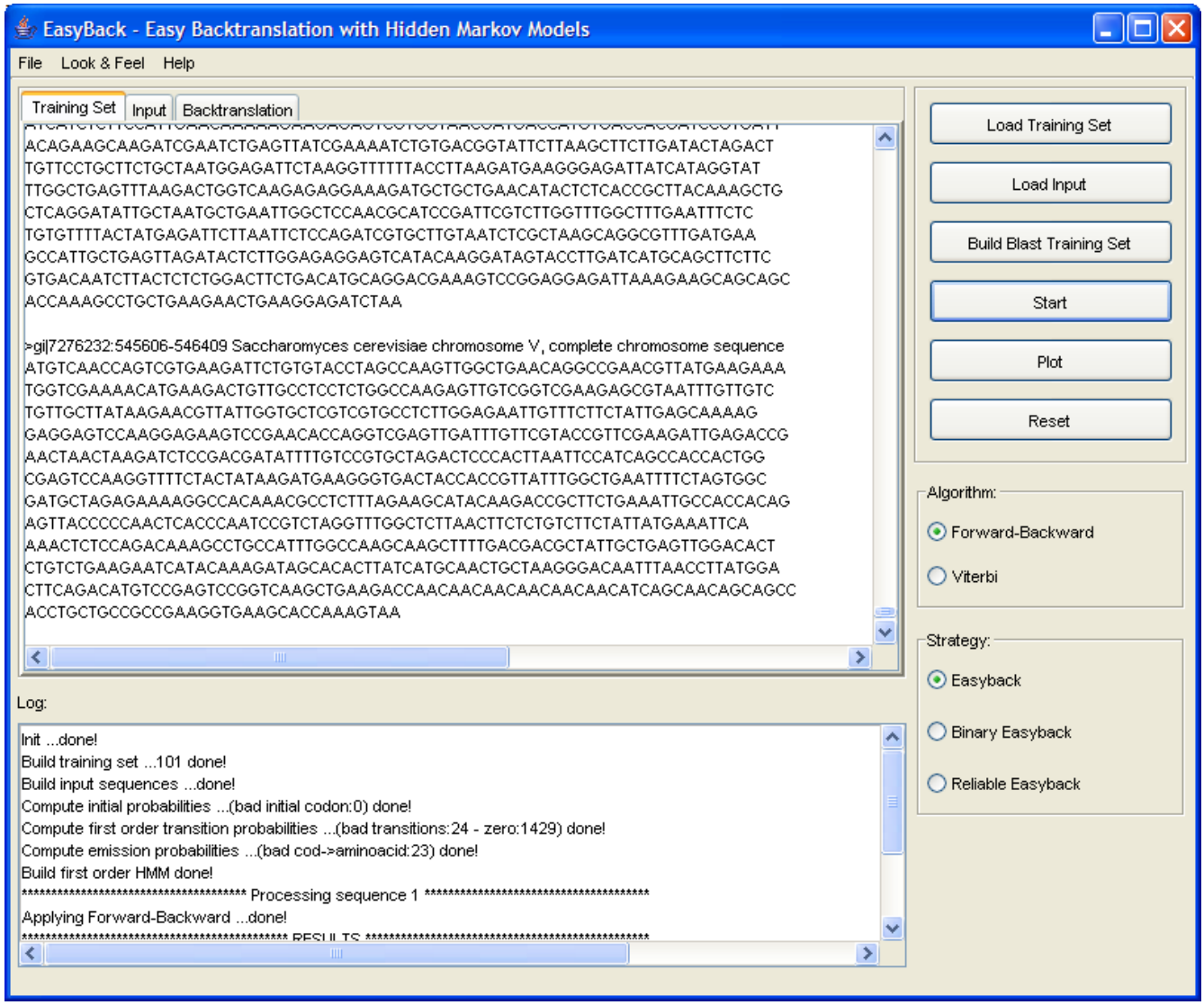

Figure 1
EasyBack main application interface.

Three different ways to apply EasyBack have been considered: *simple* (using the simple BLAST-similarity strategy), *binary* (trying to reduce the training set size), *reliable* (using forward and posterior probability diagrams to optimize prediction quality).

EasyBack uses a protein sequence to deduce cDNA (nucleotide) sequences from NCBI database. In the *simple* strategy, given a query q, a BLAST query to NCBI is performed with input q. Let T be the output of the query. The model is trained with T and eventually a prediction is returned (see Figure 3 for the pseudo-code and Figure 1 for the application interface).

In the *binary* strategy, the model is trained with the smallest set needed to make a prediction. More precisely, a BLAST query is submitted to NCBI with input q and the best 100 distinct matches are selected. Let T_1 be such a set of sequences. If the HMM fails to make a prediction with training set T_1 then add to T_1 the next best 100 (the choice of 100 matches was sufficient to make a prediction in all experimental groups of proteins, chosen with variable degree of conservation) matches, and so on until a prediction can be made. The failure condition is that for a given amino acid in the input sequence, the corresponding entry in the transition probability matrix is undefined.

Otherwise, if the HMM is able to make a prediction with T_1, then repeat the process using the best $|T_1|/2$ matches.

