Chapter 2

Enhanced Domain Detection Using Approaches From Speech Recognition

Most modern speech recognition techniques use probabilistic models to interpret a sequence of sounds [Cha93, Jel97]. Hidden Markov models, in particular, are used to recognize words. The same techniques have been adapted to find domains in protein sequences of amino acids [KBM$^+$94, DEKM98], as discussed in section 1.2. However in both cases, detection of individual constituent domains or words is impeded by noise. One technique which has been successfully used in speech recognition is to use language models to capture the information that certain word combinations are more likely than others, thus improving detection based on context. As discussed in section 1.1, only a limited set of all possible domain combinations are observed, and the pattern of occurrence is highly non-random ([AGT01b, AHT03]). Moreover, particular domain combinations are re-used in many domain architectures [VBB$^+$04]. Thus, language models from speech recognition may also be applicable to the problem of protein domain identification. I have successfully used this approach to improve domain prediction in Pfam [CBD03].

Furthermore, different species have different protein domain repertoires, even to the extent that certain protein domain families are kingdom specific. More strikingly, domain combinations are highly kingdom specific ([AGT01b, VBB$^+$04]). Thus, taxonomic context by itself may also provide extra information for domain detection, and is likely to be even more useful when used in combination with language models of domain context. I have previously used taxonomic information to improve domain identification in Pfam [CBD04].

In this chapter, I will present a unified model of domain and taxonomic context, extending the approaches of [CBD03, CBD04]. I will first provide a brief overview of some of the techniques used in speech recognition, followed by a comparison of the high-level statistics of word and domain use which will help to motivate further the application of language modelling to domain detection. I then modify the speech recognition techniques in order to apply them to domain detection and to incorporate taxonomic context. The results section comprises firstly a test of the method on proteins of known structure using the SCOP classification[AHB$^+$04], in which I will show that the combined taxonomic and domain context method performs better than the individual methods and that each perform better than a standard search which ignores context altogether. The final part of the results section consists of a scan of the combined method against all Uniprot[ABW$^+$04] proteins to determine the number of novel Pfam domain occurrences detectable with this technique.

2.1 Statistical Speech Recognition Techniques

Speech recognition has been greatly facilitated by the application of statistical models including hidden Markov models (HMMs) and Bayesian methods. The steps in the process are illustrated in figure 2.1.

Once the acoustic signal has been parsed into discrete sound symbols, the statistical approach is to build two types of model: for each word there is a phonetic model for the emission of sounds, based on observed pronunciation patterns in terms of phonemes; above this there is a language model for the emission of a sequence of words, based on word use patterns. In order to recognize a given sentence, the method seeks the sequence of words $\mathbf{D} = \mathbf{D}_1, \ldots \mathbf{D}_n$ that maximises the probability of the sentence given the acoustic evidence x and the language model $\mathbf{M}$. This probability can be split (using Bayes' rule) into a word term based on the phonetic model (first term), and a 'context' term, based on the language model (second term):

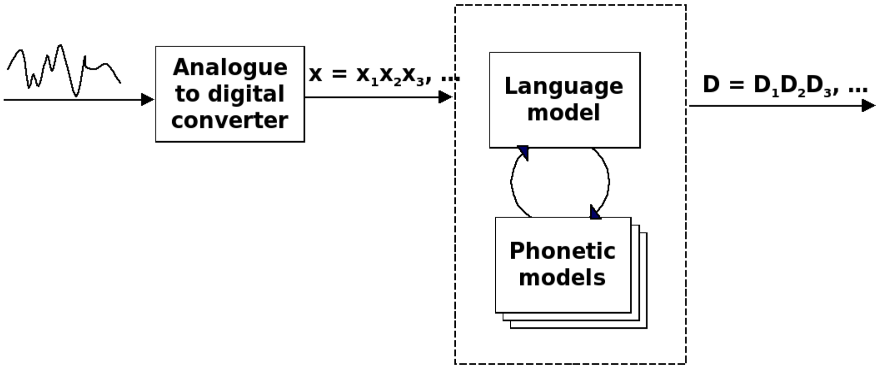


Figure 2.1: Schema for a speech recognizer. First the analogue speech waveform is converted into a sequence of phonemes, $x = x_1, x_2, \ldots$. This sequence is processed by a composite stochastic language and phoneme model.

$$P(\mathbf{D}|x, \mathbf{M}) = \frac{P(x|\mathbf{D})}{P(x|\mathbf{M})} P(\mathbf{D}|\mathbf{M}),, \tag{2.1}$$

assuming that x is conditionally independent of the language model **M** given **D**. When searching for the most likely sequence of words **D**, $P(x|\mathbf{M})$ is a fixed constant so it suffices to maximise

$$P(\mathbf{D}|x, \mathbf{M}) \propto P(x|\mathbf{D})P(\mathbf{D}|\mathbf{M}). \tag{2.2}$$

Referring again to figure 2.1 and equation 2.1 observe that statistical speech recognition naturally divides itself into the following sub-problems, each of which I discuss to the extent it applies to domain recognition. See [RJ93] for further details.

Conversion Convert the analogue signal into a discrete acoustic signal

Acoustic Modelling For each word, develop and parameterise an acoustic model capable of discriminating the given word from all others.

