# Learning Chomsky-like grammars for biological sequence families

Muggleton, SH, Bryant, CH and Srinivasan, A

# Learning Chomsky-like Grammars for Biological Sequence Families

**S.H. Muggleton** STEPHEN@CS.YORK.AC.UK
**C.H. Bryant** BRYANT@CS.YORK.AC.UK
Computer Science, University of York, York YO10 5DD, UK

**A.Srinivasan** ASHWIN@COMLAB.OXFORD.AC.UK
Oxford Uni. Computing Lab., Wolfson Building, Oxford OX1 3QD, UK

## Abstract

This paper presents a new method of measuring performance when positives are rare and investigates whether Chomsky-like grammar representations are useful for learning accurate comprehensible predictors of members of biological sequence families. The positive-only learning framework of the Inductive Logic Programming (ILP) system CProgol is used to generate a grammar for recognising a class of proteins known as human neuropeptide precursors (NPPs). As far as these authors are aware, this is both the first biological grammar learnt using ILP and the first real-world scientific application of the positive-only learning framework of CProgol. Performance is measured using both predictive accuracy and a new cost function, *Relative Advantage* ($RA$). The $RA$ results show that searching for NPPs by using our best NPP predictor as a filter is more than 100 times more efficient than randomly selecting proteins for synthesis and testing them for biological activity. The highest $RA$ was achieved by a model which includes grammar-derived features. This $RA$ is significantly higher than the best $RA$ achieved without the use of the grammar-derived features.

## 1. Introduction

This paper presents a new method of measuring performance when positives are rare and attempts to answer, by way of a case-study, the question of whether grammatical representations are useful for learning from biological sequence data. We address the question by refuting the following null hypothesis.

**Null hypothesis:** The most accurate comprehensible multi-strategy predictor in our study does not employ Chomsky-like grammar representations.

The performance of each model is measured using a new cost function, *Relative Advantage* ($RA$). Section 2 defines $RA$ and explains why it is used in preference to predictive accuracy.

The domain of the case study is the recognition of a class of proteins known as human neuropeptide precursors (NPPs). These proteins have considerable therapeutic potential and are of widespread interest in the pharmaceutical industry. Our most accurate comprehensible multi-strategy predictor of NPPs employs a Chomsky-like grammar representation.

Multi-strategy learning (Michalski & Wnek, 1997) aims at integrating multiple strategies in a single learning system, where strategies may be inferential (e.g. induction, deduction etc) or computational. Computational strategy is defined by the representational system and the computational method used in the learning system (e.g. decision tree learning, neural network learning etc).

We refute the null hypothesis as follows. A grammar is generated for a particular class of biological sequences. A group of features is derived from this grammar. Other groups of features are derived using other learning strategies. Amalgams of these groups are formed. A recognition model is generated for each amalgam using C4.5 and C4.5rules. The null hypothesis is refuted because:

1. the best performance achieved using any of the models which include grammar-derived features is higher than the best performance achieved using any of the models which do not include the grammar-derived features;
2. this increase is statistically significant;

3. the best model which includes grammar-derived features is sufficiently more comprehensible than the best 'non-grammar' model.

## 2. Relative Advantage

NPPs are identified either through purely biological means or by screening genomic or protein sequence databases for likely NPPs, followed by biological evaluation. If we wish to go beyond using sequence homology to find new members of the (generally small) NPP families, we need a recognition model for NPPs in general. However if this recognition model is poor then it may not be much better than random sampling of sequence databases and the cost-benefit of any experimental evaluation of NPPs found by such a procedure would be prohibitively small.

In developing a general recognition model for human NPPs, we are faced with three significant obstacles.

1. The number of known NPPs in the public domain databases of protein sequence (e.g. SWISS-PROT (Emmert et al., 1994)) is very small in proportion to the total number of sequences. When we developed our method of estimating RA (May 1999), SWISS-PROT contained 79,449 sequences, of which some 57 could definitely be identified as human NPPs.

2. There is no guarantee that all the human NPPs in SWISS-PROT have been properly identified. We estimate there may, in fact be up to 90 NPPs in SWISS-PROT.

3. There is no benchmark method for NPP recognition that can be used to compare any new methods. We must therefore compare our recognition model with random sampling to evaluate success.