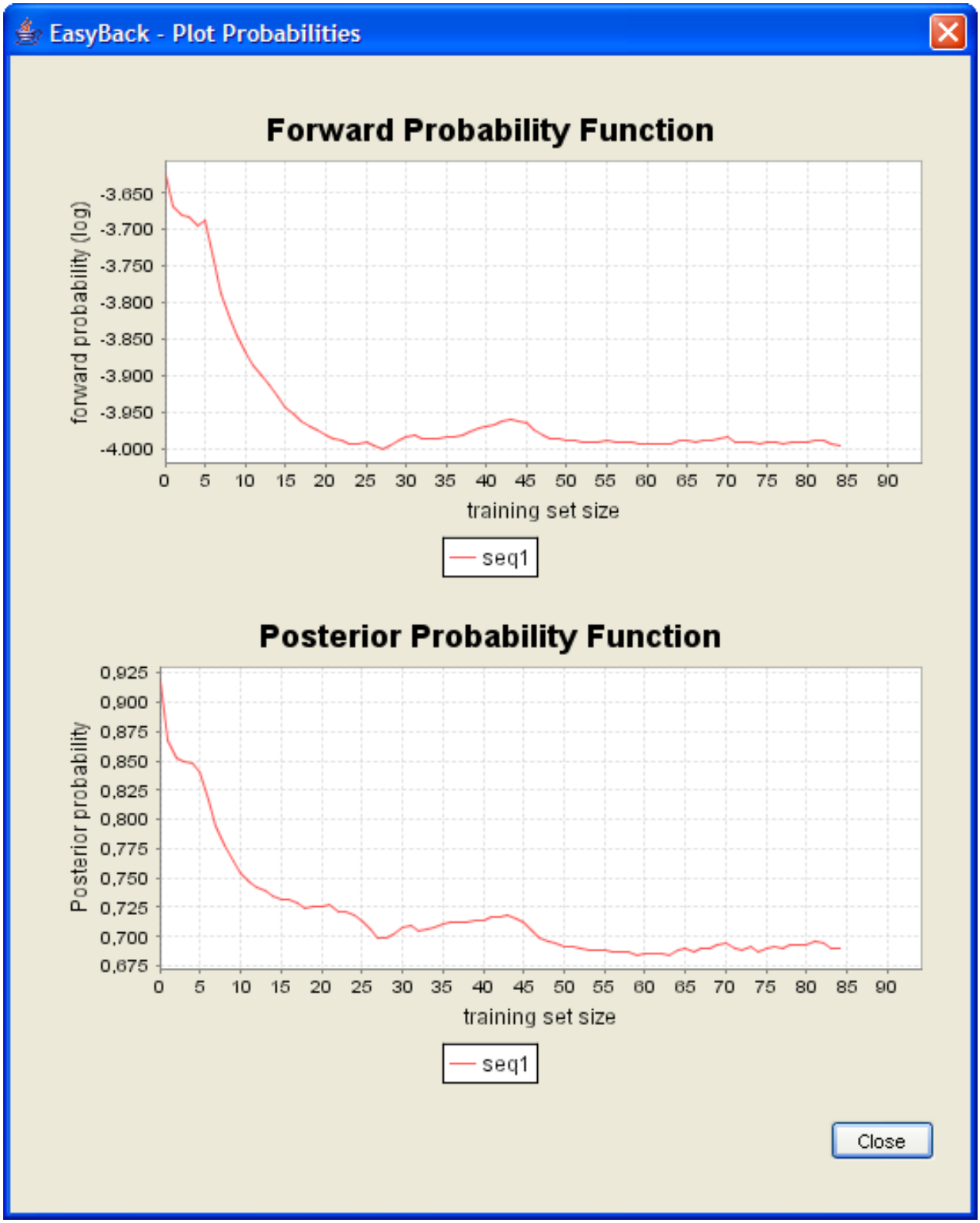

Figure 2
EasyBack interface (probabilities graphs). EasyBack computes a forward and posterior probabilities plots. Forward probability function can suggest the smallest size of the training set needed for a reliable prediction. Oscillation of the posterior probability indicates that a low percentage of amino acids has been correctly decoded.

```
Let M be a HMM model
Let q be a sequence to be backtranslated

EASY_BACK(M,q, ORGANISM)
   //biggest possible training set creation
1  T ⟵ BLAST(q,ORGANISM);
2  return EASY_BACK_CORE(M,q, T)
3  END EASY_BACK.

EASY_BACK_CORE(M,q, T)
   // train the HMM model M using the training set T
1  TRAIN(M,T);
   // make a back translation of q using M
   // by using Viterbi or Posterior decoding
2  q_back ⟵ PREDICT(M,q);
3  if (q_back = φ) then
     // prediction failed
4    return φ;
5  else
6      return q_back;
7  END EASY_BACK_CORE.
```

Figure 3
Core EasyBack algorithm. Description of EasyBack algorithm.

Let T_2 be such a set. If the HMM fails with T_2 then amend T_2 to be the best $(|T_1| + |T_2|)/2$ matches. If T_2 succeeds then amend it to the best $|T_2|/2$ matches. This binary search process stops in $O(\log(|T_1|)$ producing the final HMM prediction, which is the approximate backtranslation of the input q (see Figure 4 for the pseudo-code and Figure 1 for the application interface).

In the *reliable* strategy, a probabilistic estimation of prediction quality is made. Given a query q, a BLAST query to NCBI is performed with input q. Let T be the output of the query. The model is trained $|T|$ times, starting with a training set that contains only the first element of T and adding the next element of T iteratively. A prediction is made for each iteration and the forward and posterior probabilities are computed. The graphs of these probabilities are analyzed and the most reliable prediction is selected (see Figure 5 for the pseudo-code and Figure 1 and Figure 2 for the application interface). More precisely, the forward probability function can suggest the smallest size of the training set needed for a reliable prediction. Finally, unusual oscillation of the posterior probability indicates that a low percentage of amino acids has been correctly decoded.

Results and Discussion

Approach

EasyBack is a backtranslation tool based on a Hidden Markov Model trained with an "input-driven" training set. A HMM (for more details see [15]) describes a system comprising *N* different hidden states with transition prob-

```
Let M be a HMM model
Let q be a sequence to be backtranslated

EASY_BACK_BINARY(M,q,ORGANISM)
1  T_1 ⟵ ∅;
   //biggest possible training set creation
2  T ⟵ BLAST(q,ORGANISM);
3  size ⟵ 0;
   // repeat until a backtranslation is obtained
   // or all suitable subsets of T have been used as training set
4  do
       // set the size of the training set
5      size ⟵ min(size + 100, |T|);
       // get the first size elements in the set T
6      T_1 ⟵ GET(T, size);
7      q_back ⟵ EASY_BACK_CORE(M,q,T_1);
8  while(q_back = φ AND size < |T|);
9  if (q_back = φ) then
     // prediction failed
10   return φ;
   // in the case that a backtranslation has been found
   // a binary search strategy is used to find the minimum
   // suitable training set
11 l ⟵ 1;
12 u ⟵ |T_1|;
13 currentSolution ⟵ q_back;
14 T_2 ⟵ ∅;
15 while (l + 1 < u) do
16     step ⟵ ⌈(u−l+1)/2⌉;
17     size ⟵ l + step;
18     T_2 ⟵ GET(T_1, size);
19     q_back ⟵ EASY_BACK_CORE(M,q,T_2);
20     if (q_back = φ) then
21       l ⟵ size;
22       size ⟵ size + step;
23     else
24       currentSolution ⟵ q_back;
25       u ⟵ size;
26     end if
27 end while
28 return currentSolution;
29 END EASY_BACK_BINARY.
```

Figure 4
Binary EasyBack algorithm. Description of EasyBack algorithm with the smallest training set needed for the model to make a prediction.

```
Let M be a HMM model
Let q be a sequence to be backtranslated

 RELIABLE_EASY_BACK(M,q,ORGANISM)
 1  T_1 ⟵ ∅;
    //biggest possible training set creation
 2  T ⟵ BLAST(q,ORGANISM);
 3  size ⟵ 0;
    // repeat |T| times
 4  do
        // set the size of the training set
 5      size ⟵ size + 1;
        // get the first size elements in the set T
 6      T_1 ⟵ GET(T, size);
 7      q_back[size] ⟵ EASY_BACK_CORE(M,q,T_1);
 8      F[size] ⟵ COMPUTE_FORWARD_PROB(M,q);
 9      P[size] ⟵ COMPUTE_POSTERIOR_PROB(M,q);
10  while(size < |T|);
    // this is a supervised step performed by the user
11  q_back^best ⟵ analyze the functions F and P to get the most reliable prediction q_back;
12  return q_back^best;
13  END RELIABLE_EASY_BACK.
```

Figure 5
Reliable EasyBack algorithm. Description of EasyBack algorithm in which forward and posterior probabilities are stored and analyzed to determine the most reliable backtranslation.

abilities associated with each pair of states. The states generate observable symbols with probabilities computed from a training set. Given a series of observable symbols, the HMM can decode the most probable corresponding sequence of hidden states. In the proposed model, the hidden states are all possible codons and the observable symbols are the amino acids decoded by them. The transition probability of two codons s_i and s_j is the number of occurrences of the pair of consecutive codons $s_i s_j$ in the training set divided by the number of the occurrences of s_i not followed by a stop codon. The probability that a codon s_i generates an amino acid a, the emission probability, is the number of times a is decoded by s_i in the training set divided by the number of occurrences of a in such a set. The training set is constructed by applying a criterion of similarity between the input protein sequence q to be backtranslated and sequences in the NCBI database. More precisely, a BLAST query is submitted to NCBI with input q and the "smallest" subset of the query output that enables HMM to make a prediction is chosen as the training set. Therefore, the size of the training set is related to the number of non-zero values contained in the matrix of transition probabilities. More precisely, when the system fails to make a prediction, this means that at least one necessary transition probability value in the matrix is zero. In this case the training set must be enlarged with more sequences. The backtranslation of q is obtained by applying either the Viterbi or the Forward-Backward algorithm to the model (posterior decoding) [15]. One useful aspect of HMM is the ability to choose several strategies for posterior estimation of the reliability of a prediction (e.g. see [19] for multiple sequence alignment). The forward probability function can suggest the size of the smallest training-set needed for reliable prediction. The higher this probability, the better the prediction obtained from the training set. Furthermore, analysis of the posterior probability allows the quality of prediction to be established. More precisely, if the probability oscillates unusually as a function of the training set size, then a low percentage of amino acids has been correctly decoded.