Language Modelling Develop and parameterise a single language model

Conversion to a digital signal

The basic idea is to sample the properties of the acoustic signal at some rate (e.g 100HZ), and to find the closest match to this vector of properties from a library of reference vectors. As biological sequences are already digitized, this problem is not applicable.

Acoustic Modelling

The aim is to construct an acoustic model for each word in the language model which is capable of recognizing words from acoustic signal. A phonetic encoding is determined for each word in the vocabulary as a sequence of phonemes $\phi_1, \phi_2, ...$ from a phonetic dictionary. For each phoneme in the phonetic dictionary a HMM is created which emits over the space of sound symbols obtained from the previous step. The encoded phonemes' HMMs are concatenated to form a word HMM. Training data for the word models is obtained by recording word pronunciations. The model can be trained from this data using the Baum-Welch algorithm[DEKM98]. This step corresponds to using a profile HMM in biological sequence modelling, as discussed in section 1.2.

Language Modelling

The aim of language modelling is to create a model over all possible word combinations which reflect actual word use patterns in speech. The analogy in domain recognition is a model over all possible domain combinations which reflect protein domain occurrence patterns. Mathematically this corresponds to parameterising the distribution $P(\mathbf{D}|\mathbf{M}) = P(\mathbf{D}_1, \ldots \mathbf{D}_n|\mathbf{M})$ in a tractable form. One approach is to assume that word use is a Markov process. That is, if the joint probability is expressed in terms of conditional probabilities,

$$P(\mathbf{D}_1 \ldots \mathbf{D}_n|\mathbf{M}) = P(\mathbf{D}_1)P(\mathbf{D}_2|\mathbf{D}_1) \ldots P(\mathbf{D}_n|\mathbf{D}_1, \ldots, \mathbf{D}_{n-1}),$$

to assume that

$$P(\mathbf{D}_i|\mathbf{D}_{i-1}, \ldots, \mathbf{D}_1) = P(\mathbf{D}_i|\mathbf{D}_{i-1}, \ldots, \mathbf{D}_{i-k}).$$

In speech recognition, a second order ($k = 2$) Markov model is usually found to be most effective, which is called a *trigram* model. First order methods are called *digram* methods. In principle, the higher the order k, the more long-range dependencies can be incorporated into the model. However, for a fixed data set, as k increases less and less training data becomes available for the particular context and so the probability estimates become less and less reliable. In linguistic terms, Markov models are stochastic regular grammars, and therefore do not capture the grammatical structure of a sentence. Thus they are not capable of assigning zero probability to grammatically incorrect sentences, nor modelling long range dependencies implied by the grammatical structure. To achieve this it is necessary to use (in order of increasing complexity and ability to effectively model linguistic structures) stochastic context free grammars, tree-adjoining grammars [JY99] or context sensitive grammars [Cho59]. However, Markov models are computationally efficient and have been found to work surprisingly well in practice.

For domain recognition, it is not yet clear that there is a general higher-order grammar for domain occurrence, much less how to represent the syntax with a formal grammar. Thus, approximating the dependence of domain occurrence based on adjacent domains appears to be an appropriate way to proceed. A phenomenon which occurs in protein domain combinations but not speech is nested domains, which account for 9% of all domain combinations [ASHS04].

Training data for a language model is obtained from analysing text, typically in the subject area in which the model will be used. In one sense training a language model is straightforward, as there are no hidden variables and the transition probabilities between words can be observed directly. However, the main challenge with language modelling is data sparseness, particularly with trigram methods. The training corpus will not contain all possible trigram word combinations used in speech, and observed trigrams occur at such low frequencies that observed counts are not reliable estimators of probability. This is dealt with via smoothing, which is an integral part of language modelling and has formed the basis for much language modelling research.

Equivalence classification of words is one technique for smoothing sparse data. An example is to treat all the synonyms for a particular word as the same; another is to classify all proper names as a single word. An example from domain modelling is classifying all members of a superfamily as the same domain, or classifying regions of low complexity as a single domain. The method developed in this chapter classifies all Pfam families in the same Pfam clan as the same family.

Another smoothing technique is to interpolate lower order counts in the estimation of the trigram and digram probabilities. That is, to assign

$$P(\mathbf{D}_i|\mathbf{D}_{i-1}) = \alpha_1 P(\mathbf{D}_i) + (1-\alpha_1)\frac{\mathbf{N}(\mathbf{D}_{i-1},\mathbf{D}_i)}{\mathbf{N}(\mathbf{D}_{i-1})}, \tag{2.3}$$

$$P(\mathbf{D}_i|\mathbf{D}_{i-1},\mathbf{D}_{i-2}) = \alpha_2 P(\mathbf{D}_i|\mathbf{D}_{i-1}) + (1-\alpha_2)\frac{\mathbf{N}(\mathbf{D}_{i-2},\mathbf{D}_{i-1},\mathbf{D}_i)}{\mathbf{N}(\mathbf{D}_{i-2},\mathbf{D}_{i-1})}, \tag{2.4}$$