This domain requires a performance measure which addresses all of these issues. For domains in which positives are rare, predictive accuracy, as it is normally measured in Machine Learning (assuming equal misclassification costs):

- gives a poor estimate of the performance of a recognition model. For instance, if a learner induces a very specific model for such a domain, the predictive accuracy of the model may be very high despite the number of true positives being very small or even zero.

- does not discriminate well between models which exclude most of the (abundant) negatives but cover varying numbers of (the rare) positives. (This is illustrated later in this paper - see Table 5.)

*Table 1.* $2 \times 2$ Contingency table for the test set. The axes of the $2 \times 2$ matrix are labelled by the sets NPP sequences, Random sequences, $H$ (Hypothesis predictions) and $\overline{H}$ (complement of H). The cells of the matrix represent the cardinalities of the corresponding intersections of these sets. $n_1 + n_2 + n_3 + n_4 = n$, where $n$ is the number of instances in the test set.

| | Set of test NPP sequences | Set of test Random sequences |
|---|---|---|
| $H$ | $n_1$ | $n_2$ |
| $\overline{H}$ | $n_3$ | $n_4$ |

Therefore we define a *relative advantage* ($RA$) function which predicts the reduction in cost in using the model versus random sampling. In contrast to other performance measures, $RA$ is meaningful and relevant to experts in the domain.

### 2.1 Definition of $RA$

In the following, 'the model' refers to a recognition model for predicting whether a sequence is a NPP. RA can be defined in terms of probability as follows.
Let C = the cost of testing the biological activity of one protein via wet-experiments in the laboratory;
NPP = Sequence is a NPP;
Rec = Model recognises sequence as a NPP.

$$RA = \frac{C/Pr(NPP)}{C/Pr(NPP \mid Rec)} = \frac{Pr(NPP \mid Rec)}{Pr(NPP)} \quad (1)$$

Let testing the model on test data yield the $2 \times 2$ contingency table shown in Table 1 with the cells $n_1$, $n_2$, $n_3$, and $n_4$. Let $n = n_1 + n_2 + n_3 + n_4$ be the number of instances in the test set. If the proportion of NPPs in the test set was known to be the same as the proportion of NPPs in the database then we could estimate $Pr(NPP)$ to be $(n_1 + n_3)/n$ and $Pr(NPP \mid Rec)$ to be $n_1/(n_1 + n_2)$. These estimates cannot be used with our method because we cannot assume that the proportion of NPPs is the same in the test set and database.

In order to derive a formula for estimating RA given both a set of positives and a set of randoms, we estimate $Pr(NPP)$ and $Pr(NPP \mid Rec)$ as follows. Let $S$ be the total number of sequences in the database, of which $M$ are NPPs.

$$Pr(NPP) = \frac{no.\ of\ NPPs\ in\ the\ database}{no.\ of\ sequences\ in\ the\ database}$$

*Table 2.* 2 × 2 Contingency table for SWISS-PROT. The axes of the 2 × 2 matrix are labelled by the sets NPP sequences, Random sequences, $H$ (Hypothesis predictions) and $\overline{H}$ (complement of H). The total of the counts/frequencies in the four cells = $S$, where $S$ is the total number of sequences in the SWISS-PROT database.

| | NPP sequences in SWISS-PROT | Random sequences in SWISS-PROT |
|---|---|---|
| $H$ | $\left(\frac{n_1}{n_1+n_3}\right) M$ | $\left(\frac{n_2}{n_2+n_4}\right) (S-M)$ |
| $\overline{H}$ | $\left(\frac{n_3}{n_1+n_3}\right) M$ | $\left(\frac{n_4}{n_2+n_4}\right) (S-M)$ |

$$= M/S \quad (2)$$

$$Pr(NPP \mid Rec) = \frac{N_{db\_NPP\_recog}}{N_{db\_seq\_pred\_pos}} \quad (3)$$

where $N_{db\_NPP\_recog}$ is the number of NPPs in db which are recognised by model and $N_{db\_seq\_pred\_pos}$ is the number of sequences in db which the model predicts to be NPP.