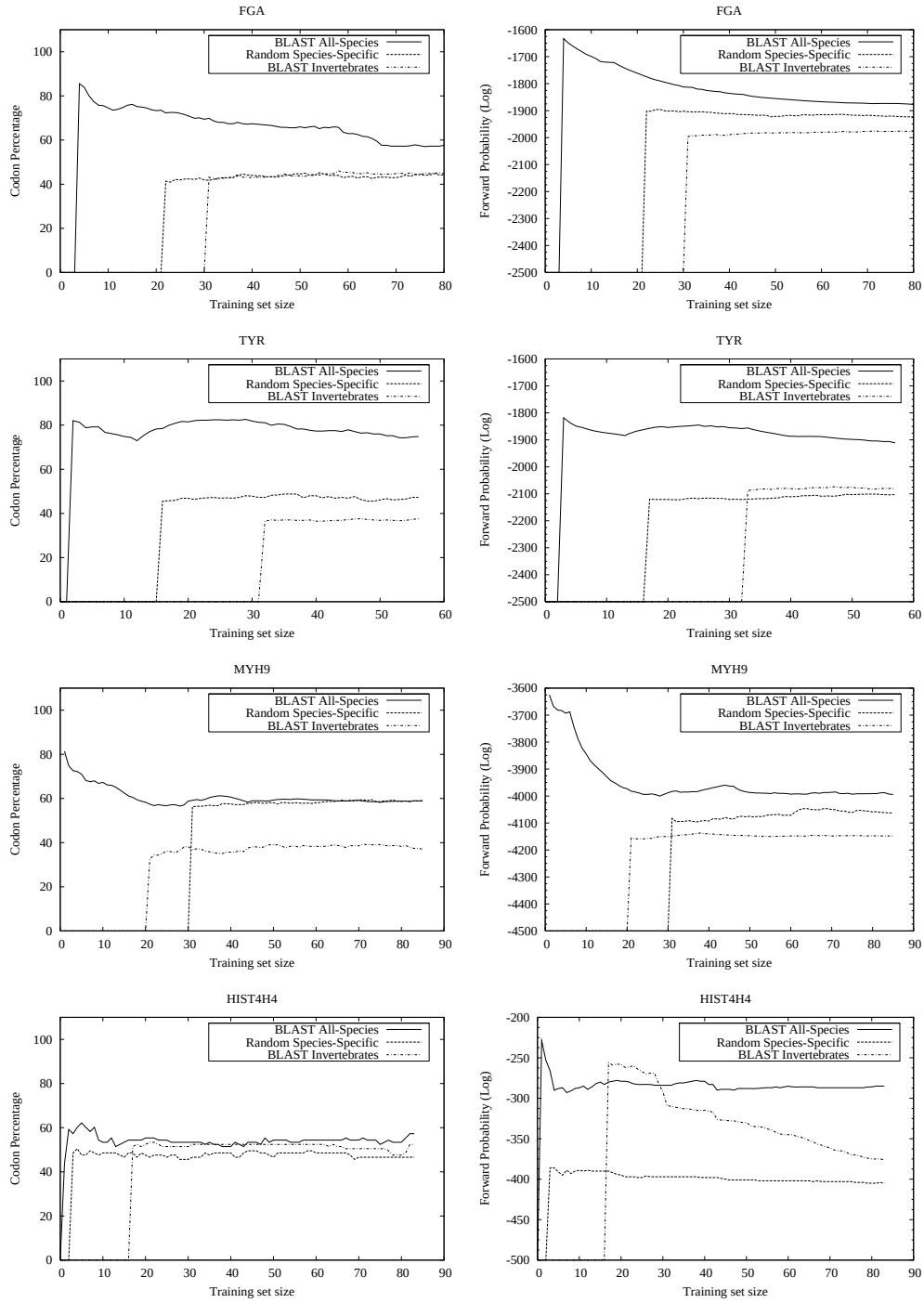

Figure 6

EasyBack Performace Analysis on FGA, TYR, MYH9 and HIST4H4. Left column: EasyBack prediction performance (percentage of amino acids correctly decoded). Input proteins are: FGA (fibrinopeptide), TYR (tyrosinase), MYH9 (myosin), HIST4H4 (histone H4). Right column: forward probability. The quality of prediction using *BLAST All-Species* training sets is higher than both *Random Species-Specific* (sequences belonging to the same organism) and *BLAST Invertebrates* (distant organisms). The forward probability can be used to estimate the best training set size. In almost all cases a high forward probability corresponds to a high quality backtranslation.

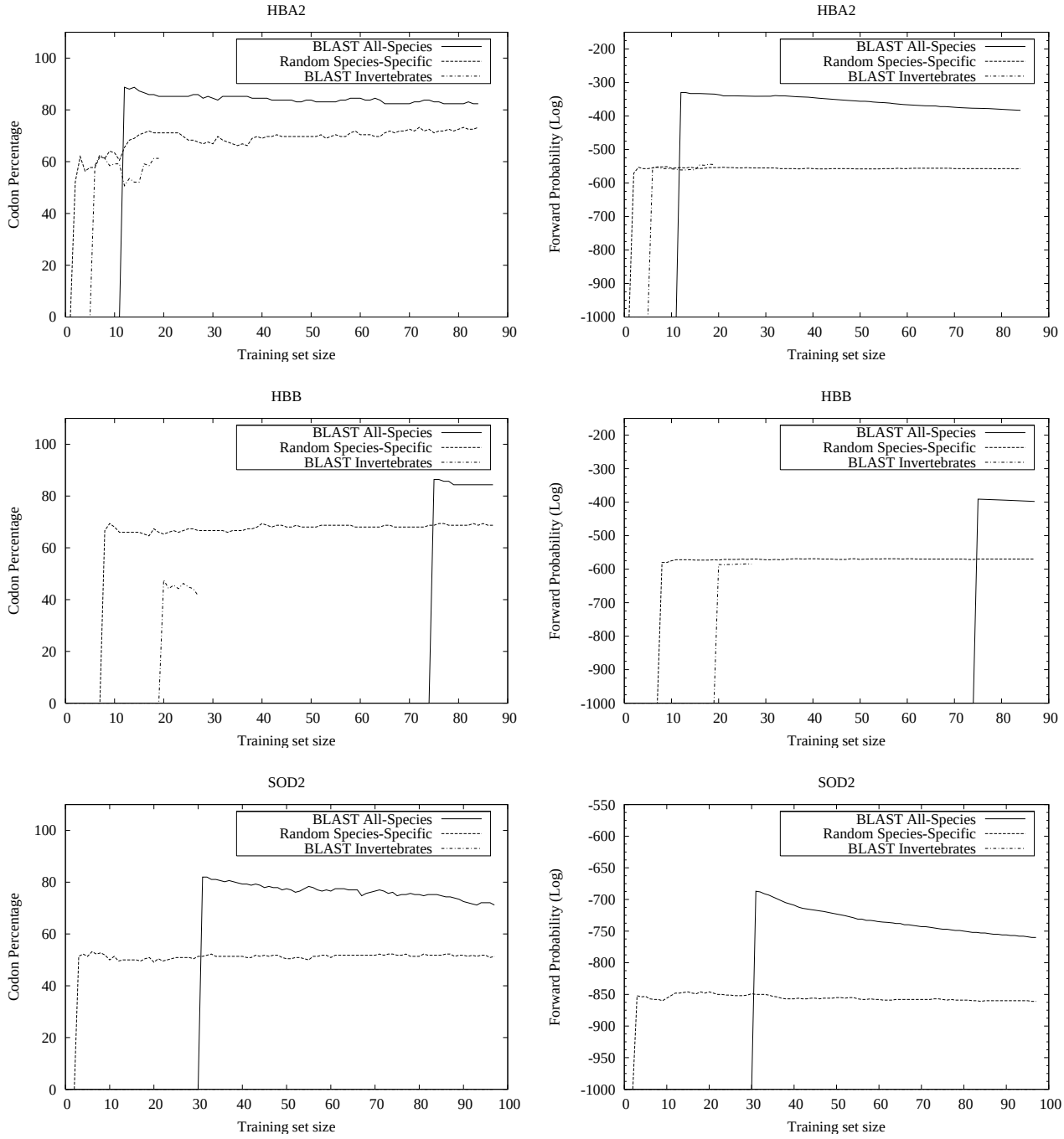

Figure 7

EasyBack Performace Analysis on HBA2, HBB and SOD2. Left column: EasyBack prediction performance (percentage of amino acids correctly decoded). Right column: forward probability. (See caption of Figure 6). Input proteins are: HBA2 (alpha globin), HBB (beta globin), and SOD2.