where $\mathbf{N}(\mathbf{D}_i)$ is the count of $\mathbf{D}_i$ in the corpus, so that even in the case the trigram is not observed it will have a non-zero probability assigned based on the digram probability. One principle often used, called back-off estimation, is that the trigram probabilities $P(\mathbf{D}_i|\mathbf{D}_{i-1}, \mathbf{D}_{i-2})$ should be more reliable if the context $\mathbf{D}_{i-2}, \mathbf{D}_{i-1}$ is observed many times in the training corpus, and similarly for the digram probabilities. Thus it makes sense that the interpolation parameters α are not constant but rather decreasing functions of the amount of context, e.g. $\alpha_2 = f(\ \mathbf{N}(\mathbf{D}_{i-2}, \mathbf{D}_{i-1}))$. A step-function is typically used to approximate f, with several categories each having a different value of α. A portion of the training data is held over to estimate the optimal values for this function.

An alternative to Markov models for approximating the joint distribution $P(\mathbf{D}_1, \ldots, \mathbf{D}_n)$, which only capture local dependencies, is the whole-sentence exponential model introduced by Rosenfeld and co-workers [RCZ01]. The whole sentence exponential model takes the form

$$P(\mathbf{D}) = \frac{1}{Z} P_0(\mathbf{D}) \exp(\sum_i \lambda_i f_i(\mathbf{D})) \tag{2.5}$$

where Z is the normalizing constant and the $f_i(\mathbf{D})$ are termed *features* of the sentence: arbitrary properties of the sentence which can be computed. $P_0(\mathbf{D})$ is an initial approximation to $P(\mathbf{D})$, which can be a uniform distribution, or the distribution obtained from the trigram model described above. It can be shown that there exists is a unique equation of the form of eq. 2.5 which satisfies the following constraints on the feature averages under $P(\mathbf{D})$,

$$E_P(f_i) = K_i, \tag{2.6}$$

provided the constraints are consistent. Moreover, among all solutions to equation 2.6 (including solutions not of exponential form), the exponential solution is closest to $P_0(\mathbf{D})$ under the Kullback-Leibler distance (see [DEKM98]). This means that in the case $P_0(\mathbf{D})$ is the uniform distribution, the exponential solution is the solution which maximises the entropy. In this sense the exponential solution is appealing because it maximises the uncertainty of the distribution while still satisfying all of the constraints presented. So, given a training corpus, the strategy of whole-sentence exponential modelling is to first choose features f_i which capture particular aspects of the data, then to calculate empirical averages of the f_i over all sentences $\mathbf{D}'$ in the training corpus

$$K_i = \frac{1}{N} \sum_{\mathbf{D}'} f_i(\mathbf{D}'),$$

and finally to find the unique equation of the form eq. 2.5 which satisfies the constraints in eq. 2.6. An iterative procedure is available to find this solution, and is given in [RCZ01]. The main challenge in implementing this procedure is that it requires calculating the average $E_p(f_i)$ over all possible sentences $\mathbf{D}$ at each step in the iteration. This is approximated by Rosenfeld and colleagues using a sampling technique.

In speech recognition the features used in an exponential model include: the number of times a particular n-gram occurs, either sequentially, or in the entire sentence; existence of particular grammatical structures; pauses etc. For domain recognition this framework could incorporate arbitrary co-occurrence patterns (not just adjacent co-occurrence), expected distribution over the number of repeats as well as protein specific information such as taxonomy, function and localisation.

In this work I focus on applying the Markov rather than exponential model approach to language modelling. The Markov model is substantially more efficient to train and to score, and has been used successfully in speech recognition. Moreover, early results from whole-sentence models do not appear to provide a significant improvement in performance [RCZ01]. However, the exponential model does appear to provide significantly more flexibility and is certainly an avenue for further investigation.

2.2 Patterns of domain occurrence and co-occurrence

To motivate the application of language models to protein domain recognition it is interesting to observe the patterns of domain occurrence in relation to the pattern of word occurrence.

Zipf [Zip35] first described the power law behaviour of word occurrence. The Zipf distribution for words is displayed in figure 2.3 and reflects the fact that some words are used very frequently while most words are used rarely. The power law distribution is of the form $\mathbf{N}(D) = aR(D)^{-b}$ where $\mathbf{N}$ is the count of a word and $R(D)$ is the rank of the word according to its count. A Zipf distribution also satisfies $b = 1$. Power law behaviour has been observed in many biological contexts, including the distribution of protein families and folds [QLG01], occurrence of DNA k-mers, occurrence of pseudogenes and levels of gene expression [LQZ+02].