Table 2.1 shows the expected result of using the learned recognition model on the entire SWISS-PROT database. From Equation 3 and Table 2.1 it follows that:

$$\begin{aligned} Pr(NPP \mid Rec) &\simeq \frac{\left(\frac{n_1}{n_1+n_3}\right) \times M}{\left(\frac{n_1}{n_1+n_3}\right) M + \left(\frac{n_2}{n_2+n_4}\right) (S-M)} \\ &= (Mp_1)/(Mp_1 + (S-M)p_2) \quad (4) \end{aligned}$$

where $p_1 = n_1/(n_1+n_3)$ and $p_2 = n_2/(n_2+n_4)$. Substituting Equations 2 and 4 into Equation 1 gives

$$\begin{aligned} RA &= \frac{(Mp_1)/(Mp_1 + (S-M)p_2)}{M/S} \\ &= \frac{Sp_1}{Sp_2 + M(p_1-p_2)} \quad (5) \end{aligned}$$

### 2.2 Estimating Relative Advantage

In the following Relative Advantage over the entire population is represented by $RA$ in capital letters where as Relative Advantage over a sample is denoted by lower case i.e. $ra$. As the value of $M$ is not known, we estimate $\sum_{M=57}^{90} RA$. Therefore we integrate Equation 5 with respect to M. The lower limit of $M$ is equal to the number of known NPPs in SWISS-PROT. The upper limit of $M$ is the most probable maximum number of NPPs in SWISS-PROT i.e. a total of the known NPPs and those proteins which have yet to be scientifically recognised as a NPP.

$$\begin{aligned} \sum_{M=57}^{90} RA &\simeq Sp_1 \times \int_{M=57}^{90} \frac{1}{(p_1-p_2)M + Sp_2} \partial M \\ &\quad + RA(57) \\ &= \frac{Sp_1}{(p_1-p_2)} \ln \frac{90(p_1-p_2) + Sp_2}{57(p_1-p_2) + Sp_2} \quad (6) \end{aligned}$$

We estimate $\sum_{M=57}^{90} RA$ by summing an estimate of the $\sum_{M=57}^{90} RA$ for each instance in the test set as follows, where $n$ is the number of instances in the test set. This method has the advantage that it allows the significance of the difference between the RA of two models to be gauged (see Section 2.3).

$$\sum_{k=1}^{n} \sum_{M=57}^{90} ra_k \quad (7)$$

From Equation 7 and the contingency table it follows that:

$$\sum_{M=57}^{90} ra = \frac{1}{n} \sum_{i=1}^{4} \left( n_i \sum_{M=57}^{90} ra_i \right) \quad (8)$$

Each $\sum_{M=57}^{90} ra_i$ is estimated by substituting $p_1 = \frac{a}{a+c}$ and $p_2 = \frac{b}{b+d}$ into Equation 6. The values of $a$, $b$, $c$ and $d$ are determined by three steps.

1. Whatever the $i$ value, $a$, $b$, $c$ and $d$ are initially given the values of the corresponding counts/frequencies in the contingency table for the test set (see Table 1).

2. Each one of $a$, $b$, $c$ and $d$, is decremented providing that the value before subtraction is greater than 1.

   We do not decrement when the value before subtraction is zero because this can result in $p_1$ or $p_2$ having negative values; this does not make sense because $p_1$ and $p_2$ are probabilities. We do not decrement when the value is one because this can cause $p_1$ or $p_2$ to have the value zero, which in turn has a *highly disproportionate* effect on the value of $\sum_{M=57}^{90} ra_i$.

3. The value of either $a$, $b$, $c$ or $d$ is incremented to reflect the classification of an instance in the cell $n_i$.

For instance, if $i = 2$ and all the counts in the contingency table are greater than one then $a = n_1 - 1, b = n_2, c = n_3 - 1, d = n_4 - 1$.

Note that Steps 1 and 2 assign the same prior probability to each instance because the effect of each

step is not dependent upon which cell the current instance belongs to. Therefore this method of estimating $\sum_{M=57}^{90} RA$ has the properties of a) producing identically distributed random variables representing the outcome for each instance; b) having a sample mean which approaches the population mean in the limit and c) having a relatively small sample variance.