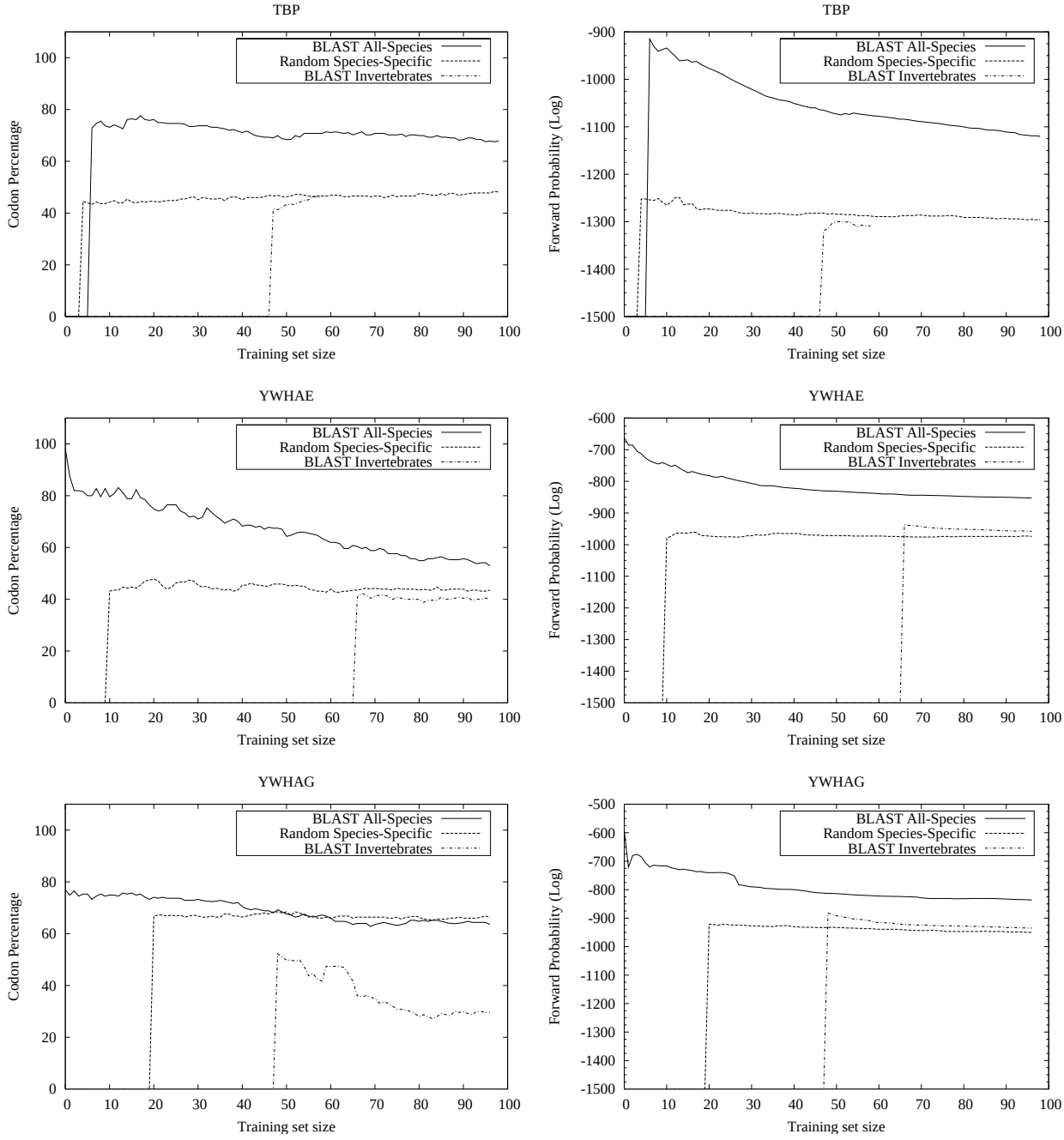

Figure 8

EasyBack Performace Analysis on YWHAE, YWHAG and TBP. Left column: EasyBack prediction performance (percentage of amino acids correctly decoded). Right column: forward probability. (See caption of Figure 6). Input proteins are: YWHAE (NP_036611.2), YWHAG (NP_006752.1), and TBP.

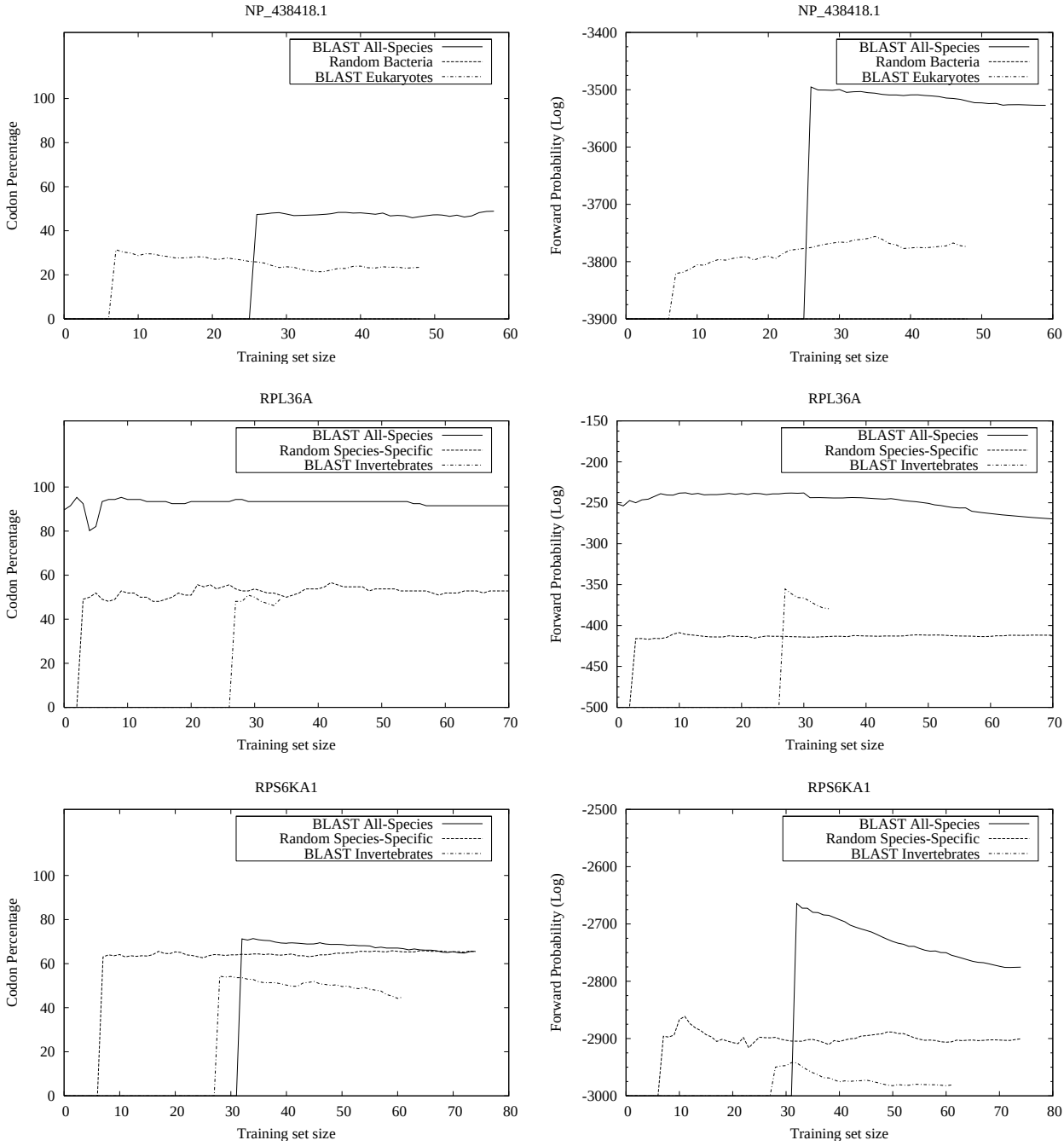

Figure 9

EasyBack Performace Analysis on NP_438418.1, RPL36A and RPS6KA1. Left column: EasyBack prediction performance (percentage of amino acids correctly decoded). Right column: forward probability. The quality of prediction using *BLAST All-Species* training sets is higher than both *Random Species-Specific* (sequences belonging to the same organism) and *BLAST Invertebrates* for RPL36A and RPS6KA1 and *BLAST Eukaryotes* for NP_438418.1 (distant organisms). The forward probability can be used to estimate the best training set size. In almost all cases a high forward probability corresponds to a high quality back-translation. Input proteins are: NP_438418.1 (from *Haemophilus influenzae* species), RPL36A (ribosomal protein L36a), RPS6KA1 (ribosomal protein S6 kinase, 90 kDa).