The Zipf curve for words in figure 2.3 applies from rank 3 to rank 2000 but breaks down after this. Figure 2.2 shows a power law distribution for Pfam domains. The slope of this

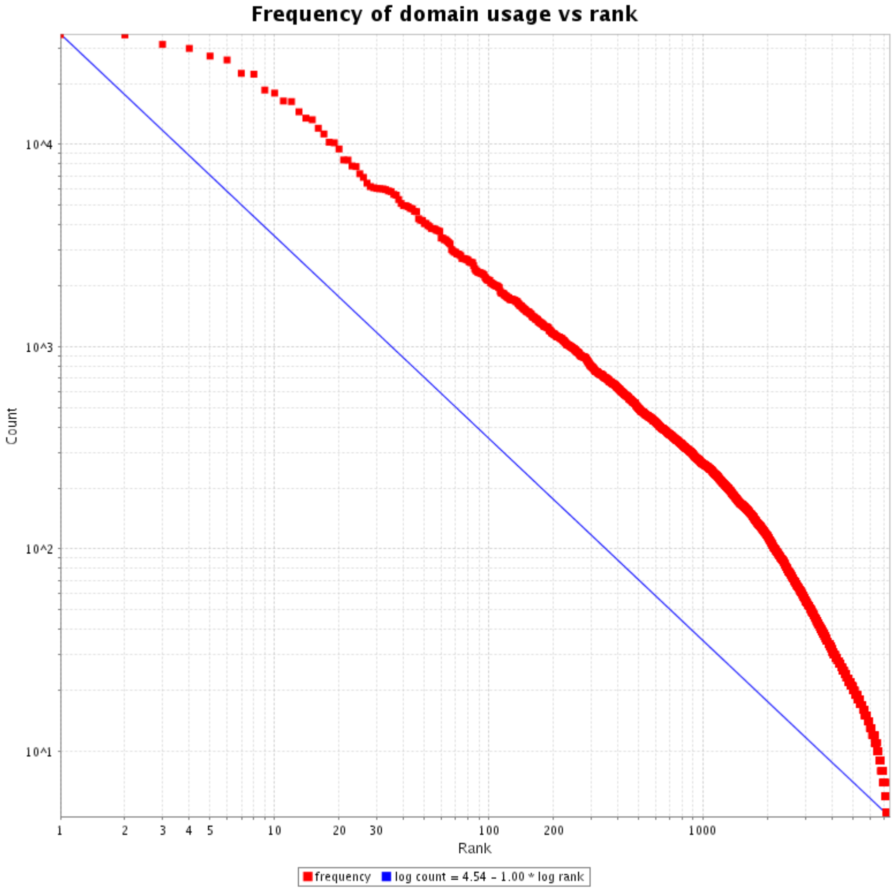


Figure 2.2: Plot of frequency of domain occurrence vs domain rank (according to frequency, decreasing from left to right). The domains models are from Pfam 15.0, and are scored over all proteins from Uniprot. Pfam clans have been used to group closely related domains into a single entry. The blue line shows the $\log y = C - 1.0 \log x$ line interpolated between the highest and lowest ranked domain.

graph is approximately 1.0 from rank 5 to rank 2000, but the gradient is higher at high-rank domains and lower at low-rank domains. As domain annotation improves, we expect to find novel small families, but for some of these small families to have more than 200 instances. These families will then be of higher rank than than those known families of rank 2000 and above. We also expect to increase the number of instances in small families as they are not as well characterised as larger families. The combined effect should be to expand the region of the graph following Zipf's law to toward the right. It appears that Zipf's law fits protein domains at least as well as words.

Next, I consider the different patterns of domain occurrence given different taxonomic

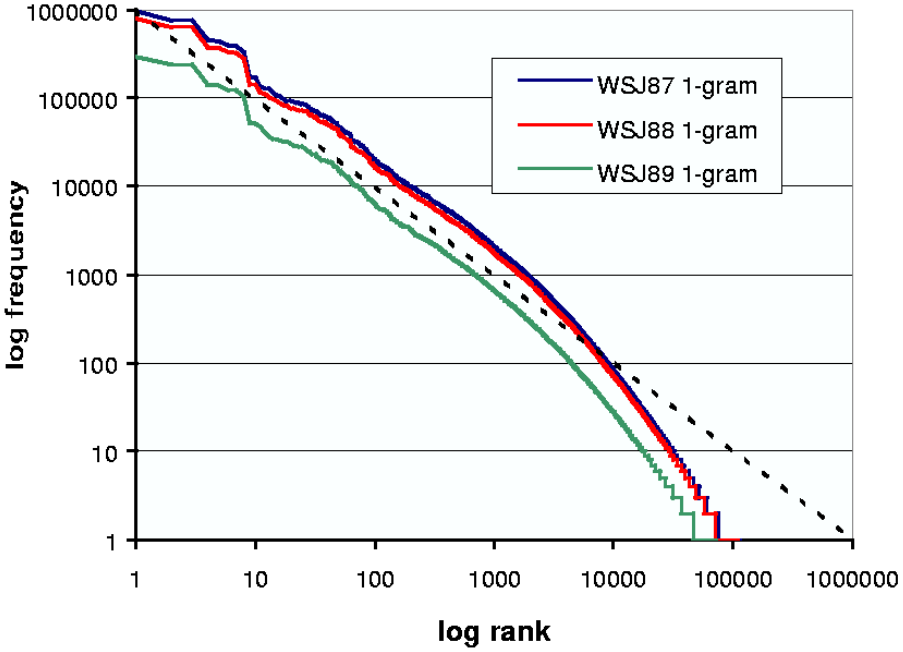


Figure 2.3: Plot of frequency of word occurrence, taken from the Wall Street Journal from 1987, 1988 and 1989 with sizes approximately 19 million, 16 million and 6 million words respectively. This graph is taken from [HSGMS02]