The final step of our method for estimating $RA$ is to take the mean of the summed values.

$$Mean\ RA = \frac{\sum_{M=57}^{90} ra_i}{90-(57-1)} = \frac{\sum_{M=57}^{90} ra_i}{34} \quad (9)$$

### 2.3 Assessing the Significance of the Difference Between the RA of Two Models

We compare the performance of two recognition models, $H_1$ and $H_2$, by comparing their $\sum_{M=57}^{90} RA$ values. Let $d$ be difference in $\sum_{M=57}^{90} RA$ values over the entire population, i.e. for all the proteins in SWISS-PROT, and $\hat{d}$ be the observed difference on the test set.

$$d = \sum_{M=57}^{90} RA_{H_1} - \sum_{M=57}^{90} RA_{H_2} \quad (10)$$

$$\hat{d} = \sum_{M=57}^{90} ra_{H_1} - \sum_{M=57}^{90} ra_{H_2} \quad (11)$$

$\hat{d}$ is an unbiased estimator for the true difference because it is calculated using an independent test set. To determine whether the observed difference is statistically significant we address the following question. What is the probability that $\sum_{M=57}^{90} RA_{H_1} > \sum_{M=57}^{90} RA_{H_2}$, given the observed difference, $\hat{d}$.

If $D$ is a random variable representing the outcome of estimating d by random sampling then, according to the Central Limit Theorem, $\hat{\mu}_D$ is normally distributed in the limit. It has an estimated mean $\hat{d}$ and has an estimated variance of $\hat{\sigma}_D^2/n$. The variance of a random variable, X, is $\sigma_X^2 = E((X)^2) - (E(X))^2$. Therefore, since D is a random variable:

$$\hat{\sigma}_D^2 = \hat{\mu}_{D^2} - \hat{\mu}_D^2 \quad (12)$$

We calculate $\hat{\mu}_{D^2}$ as follows. Let testing the model on test data yield the $4 \times 4$ contingency table shown in Table 3 with the cells $n_{i,j}$. (Note that only those cells shown in bold font can have a count greater than zero because an instance cannot be both an NPP and a Random.)

*Table 3.* $4 \times 4$ Contingency Table. The rows of the $4 \times 4$ matrix are labelled by the cells of the $2 \times 2$ contingency table for $H_1$. The columns of the $4 \times 4$ matrix are labelled by the cells of the $2 \times 2$ contingency table for $H_2$. The cells of the $4 \times 4$ matrix represent the cardinalities of the corresponding intersections of these sets. $\sum_{i=1}^{4} \sum_{j=1}^{4} n_{i,j} = n$, where $n$ is the number of instances in the test set.

| | $n_1$ | $n_2$ | $n_3$ | $n_4$ |
|---|---|---|---|---|
| $n_1$ | $\mathbf{n_{1,1}}$ | $n_{1,2}$ | $\mathbf{n_{1,3}}$ | $n_{1,4}$ |
| $n_2$ | $n_{2,1}$ | $\mathbf{n_{2,2}}$ | $n_{2,3}$ | $\mathbf{n_{2,4}}$ |
| $n_3$ | $\mathbf{n_{3,1}}$ | $n_{3,2}$ | $\mathbf{n_{3,3}}$ | $n_{3,4}$ |
| $n_4$ | $n_{4,1}$ | $\mathbf{n_{4,2}}$ | $n_{4,3}$ | $\mathbf{n_{4,4}}$ |

$$\hat{\mu}_{D^2} = \frac{1}{n} \sum_{i=1}^{4} \sum_{j=1}^{4} \left( n_{i,j} \left( \sum_{M=57}^{90} ra_i - \sum_{M=57}^{90} ra_j \right)^2 \right) \quad (13)$$

Given that $p(\sum_{M=57}^{90} RA_{H_1} > \sum_{M=57}^{90} RA_{H_2}) = p(\sum_{M=57}^{90} RA_{H_1} - \sum_{M=57}^{90} RA_{H_2} > 0)$ we evaluate our null hypothesis by estimating $p(d < 0)$ using the Central Limit Theorem.

$$\int_{x=-\infty}^{0} Pr(d = x)dx = \int_{x=-\infty}^{0} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} dx \quad (14)$$

where $\mu = \hat{\mu}_D$ and $\sigma = \hat{\sigma}_D/\sqrt{n}$.