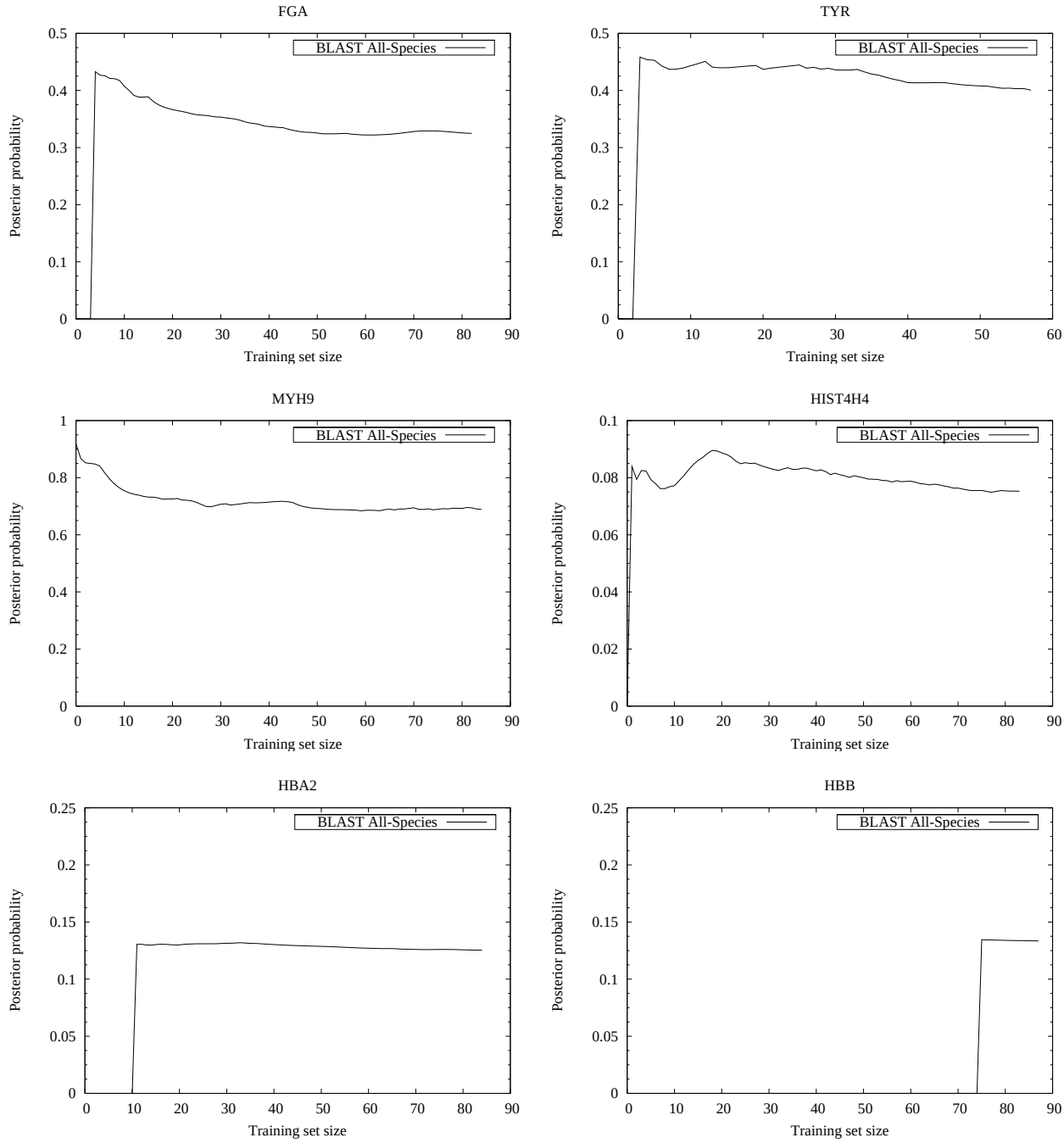

Figure 10

EasyBack posterior probabilities. The oscillating behavior of the posterior probability of histone H4 corresponds empirically to the low quality of its backtranslation (see the graph reporting the correctly decoded codon percentage of HIST4H4 in Figure 6).

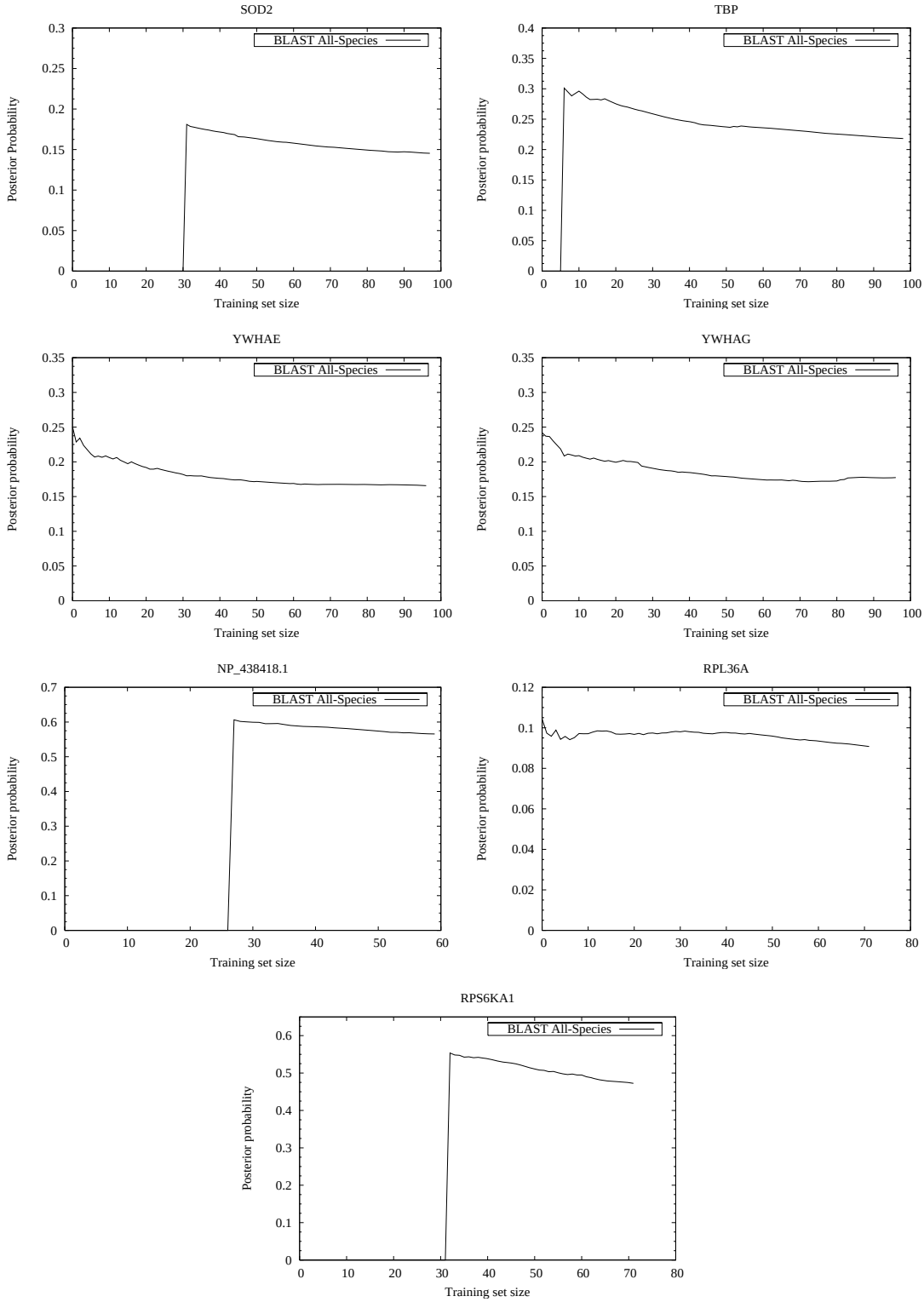

Figure 11

EasyBack posterior probabilities. In these cases non-oscillating behavior is reported. This is associated with high quality backtranslation (see graphs reporting their correctly decoded codon percentages in Figure 7, Figure 8 and Figure 9).