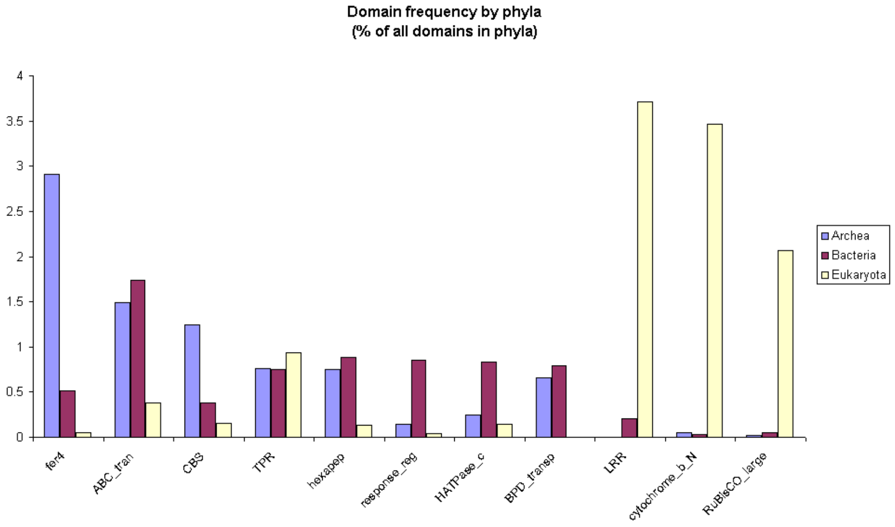


Figure 2.4: Distribution of example domains amongst archaea, eukaryota and bacteria from all proteins in Uniprot. The top 5 domains for each phyla are included. This graph was not constructed on a genome basis and redundancies in the Uniprot database have not been removed, thus the graph may display bias due to over-representation of particular sequences in Uniprot.

contexts. Fig. 2.4 shows examples of domains which have biased taxonomic distribution. For example, the 4Fe-4S binding domain comprises 2.9% of archaeal domains in Pfam, but only 0.5% of bacterial domains and 0.05% of eukaryotic domains. Therefore a weak 4Fe-4S binding domain signal in archaea is more likely to be a real signal than a weak eukaryota 4Fe-4S binding domain signal. Intuitively, less amino-acid based evidence is required to believe an 4Fe-4S binding domain in archaea than in eukaryota.

Figure 2.5 demonstrates different patterns of co-occurrence of the TPR domain across three kingdoms of life. The TPR domain mediates protein-protein interactions and is observed in eukaryota, bacteria and archaea, as can be seen in figure 2.4. In each of the three kingdoms there is a high probability of observing TPR following another TPR repeat. Uniquely in eukaryota, a TPR domain is frequently observed following an APC8 (Anaphase promoting complex sub-unit 8) domain and also following a PRP1_N (PRP splicing factor, N-terminal) domain. Uniquely in bacteria, a TPR domain has high probability following a NB-ARC (signalling motif found in bacteria and eukaryota) and following an FF domain (also involved in protein-protein interaction and found in eukaryotes and bacteria). Uniquely to Archaea, there is a high probability of observing a TPR domain following a CW_binding_2