## 3. Sequence Data in Biology

Research in the biological and medical sciences is being transformed by the volume of data coming from projects which will reveal the entire genetic code (genome sequence) of Homo sapiens as well as other organisms that help us understand the genetic basis of human disease. A significant challenge in the analysis and interpretation of genetic sequence data is the accurate recognition of patterns that are diagnostic for known structural or functional features within the protein. Although regular expressions can describe many of these features they have some inherent limitations as a representation of biological sequence patterns. In recent years attention has shifted towards both the use of neural network approaches (see (Baldi & Brunak, 1998)) and to probabilistic models, in particular hidden Markov models (see (Durbin et al., 1998)). Unfortunately, due to the complexity of the biological signals, considerable expertise is often required to 1) select the optimal neural network architecture or hidden Markov model prior to training and 2) understand the biological relevance of detailed features of the model.

A general linguistic approach to representing the structure and function of genes and proteins has intrinsic appeal as an alternative approach to probabilistic methods because of the declarative and hierarchical nature of grammars. While linguistic methods have provided some interesting results in the recognition of complex biological signals (Searls, 1997) general methods for learning new grammars from example sentences are much less developed.

We considered it valuable to investigate the application of Inductive Logic Programming methods to the discovery of a language that would describe a particularly interesting class of sequences – neuropeptide precursor proteins (NPP). Unlike enzymes and other structural proteins, NPPs tend to show a lower overall sequence similarity despite some evidence of common ancestry within certain groups. This confounds pattern discovery methods that rely on multiple sequence alignment and recognition of biological conservation. NPPs are highly variable in length and undergo specific enzymatic degradation (proteolysis) before the biologically active short peptides (neuropeptides) are released. As a consequence NPPs pose a particular challenge in sequence pattern discovery and recognition. We addressed this challenge by devising the context-free definite clause grammar shown in Fig. 1. We represent protein sequences using the alphabet {`A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y`}, where each letter represents a particular amino acid residue. The start and end represent cleavage sites and the middle-section represents the mature neuropeptide i.e. what remains after cleavage has taken place. A HMM approach is not suitable for NPP sequences because their length is highly variable, they have low overall sequence similarity and they undergo specific enzymatic degradation.

The next section describes an experiment which tries to refute the null hypothesis (see Section 1). It describes the materials used in the experiment and the three steps of the experimental method and presents the results.

# 4. Experiment

## 4.1 Materials

Data was taken from the annotated protein sequence database SWISS-PROT. Our data set[1] comprises a subset of positives i.e. known NPPs and a subset of randomly-selected sequences. It is not possible to identify a set of negative examples of NPPs with certainty because there will be proteins which have yet to be recognised scientifically as a NPP. The subset of positives contains all of the 44 known NPP sequences that were in SWISS-PROT at the time the data set was prepared. 10 of the 44 precursors were reserved for the test set. These sequences are unrelated by sequence similarity to the remaining 34. The subset of randoms contains all of the 3910 full length human sequences in SWISS-PROT at the time the data set was prepared. 1000 of the 3910 randoms were reserved for the test set.

[1]The data set is available at `ftp://ftp.cs.york.ac.uk/pub/aig/Datasets/neuropeps/`.

## 4.2 Method

The method may be summarised as follows:

1. A grammar is generated for NPP sequences using CProgol (Muggleton, 1995) version 4.4 (see Section 4.2.1).
2. A group of features is derived from this grammar. Other groups of features are derived using other learning strategies. (See Section 4.2.2).
3. Amalgams of these groups are formed. A rule set is generated for each amalgam using C4.5 (Release 8) (Quinlan, 1993) and C4.5rules[2] and its performance is measured using $MeanRA$. The null-hypothesis (see Section 1) is then tested by comparing the $MeanRA$ achieved from the various amalgams. (See Section 4.2.3).

During both the generation of the grammar using CProgol and the generation of propositional rule sets using C4.5 and C4.5rules we adopt the background information used in Muggleton et al. (1992) to describe physical and chemical properties of the amino acids.