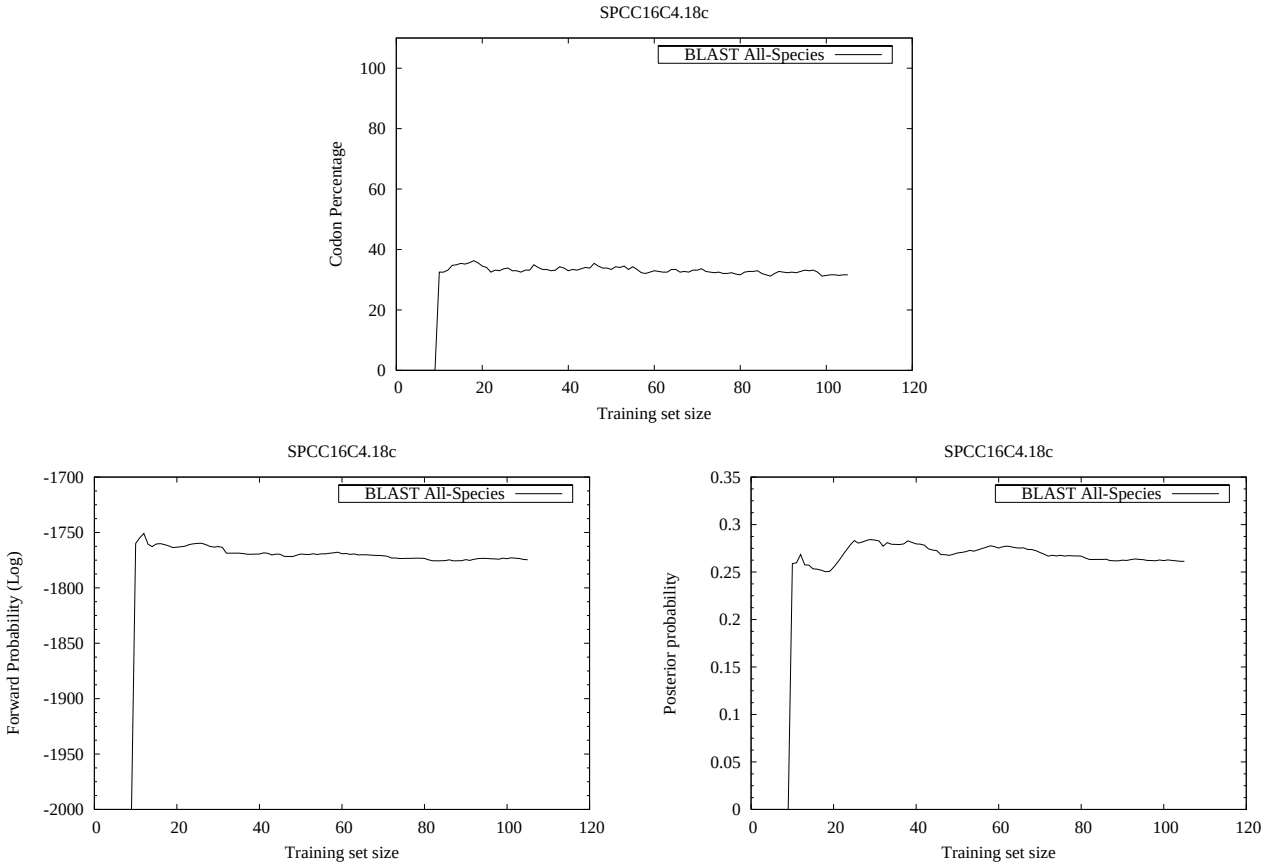

Figure 12
EasyBack Performace Analysis on SPCC16C4.18c. SPCC16C4.18c shows oscillating posterior probabilities corresponding to low quality decoding.

Test sets

To assess the efficiency of the proposed method, a set of *Homo sapiens* and prokaryotic proteins with various degrees of primary structure conservation was backtranslated (the conservation degree of the experimental set of proteins was obtained by calculating the proportion of amino acid sites at which the two sequences under study were identical [20]):

• Proteins present in all eukaryotes: histone H4 (HIST4H4) (97.7%), SOD2 [21] (67.1%), NP_006752.1 (YWHAG), and NP_036611.2 (YWHAE).

• Proteins present in all metazoa: TBP (81%), fibrinopeptide (FGA) (60.9%), and myosin (MYH9) (59.7%).

• Proteins present only in vertebrates: tyrosinase (TYR) (73.9%), alpha globin (HBA2) (65.2%), and beta globin (HBB) (62.5%).

• Proteins present in prokaryotes: NP_438418.1 (*Haemophilus influenzae*).

• Ribosomal proteins: ribosomal protein L36a (RPL36A), ribosomal protein S6 kinase, 90 kDa (RPS6KA1).

EasyBack was trained with three different kinds of training sets:

• *BLAST All-Species*. This training set was obtained by querying the NCBI all-species database with the input sequence, using BLAST;

• *Random Species-Specific*. This training set was obtained by randomly choosing sequences that belong to species expressing the input protein;

Table 1: Comparison of existing backtranslation tools.

	Training Set	Learning Method	Unsupervised
BBOCUS [10]	Species specific	Clustering and codon frequencies	No
EasyBack	**Any**	**HMM**	**Yes**
BACKTRANSEQ [11]	Species specific	Codon frequencies	Yes
LBT [14]	Any	Alignment and local codon frequencies	Yes
TIP [12]	Any	Genetic algorithm and statistics	Yes

• *BLAST Invertebrates*: for *Homo sapiens* proteins, the training set was obtained by querying the NCBI invertebrates database with the input sequence, using BLAST;

• *BLAST Eukaryote*: for prokaryotic proteins, the training set was obtained by querying the NCBI eukaryote database with the input sequence, using BLAST.

Since biological sequence databases are notorious for having multiple copies of sequence fragments in different entries, homologous found with BLAST that contained portions of the sequence under test were carefully manually eliminated to make the testing process fair. On the other hand this manual filtering is not necessary for an unknown input amino acid sequence. This was the reason for not considering *BLAST Species-Specific* training sets (insufficient numbers of sequences were returned). *BLAST Species-Specific* training set was obtained by querying with the input sequence, using BLAST, the sequences of NCBI database belonging to species of the input protein.

The results show that EasyBack clearly performs better, in terms of percentage of correctly decoded codons, when trained with *BLAST All-Species* (see left column of Figures 6, 7, 8, 9). However, the prediction quality is degraded if sequences belonging to a distantly-related organism are chosen as training set (e.g. Homo sapiens SOD2 on Invertebrates data set, Bacteria NP_438418.1 on Eukaryotes). Moreover, HMM trained only with sequences from organisms other than the one from which the sequence under test was obtained showed no decrease in prediction quality (these experiments are not reported here since the performance was very close to that with *BLAST All-Species*).

The results summarized in the right column of Figure 6 and in Figure 7, 8, 9 show that, for all cases except HISTH4, the most reliable prediction is obtained using the training set with the highest forward probability. Moreover, the quality of the prediction can be estimated by analyzing posterior probability. The unusual oscillation of posterior probability in Figure 10 and 11 for Histon H4 and Figure 12 for SPCC16C4.18c from Schizosaccharomyces pombe indicates that only a low percentage of the amino acids were correctly decoded.

Despite experiments show that similarity is more relevant than species specificity, a reliable prediction depends on how the training set is "biologically related" to the input sequence. Acquiring knowledge able to correlate the quality of prediction to the composition of the training set is a hard problem and will be subject of future research. For example, prediction quality for RPL36A was significantly higher than Hist4H4. On the other hand for both proteins prediction quality did not decrease by augmenting the

Table 2: Comparisons of EasyBack with TIP and BACKTRANSEQ based on percentages of amino acids correctly decoded.

Test ID	EasyBack-FB	EasyBack-Vit	TIP	BACKTRANSEQ
YWHAG	0.64	**0.66**	0.53	0.66
YWHAE	0.53	**0.55**	0.37	0.38
HIST4H4	**0.57**	0.55	0.46	0.48
TBP	**0.68**	**0.68**	0.43	0.44
SOD2	**0.72**	0.70	0.44	0.49
MYH9	**0.59**	0.58	0.45	0.53
TYR	0.75	**0.76**	0.45	0.39
HBA2	0.82	**0.83**	0.70	0.75
HBB	**0.84**	**0.84**	0.63	0.59
FGA	**0.57**	0.54	0.35	0.39
RPL36A	**0.92**	**0.92**	0.52	0.51
RPS6KA1	**0.66**	0.65	0.42	0.59
NP_438418.1	**0.49**	0.47	0.05	0.18

Easyback and TIP were tested using *BLAST All-Species* training sets. BACKTRANSEQ used Species-Specific training sets. For all systems, each training set comprised 100 sequences. EasyBack-FB and EasyBack-Vit denote Forward-Backward and Viterbi, respectively. See Figure 13.

Table 3: Comparisons of EasyBack with TIP and BACKTRANSEQ based on the percentages of amino acids correctly decoded.

Test ID	EasyBack-FB	EasyBack-Vit	TIP	BACKTRANSEQ
YWHAG	**0.77**	**0.77**	0.58	0.66
YWHAE	**0.98**	**0.98**	0.51	0.38
HIST4H4	0.44	0.44	0.45	**0.48**
TBP	**0.73**	**0.73**	0.43	0.44
SOD2	0.82	**0.83**	0.49	0.49
MYH9	**0.81**	0.80	0.49	0.53
TYR	**0.82**	0.81	0.46	0.39
HBA2	**0.89**	0.88	0.63	0.75
HBB	**0.86**	**0.86**	0.59	0.59
FGA	**0.86**	**0.86**	0.47	0.39
RPL36A	**0.83**	**0.83**	0.51	0.51
RPS6KA1	**0.71**	**0.71**	0.5	0.59
NP_438418.1	**0.47**	**0.46**	0.05	0.18

The training set is the minimal subset of the query output sufficient to make a prediction obtained by a binary search strategy. EasyBack-FB and EasyBack-Vit denote Forward-Backward and Viterbi, respectively. See Figure 14.

training set. The mathematical explanation of this phenomenon can by expressed in terms of a better agrement in RPL36A vs Hist4H4 in the Markovian codon transition/emission probabilities among the elements in the training set. In any case EasyBack is able to estimate prediction quality and optimal training set size by forward and posterior probability computation respectively.