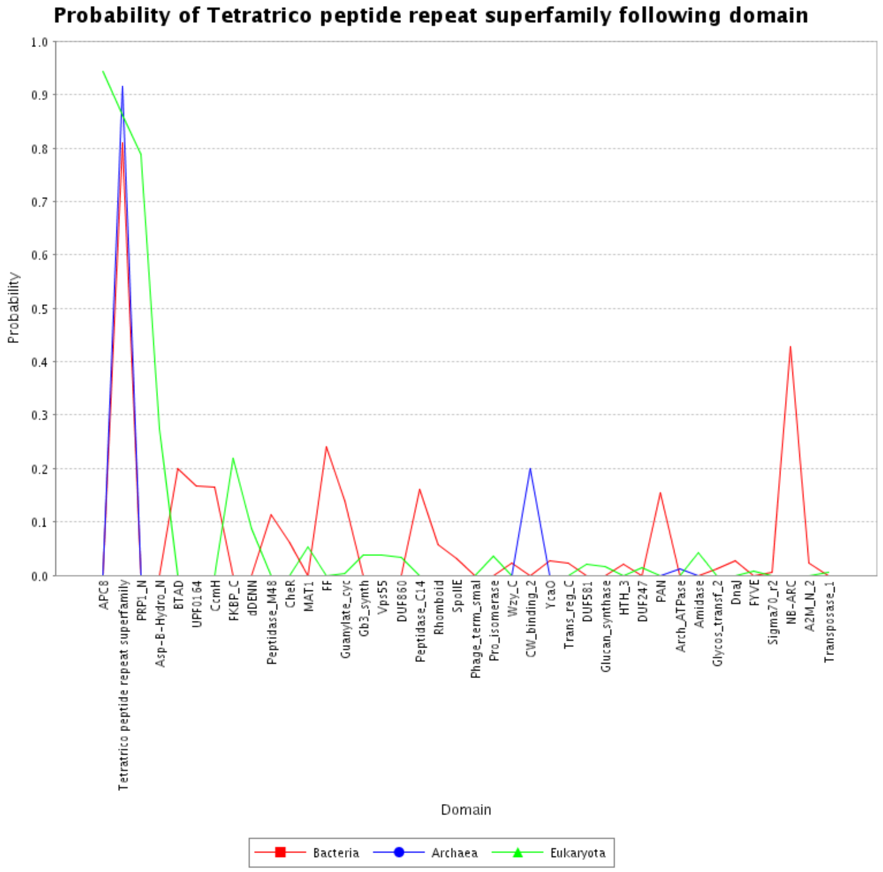


Figure 2.5: Observed probability of Tetratrico Peptide Repeats in different contexts. The probability of observing a member of the TPR clan given the combined taxonomic and domain context - Bacteria (red), Archae (blue) and Eukaryota (green) - and preceding domain.

(putatively involved in cell wall binding) domain. This reinforces the findings of [AGT01b] that domain combinations are highly kingdom specific, and also indicates the importance of building language models which take the taxonomic context into account.

2.3 Methods: Application to protein domain detection

As discussed in section 2.1, profile HMM techniques introduced in section 1.2 broadly map to the acoustic modelling problem in speech recognition [KBM$^+$94, DEKM98]. In this section, I will modify the language modelling techniques outlined in section 2.1 to apply them to protein domain recognition.

2.3.1 Formulation

Let $\mathbf{M}$ denote the combined language and taxonomy model. For each amino acid sequence x with taxonomy T my approach is to annotate the sequence with the domain sentence $\mathbf{D} = \mathbf{D}_1, \mathbf{D}_2, \dots \mathbf{D}_n$ matching amino acid segments $\mathbf{D}_i \leftrightarrow x_{[s_i,e_i]}$ if the probability $P(\mathbf{D}|x, \mathbf{M})$ is sufficiently high. Let $\mathbf{R}$ denote the background model for generating the sequence independently residue by residue according to an average compositional model. Note that in this formulation $\mathbf{D}$ stands for the N- to C-terminal linear sequence of domains as well as the particular set of (start, end) protein co-ordinates for each of the $\mathbf{D}_i$. I require that the $\mathbf{D}_i$ do not overlap, but place no restriction on the size of the gaps between the $\mathbf{D}_i$. It is not required that the $x_{[s_i,e_i]}$ completely cover the protein. It should be noted that in the case where a protein domain has not yet been modelled (for instance, it does not appear in the pdb, and it has not yet been discovered in sequence space), a relatively large gap may result in the correct domain annotation of a sequence. Also, transmembrane and low complexity regions are not modelled. Residues which are not within a $x_{[s_i,e_i]}$ will be assumed to be emitted under the model $\mathbf{D}$ according to the background distribution $\mathbf{R}$. Then

$$P(\mathbf{D}|x,T,\mathbf{M}) = \tfrac{P(x|\mathbf{D},T,\mathbf{M})}{P(x|T,\mathbf{M})} P(\mathbf{D}|T,\mathbf{M}) \tag{2.7}$$

$$\propto \tfrac{P(x|\mathbf{D},T,\mathbf{M})}{P(x|\mathbf{R})} P(\mathbf{D}|T,\mathbf{M}) \tag{2.8}$$

$$= \left(\prod_i \tfrac{P(x_{[s_i,e_i]}|\mathbf{D}_i,T,\mathbf{M})}{P(x_{[s_i,e_i]}|\mathbf{R})} P(\mathbf{D}_i)\right) \times \left(\prod_i \tfrac{P(\mathbf{D}_i|T,\mathbf{M},\mathbf{D}_1,\dots\mathbf{D}_{i-1})}{P(\mathbf{D}_i)}\right), \tag{2.9}$$

assuming independence of the amino acid fragments $x_{[s_i,e_i]}$ from the other fragments $x_{[s_j,e_j]}$, $j \neq i$ conditional on $\mathbf{D}_i, T, \mathbf{M}$. Because I am only interested in maximising $P(\mathbf{D}|x,T,\mathbf{M})$ over all possible domain sentences and fixed x, the term $P(x|T,\mathbf{M})$, which is independent of the domain sentence, has been replaced with $P(x|\mathbf{R})$. Then residues not belonging to any sequence fragment $x_{[s_i,e_i]}$ cancel out between the numerator and denominator.

Taking logs and defining the overall sentence score $\mathbf{SS}_{x,T,M}$

$$\log P(\mathbf{D}|x,T,\mathbf{M}) \propto$$

$$\mathbf{SS}_{x,T,M}(\mathbf{D}) := \left(\sum_i \log \frac{P(x_{[s_i,e_i]}|\mathbf{D}_i)}{P(x_{[s_i,e_i]}|\mathbf{R})} - \tau_{\mathbf{D}_i}\right) + \left(\sum_i \frac{P(\mathbf{D}_i|T,\mathbf{M},\mathbf{D}_1,\dots\mathbf{D}_{i-1})}{P(\mathbf{D}_i)}\right), \tag{2.10}$$

with domain score threshold $\tau_{\mathbf{D}} = \log \frac{1}{P(\mathbf{D})}$. Note that $P(x_{[s_i,e_i]}|\mathbf{D}_i)$ represents the probability that the model for domain $\mathbf{D}_i$ generated the sequence $x_{[s_i,e_i]}$; and that $P(x_{[s_i,e_i]}|\mathbf{R})$ represents the probability that the sequence was generated independently residue by residue according to a background composition model. Also, $P(\mathbf{D})$ represents the probability of obtaining $\mathbf{D}$ according to a background distribution over domains. The left-hand bracket scores the fit of the domain sentence to the amino-acid sequence, while the right-hand bracket is the context dependent score.