Table 4 summarises how some of the properties SWISS-PROT changed over the duration of the experiments described in this paper and the subsequent preparation of this paper. All the $MeanRA$ measurements in this paper are based on the properties as they stood in May, 1999; these were the most up-to-date values available at the time the measurements were made.[3]

### 4.2.1 Grammar Generation

A *NPP grammar* contains rules that describe legal neuropeptide precursors. Fig. 1 shows an example of such a grammar, written as a Prolog program. This section

[2]The default settings of C4.5 and C4.5rules were used.

[3]When measuring performance using $MeanRA$ there is no requirement that the size of the test data set is equal to the number of known human NPPs in SWISS-PROT.

*Table 4.* Properties of sequences in SWISS-PROT at the time the data set described in Section 4.1 was prepared and in May, 1999.

| | Prep time[a] | May '99 |
|---|---|---|
| Number of sequences | 64,000 | 79,449 |
| Number of known human NPPs | 44 | 57 |
| Most probable maximum number of human NPPs | Not known | 90 |

[a] At the time the data set was prepared

```
npp(A,B):- signal(A,C),
           star(C,D),
           neuro_peptide(D,E),
           star(E,B).
signal(A,C):- ...
neuro_peptide(D,E):- start(D,F),
                     middle(F,G),
                     end(G,E).
start(D,F):- ...
middle(F,G):- ...
end(G,E):- ...
```

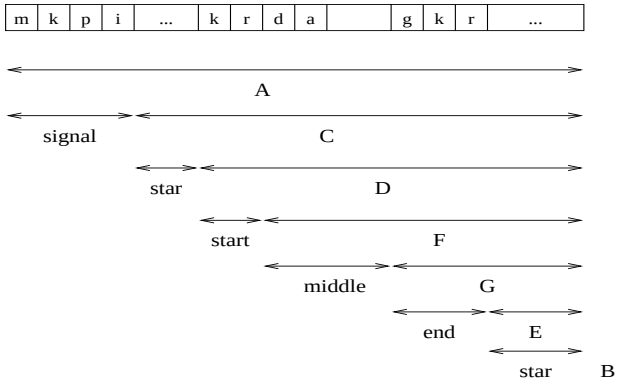


*Figure 1.* Grammar rules describing legal NPP sequences. The rules comply with Prolog syntax. $npp(X, Y)$ is true if there is a precursor at the beginning of the sequence $X$, and it is followed by a sequence $Y$. The other dyadic predicates are defined similarly. $star(X, Y)$ is true if, at the beginning of the sequence $X$, there is some sequence of residues whose length is not specified and which is followed by another sequence $Y$. Definitions of the predicates denoted by '...' are to be learnt from data of known NPP sequences.

describes how production rules for signal peptides and neuropeptide starts, middle-sections and ends were generated using CProgol. These were used to complete the context-free definite clause grammar structure shown in Fig. 1.

The grammar to be learnt by CProgol contains dyadic non-terminals of the form `p(X,Y)`, which denote that property `p` began the sequence `X` and is followed by a sequence `Y`. To learn production rules for these non-terminals from the training set, CProgol was provided with:

1. extensional definitions of these non-terminals.
2. definitions of the non-terminals `star/2` and `run/3`. `star/2` represents some sequence of unnamed residues whose length is not specified. `run/3` represents a run of residues which share a specified property.
3. production rules for various domain-specific subsequences and patterns. This natural inclusion of existing biochemical knowledge illustrates how the grammar-based approach presents a powerful method for describing NPPs.

Certain restrictions were placed on the length of NPPs, signal peptides and neuropeptides because pilot experiments had shown that they increased the accuracy of the grammar. These constraints only affect the values of features derived from the grammar. They do not constrain the value of the sequence length feature described at the end of Section 4.2.2.

### 4.2.2 Feature Groups

**1) The grammar features** Each feature in this group is a prediction about a NPP sequence made by parsing the sequence using the grammar generated by CProgol. **2) The SIGNALP features** Each feature in this group is a summary of the result of using SIGNALP on a sequence. SIGNALP (Nielsen et al., 1997) represents the pre-eminent automated method for predicting the presence and location of signal peptides. **3) The proportions features** Each feature in this group is a proportion of the number of residues in a given sequence which either are a specific amino-acid or which have a specific physicochemical property of an amino-acid. **4) The sequence length feature** This feature is the number of residues in the sequence.