Comparisons

EasyBack was successfully compared with TIP [12] and BACKTRANSEQ [11] (see Table 1 for details). In all the experiments described below, the same training sets obtained using *BLAST All-Species* criteria for EasyBack and TIP were used. In contrast BACKTRANSEQ was designed to be used only with *Species-Specific* training sets (each amino acid is decoded by the most frequent coding codon in the species). In the first comparison (Figure 13, Table 2) a training set of a fixed size (100 sequences) was used. In the second comparison (Figure 14, Table 3), the binary strategy procedure described in the Methods section was applied to generate the "smallest" training set needed for prediction. EasyBack, TIP and BACKTRANSEQ were also compared using species-specific training sets. For TIP and EasyBack, the training sets were chosen randomly; for BACKTRANSEQ, the most frequent codon criterion was used. The results show that species-specific training sets give lower-quality predictions. Once again, EasyBack outperformed TIP and BACKTRANSEQ. Moreover, a statistical analysis was performed to support the quality of EasyBack predictions. Table 4 contains Friedman Rank test for all pairwise comparisons of EasyBack, TIP and BACKTRANSEQ. Moreover, a statistical analysis was performed to support the quality of EasyBack predictions. Table 4 contains Friedman rank test for all pairwise comparisons of EasyBack, TIP and BACKTRANSEQ.

Conclusion

In this paper, a backtranslation tool using a Hidden Markov Model, trained with a set of sequences most similar to the input, has been shown to outperform other published methods. All-species similarity gives better results than species-specific similarity. Furthermore, the proposed system is parameter-free and fully automated

Table 4: Significance test for differences in experiments reported in Figures 13 and 14.

	EasyBack-FB	EasyBack-Vit	TIP	BACKTRANSEQ
EasyBack-FB	-	-(0.57)	+0.0003	+0.0023
EasyBack-Vit	-(0.29)	-	+0.0003	+0.0008
TIP	-0.0023	-0.0023	-	-(0.05)
BACKTRANSEQ	-0.0023	-0.0023	+(0.4)	-

Entries show the *p*-values indicating the significance of comparisons between two backtranslation methods using the Friedman rank test. Entries above the diagonal refer to experiments in Figure 13. Entries below the diagonal refer to experiments in Figure 14. The (+) method on the left had lower average rank (better performance); the (-) method had higher average rank (worse performance); parentheses denote non-significant *p*-values > 0.05.

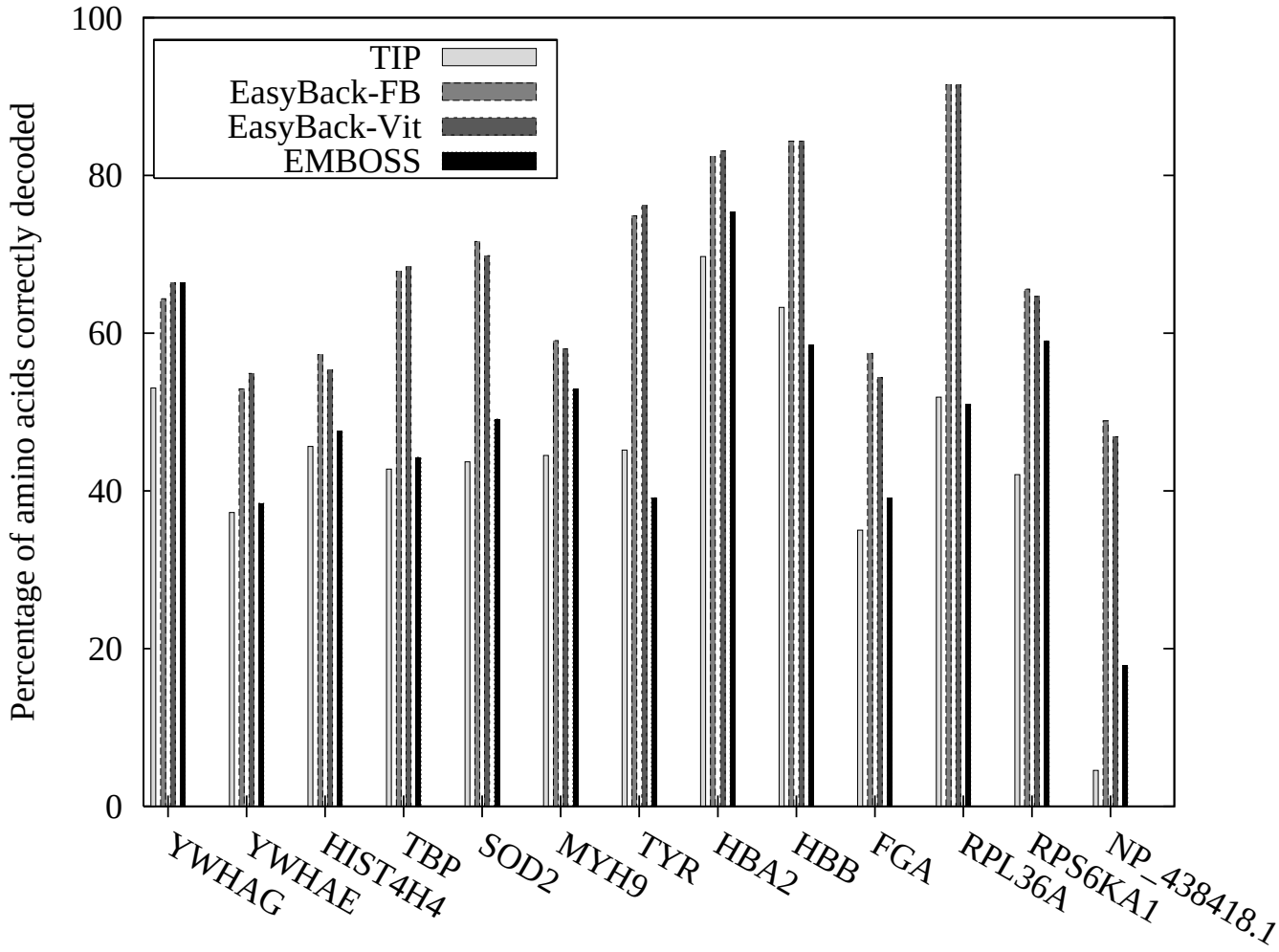

Figure 13

EasyBack vs TIP and BACKTRANSEQ. Performance of EasyBack compared with TIP and BACK-TRANSEQ based on percentages of amino acids correctly decoded. Easyback and TIP were tested using *BLAST All-Species* training sets. BACK-TRANSEQ used Species-Specific training sets. For all systems each training set comprised 100 sequences. Classical Viterbi algorithm (EasyBack-Vit) and a posterior decoding technique based on a forward-backword algorithm (EasyBack-FB) were used to make predition.

and allows the quality of prediction to be estimated (that is a clear advantage of the proposed method).

The results demonstrate that the performance of Easy-Back, in terms of the percentage of amino acids correctly decoded, is considerably better than compared systems.

Availability and requirements

• **Project name**: EasyBack

• **Project home page**: http://alpha.dmi.unict.it/~ctnyu/easyback.html

• **Operating system(s)**: e.g. Platform independent

• **Programming language**: Java

• **Other requirements**: Java 1.5.0_05 or higher

• **License**: Free for academic and commercial users under the GNU Lesser General Public License (LGPL)

Authors' contributions

CDP, MP and MR proposed the problem and provided the test input sequences. AF, RG, GP and AP designed, analyzed, implemented and tested the proposed algorithm. Each author contributed equally in writing the paper. All authors read and approved the final manuscript. Authors of each department are listed in alphabetic order.