A simplified view of the Pfam annotation process [BCD$^+$04] is that a domain $\mathbf{D}$ annotating the sequence fragment $x_{[s_i,e_i]}$ is recognized as real if the domain log-odds ratio is greater than a manually curated threshold,

$$\log \frac{P(x_{[s_i,e_i]}|\mathbf{D}_i)}{P(x_{[s_i,e_i]}|\mathbf{R})} > \tau_{\mathbf{D}_i}. \tag{2.11}$$

This log-odds ratio is calculated using the HMMER package [Edd98]. The actual process is somewhat more complicated. As outlined in section 1.2, HMMER calculates the log-odds ratio that the model generated the full sequence x allowing for multiple matches of the domain model $\mathbf{D}_i$ to the sequence. This is called the *sequence score*. HMMER also calculates the contribution from each of the repeated domains $\mathbf{D}_i$, which is called the *domain score*. Pfam enforces a threshold on both the domain and sequence scores, whereas eq. 2.11 just shows the domain score threshold.

Comparison of eqs. 2.10 and 2.11 reveals that the standard approach is essentially equivalent to ignoring the context term . My approach is to maximise the sentence score $\mathbf{SS}_{x,T,M}$ given in eq. 2.10 over all domain sentences $\mathbf{D}$, using the Pfam domain threshold for $\tau_{\mathbf{D}_i}$, and the HMMER domain score for $\frac{P(x_{[s_i,e_i]}|\mathbf{D}_i)}{P(x_{[s_i,e_i]}|\mathbf{R})}$.

2.3.2 Context model and smoothing strategy

The combined taxonomic and language context model is parameterised by considering a different Markov language model $\mathbf{M}_T$ for each taxonomy T. Begin and end states are included in the modelling in order to capture associations of domains with the beginning and end of proteins. A Markov model of order k asserts that the conditional probability of the i^{th} domain given all preceding domains is only dependent on the k preceding domains:

$$P(\mathbf{D}_i|\mathbf{M}_T, \mathbf{D}_1 \ldots \mathbf{D}_{i-1}) = P(\mathbf{D}_i|\mathbf{D}_{i-k} \ldots \mathbf{D}_{i-1}). \tag{2.12}$$

The terms in eq. 2.12 are calculated using the observed counts in the Pfam database (denoted by $\mathbf{N}$) and are smoothed recursively using lower order domain contexts and higher taxa as described for speech recognition. In the following, T_0 denotes the species of the protein in question, T_j the j^{th} parent taxon and T_m is the root of the taxonomy. For a fixed taxon T_j the probabilities are smoothed over domain contexts:

$$\begin{aligned}\hat{P}(\mathbf{D}_i|\mathbf{M}_{T_j},\mathbf{D}_{i-k}\dots\mathbf{D}_{i-1}) =& (1-\alpha)\cdot\left(\frac{\mathbf{N}(T_j,\mathbf{D}_{i-k},\dots\mathbf{D}_{i-1},\mathbf{D}_i)}{\mathbf{N}(T_j,\mathbf{D}_{i-k},\dots\mathbf{D}_{i-1})}\right)\\ &+\alpha\cdot\hat{P}(\mathbf{D}_i|M_{T_j},\mathbf{D}_{i-k+1}\dots\mathbf{D}_{i-1})\end{aligned} \tag{2.13}$$

$$\hat{P}(\mathbf{D}_i|\mathbf{M}_{T_j},\mathbf{D}_{i-1}) = (1-\alpha)\cdot\left(\frac{\mathbf{N}(T_j,\mathbf{D}_{i-1},\mathbf{D}_i)}{\mathbf{N}(T_j,\mathbf{D}_{i-1})}\right)+\alpha\cdot\hat{P}(\mathbf{D}_i|\mathbf{M}_{T_j}) \tag{2.14}$$

$$\hat{P}(\mathbf{D}_i|\mathbf{M}_{T_j})) = \frac{\mathbf{N}(T_j,\mathbf{D}_i)}{\sum_{\mathbf{D}}\mathbf{N}(T_j,\mathbf{D})}. \tag{2.15}$$

The sum in eq. 2.15 is over all domain occurrences in the Pfam database. The interpolation parameter α is a fixed constant between 0 and 1. Back-off estimation, as described for speech recognition, allows α to be a decreasing function of the amount of context $\mathbf{N}(T_j,\mathbf{D}_{i-k},\dots,\mathbf{D}_{i-1})$. This was investigated and not found to significantly improve the classification.