### 4.2.3 Propositional Learning

The training and test data sets for C4.5 were prepared as follows.

1. Recall from Section 4.1 that our data comprises 44 positives and 3910 randoms. 40 of the 44 positives occur in the set of 3910 randoms. As C4.5 is designed to learn from a set of positives and a set of negatives, these 40 positives were removed from the set of randoms. Of the 40 positives which are in the set of randoms, 10 are in the test set. Hence the set of $(3910 - 40)$ sequences were split into a training set of $(2910 - 30 = 2880)$ and a test set of $(1000 - 10 = 990)$.

2. Values of the features were generated for each training and test sequence. Each sequence was represented by a data vector comprised of these feature values and a class value ('1' to denote a NPP and '0' otherwise).

3. Finally to ensure that there were as many '1' sequences as '0' sequences a *training* set of 2880 NPPs was obtained by sampling with replacement. Thus the training data set input to C4.5 comprised $(2 \times 2880)$ examples. (No re-adjusting was done on the test data.)

Amalgams of the feature groups described in the previous section were formed. The amalgams are listed in Table 5. The following procedure was followed for each one: (1) training and test sets were prepared as described above; (2) a decision tree was generated from the training set using C4.5; (3) a rule set was generated from this tree using C4.5rules; (4) a $2 \times 2$ contingency table was drawn-up based on the predictions of this rule set on the test set; (5) $MeanRA$ was estimated from this contingency table.

The refutation of the null hypothesis was then attempted as described in Section 1.

### 4.3 Results and Analysis

Table 5 shows the $MeanRA$ and predictive accuracy for each amalgam of feature groups. The highest $MeanRA$ (107.7) was achieved by one of the grammar amalgams, namely the 'Proportions + Length + SignalP + Grammar' amalgam. The best $MeanRA$ achieved by any of the amalgams which do not include the grammar-derived features was the 49.0 attained by the 'Proportions + Length' amalgam. This difference is statistically significant: $p(d < 0)$ is well below 0.0001.

Table 5 shows that predictive accuracy is not a good measure of performance for this domain because it does not discriminate well between the amalgams: despite covering varying numbers of (the rare) positives, all the models are awarded a similar (high) score by predictive accuracy because they all exclude most of the abundant negatives.

*Table 5.* Estimates of $MeanRA$ and predictive accuracy of the amalgams of the feature groups.

| Amalgam | $Mean$ $RA$ | Predictive Accuracy (%) |
|---|---|---|
| Only props | 0 | $96.7 \pm 0.6$ |
| Only Length | 1.6 | $91.8 \pm 0.9$ |
| Only SignalP | 11.7 | $98.1 \pm 0.4$ |
| Only Grammar | 10.8 | $97.0 \pm 0.5$ |
| Props + Length | 49.0 | $98.6 \pm 0.4$ |
| Props + SignalP | 15.0 | $98.3 \pm 0.4$ |
| Props + Grammar | 31.7 | $98.2 \pm 0.4$ |
| SignalP + Grammar | 0 | $98.6 \pm 0.4$ |
| Length + Grammar | 0 | $96.2 \pm 0.6$ |
| Length + SignalP | 34.4 | $98.7 \pm 0.4$ |
| Length + SignalP + Grammar | 0 | $98.0 \pm 0.4$ |
| Props + Length + SignalP | 29.2 | $98.7 \pm 0.4$ |
| Props + Length + Grammar | 33.2 | $98.5 \pm 0.4$ |
| Props + SignalP + Grammar | 15.0 | $98.3 \pm 0.4$ |
| Props + Length + SignalP + Grammar | 107.7 | $99.0 \pm 0.3$ |

## 5. Discussion

This paper has shown that the most accurate comprehensible multi-strategy predictors of biological sequence families employ Chomsky-like grammar representations.

The positive-only learning framework of the Inductive Logic Programming (ILP) system CProgol was used to generate a grammar for recognising a class of proteins known as human neuropeptide precursors (NPPs). As far as these authors are aware, this is both the first biological grammar learnt using ILP and the first real-world scientific application of the positive-only learning framework of CProgol.