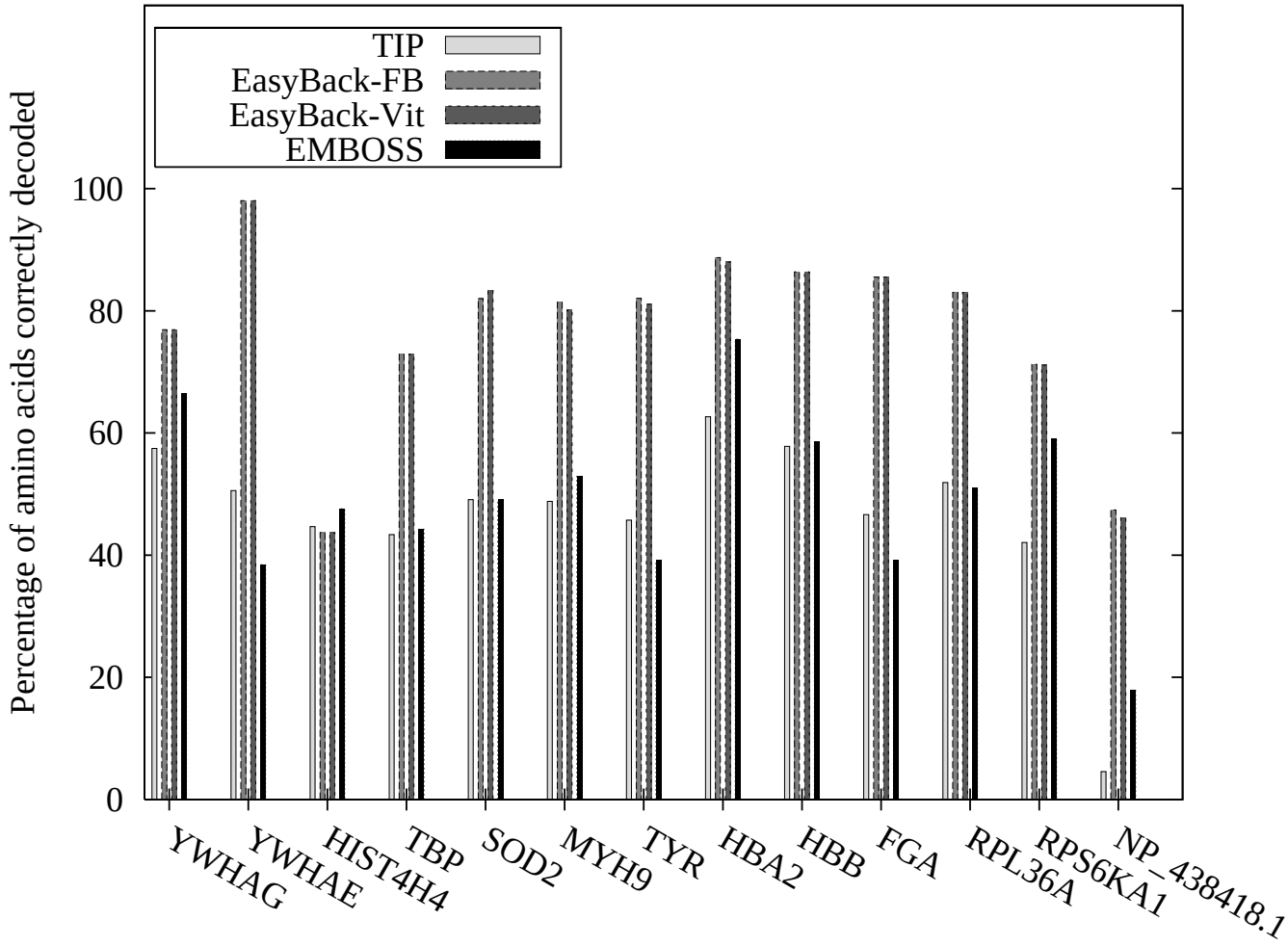

Figure 14

EasyBack vs TIP and BACKTRANSEQ (binary search strategy). Performance of EasyBack compared with TIP and BACKTRANSEQ based on the percentages of amino acids correctly decoded. The training set is the minimal subset of the query output sufficient to make a prediction obtained by a binary search strategy. EasyBack-FB and EasyBack-Vit denote Forward-Backward and Viterbi, respectively.

Acknowledgements

We thank the anonymous reviewers for their constructive comments. Moreover we thank all the users who have downloaded our software and contributed to its improvement. We would like to thank Dr. D. Skripin for his contribution in generating training sets. Authors were in part supported by PROGETTO FIRB ITALY-ISRAEL grant n. RBIN04BYZ7: "Algorithms for Patterns Discovery and Retrieval in discrete structures with applications to Bioinformatics" and by Sicily Region grant PROGETTO POR 3.14: "Ricerca e Sviluppo suite di programmi per l'analisi biologica, denominata: BIOWARE".

References

1. **Software SGD** [http://www.evolvingcode.net/codon/]
2. Richardson S, Wheelan S, Yarrington R, Boeke J: **GeneDesign: rapid, automated design of multikilobase synthetic genes.** *Genome Res* 2006, **16:**550-556.
3. Villalobos A, Ness J, Gustafsson C, Minshull J, Govindarajan S: **Gene Designer: a synthetic biology tool for constructing artificial DNA segments.** *BMC Bioinformatics* 2006, **7:**285.
4. Supek F, Vlahoviček K: **Comparison of codon usage measures and their applicability in prediction of microbial gene expressivity.** *BMC Bioinformatics* 2005, **6:**182.
5. Grantham R, Gautier C, Gouy M, Jacobzone M, Mercier R: **Codon catalog usage is a genome strategy modulated for gene expressivity.** *Nucleic Acids Res* 1981, **9:**r43-r74.
6. Grantham R, Gautier C, Gouy M, Mercier R, Pave A: **Codon catalog usage and the genome hypothesis.** *Nucleic Acids Res* 1980, **8(1):**r49-r62.
7. Gu W, Zhou T, Ma J, Sun X, Lu Z: **The relationship between synonymous codon usage and protein structure in *Escherichia coli* and *Homo sapiens*.** *Biosystems* 2004, **73:**89-97.
8. Sharp P, Cowe E, Higgins D: **Codon usage patterns in *Escherichia coli, Bacillus subtilis, Saccharomyces cerevisiae, Schizosaccharomyces pombe, Drosophila melanogaster* and *Homo sapiens*: a review of the considerable within-species diversity.** *Nucleic Acids Res* 1988, **16:**8207-8211.
9. Sharp P, Tuohy T, Mosurski K: **Codon usage in *yeast*: cluster analysis clearly differentiates highly and lowly expressed genes.** *Nucleic Acids Res* 1986, **14:**5125-5143.
10. Pesole G, Attimonelli M, Liuni S: **A backtranslation method based on codon usage strategy.** *Nucleic Acids Res* 1988, **15(5):**1715-1728.

11. Rice P, Longden I, Bleasby A: **EMBOSS: the european molecular biology open software suite.** *Trends Genet* 2000, **16(6):**276-277.
12. Moreira A, Maass A: **TIP: protein backtranslation aided by genetic algorithms.** *Bioinformatics* 2004, **20(13):**2148-2149.
13. Moreira A: **Genetic algorithm for the imitation of genomic styles in protein backtranslation.** *Theoretical Computer Science* 2004, **322(2):**297-312.
14. Giugno R, Pulvirenti A, Ragusa M, Facciola L, Patelmo L, Pietro VD, Pietro CD, Purrello M, Ferro A: **Locally sensitive backtranslation based on multiple sequence alignment.** *Proceeding of the IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology San Diego, CA* 2004:231-237.
15. Rabiner L: **A tutorial on Hidden Markov Models and selected applications in speech recognition.** *Proceedings of The IEEE* 1989, **77(2):**257-286.
16. **NCBI** [http://www.ncbi.nlm.nih.gov/]
17. **JFreeChart** [http://jfree.org/jfreechart/]
18. Durbin R, Eddy S, Krogh A, Mitchison G: *Biological sequence analysis: probabalistic models of proteins and nucleic acids* Cambridge: Cambridge University Press; 1998.
19. Do CD, Mahabhashyam MSP, Brudno M, Batzoglou S: **ProbCons: probabilistic consistency-based multiple sequence alignment.** *Genome Res* 2005, **15:**330-340.
20. Nei M, Kumar S: **Prospects for inferring very large phylogenies by using the neighbor-joining method.** *Proc Natl Acad Sci USA* 2004, **101(30):**11030-11035.
21. Purrello M, Pietro CD, Ragusa M, Pulvirenti A, Giugno R, Pietro VD, Emmanuele G, Travali S, Scalia M, Shasha D, Ferro A: **In vitro and in silico cloning of *Xenopus laevis* SOD2 and its phylogenetic analysis.** *DNA and Cell Biol* 2005, **24(2):**111-116.