Figure 2 illustrates the advantage of using our best recognition model to search for a novel NPP. If one searches for a NPP by randomly selecting sequences from SWISS-PROT for synthesis and subsequent biological testing then, at most, only one in every 2408 sequences tested is expected to be a novel NPP. Using our best recognition model as a filter makes the search for a NPP far more efficient. Approximately one in every 22 of the randomly selected SWISS-PROT se-

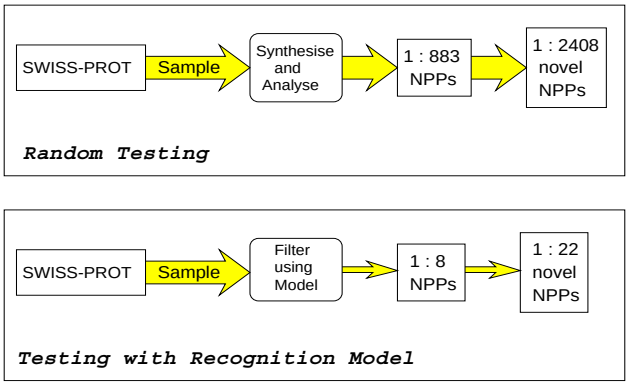


*Figure 2.* Finding novel NPPs in SWISS-PROT. Comparison of random testing and testing with our best recognition model.

quences which pass through our filter is expected to be a novel NPP.

The best 'non-grammar' recognition model does not provide any biological insight. However the best recognition model which includes grammar-derived features is broadly comprehensible and contains some intriguing associations that may warrant further analysis. This model is being evaluated as an extension to existing methods used in SmithKline Beecham for the selection of potential neuropeptides for use in experiments to help elucidate the biological functions of G-protein coupled receptors.

The new cost function presented in this paper, Relative Advantage ($RA$), may be used to measure performance of a recognition model for any domain where:

1. the proportion of positives in the set of examples is very small.
2. there is no guarantee that all positives can be identified as such. In such domains, the proportion of positive examples in the population is not known and a set of negatives cannot identified with complete confidence.
3. there is no benchmark recognition method.

We have developed a general method for assessing the significance of the difference between $RA$ values obtained in comparative trials. $RA$ is estimated by summing the estimate of performance on each test set instance. The method uses identically distributed random variables representing the outcome for each instance; a sample mean which approaches the population mean in the limit and a relatively small sample variance.

## Acknowledgements

This research was conducted as part of a project entitled 'Using Machine Learning to Discover Diagnostic Sequence Motifs' supported by a grant from SmithKline Beecham. SmithKline Beecham identified the problem addressed by the case-study, prepared the data set, and provided expertise on NPPs and general Bioinformatics methods. During part of his time spent writing this article C.H.B. was supported by the EPSRC grant (GR/M56067) entitled 'Closed Loop Machine Learning'. A.S. holds a Nuffield Trust Research Fellowship at Green College, Oxford. The authors would like to thank Marcel Turcotte for his advice on HMM software.

## References

Baldi, P., & Brunak, S. (1998). *Bioinformatics: the machine learning approach*. Cambridge, MA: MIT Press.

Durbin, R., Eddy, S., Krogh, A., & Mitchison, G. (1998). *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge: Cambridge University Press.

Emmert, D., Stoehr, P., Stoesser, G., & Camerson, G. (1994). The European Bioinformatics Institute (EBI) databases. *Nucleic Acids Research*, *22*, 3445–3449.

Michalski, R., & Wnek, J. (1997). Guest editors' introduction. *Machine Learning*, *27*, 205–208.

Muggleton, S. (1995). Inverse entailment and Progol. *New Generation Computing*, *13*, 245–286.

Muggleton, S., King, R., & Sternberg, M. (1992). Protein secondary structure prediction using logic-based machine learning. *Protein Engineering*, *5*, 647–657.

Nielsen, H., Engelbrecht, J., Brunak, S., & vonHeijne, G. (1997). Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. *Protein Engineering*, *10*, 1–6.

Quinlan, J. (1993). *C4.5: Programs for machine learning*. San Mateo, CA: Morgan Kaufmann.

Searls, D. (1997). Linguistic approaches to biological sequences [review]. *Computer Applications in the Biosciences*, *13*, 333–344.