Next, contributions from higher order taxa are recursively interpolated

$$\begin{aligned}P(\mathbf{D}_i|\mathbf{M}_{T_j},\mathbf{D}_{i-1}\dots\mathbf{D}_{i-k}) =& (1-\beta)\cdot\hat{P}(\mathbf{D}_i|\mathbf{M}_{T_j},\mathbf{D}_{i-k}\dots\mathbf{D}_{i-1})\\ &+\beta\cdot P(\mathbf{D}_i|\mathbf{M}_{T_{j+1}},\mathbf{D}_{i-k}\dots\mathbf{D}_{i-1})\end{aligned} \tag{2.16}$$

$$P(\mathbf{D}_i|\mathbf{M}_{T_m},\mathbf{D}_{i-k}\dots\mathbf{D}_{i-1}) = \hat{P}(\mathbf{D}_i|\mathbf{M}_{T_m},\mathbf{D}_{i-k}\dots\mathbf{D}_{i-1}). \tag{2.17}$$

The parameter β represents the degree to which the estimation is based on nodes higher up in the taxonomy rather than the leaves. Note that this strategy is a smoothing strategy which recursively interpolates counts of species which are similar according to the NCBI taxonomy. In order to avoid over-fitting a taxonomy which has low coverage in Uniprot, only those nodes in the taxonomy below which there is a sufficient sample size, $10,000$ proteins in this case, are retained. For proteins which have a species T_0 which does not meet this criteria, T_0 is set equal to the first ancestor taxonomy in the modified taxonomy tree (which may be the root of the tree, if none of the kingdom-specific ancestor taxa meet the sample size criteria).

The interpolation parameters can be trained from training data which is held over from generating the counts for the context models. All that is required is some form of objective function, and then an optimization technique can be used to find the parameters which optimise the objective function. In the results section, held data from the SCOP test are used to estimate these interpolation parameters.

2.3.3 Context score of a domain in a protein with fixed context

I need to consider how to score an arbitrary Pfam domain instance on a protein with fixed context (i.e. the other domains on the protein are already known). This is required for the SCOP test in section 2.4.1. My approach is to consider the difference between the sentence score $\mathbf{SS}_{x,T,M}$ for the domain sequence including and excluding the domain in question. Denote by $\mathbf{d}_l$ the Pfam family which I am scoring, and by $\mathbf{D}$ the fixed (pre-annotated) context of the protein such that no $\mathbf{D}_i$ in $\mathbf{D}$ overlaps with $\mathbf{d}_l$. Then, define the sentence score for a single domain as

$$\mathbf{SS}_{x,T,M}(\mathbf{d}_l) = \mathbf{SS}_{x,T,M}(\mathbf{D} \bigcup \mathbf{d}_l) - \mathbf{SS}_{x,T,M}(\mathbf{D} \setminus \mathbf{d}_l) \qquad (2.18)$$

2.3.4 Dynamic programming algorithm

The space of all potential domain assignments for a particular protein is large, and hence an algorithm which concentrates on searching probable domain assignments is required. My approach is to first run HMMER against the protein for each Pfam family, keeping only those hits which have HMMER e-value less than 1000. In this way, a list $\mathbf{d} = \mathbf{d}_1 \ldots \mathbf{d}_m$ of potential domains is obtained, ordered by end position, with corresponding amino acid fragments $x_{[s_i,e_i]}$. The search space is now all possible subsequences of domains in this list. The search through this reduced space is optimized using a dynamic programming technique.

Firstly, assume that the language model is a first order Markov model. In that case, the goal is to find the domain sentence $\mathbf{D} = \mathbf{D}_1 \ldots \mathbf{D}_n$, a sublist of $\mathbf{d}$ which maximises the protein log-odds score $\mathbf{SS}_{x,T,M}(D)$, where

$$\mathbf{SS}_{x,T,M}(\mathbf{D}) \quad = \textstyle\sum_{i=1}^{i=n+1} H(\mathbf{D}_i) + C(\mathbf{D}_i|\mathbf{D}_{i-1}) \qquad (2.19)$$

$$H(\mathbf{D}_i) \quad = \mathrm{HMMER}(\mathbf{D}_i) - \tau_{\mathbf{D}_i} \qquad (2.20)$$

$$C(\mathbf{D}_i|\mathbf{D}_{i-1}) \quad = \log\left(\tfrac{P(\mathbf{D}_i|\mathbf{D}_{i-1})}{P(\mathbf{D}_i)}\right). \qquad (2.21)$$

Note that $H(\mathbf{D}_i)$ is just the HMMER score for the domain minus the threshold, and that $C(\mathbf{D}_i|\mathbf{D}_{i-1})$ is termed the transition score. Denote the begin and end states as $\mathbf{D}_0, \mathbf{D}_{n+1}$ respectively, so that $C(\mathbf{D}_1|\mathbf{D}_0)$ is the transition score coming from the begin state and $C(\mathbf{D}_{n+1}|\mathbf{D}_n)$ is the transition score going to the end state. As the end state contributes no sequence-based score, $H(\mathbf{D}_{n+1})$ is set to zero.

Define $\mathbf{D}^i$ to be the highest scoring domain sentence which ends in domain $\mathbf{d}_i$ without overlaps. The following recursion relation then applies:

$$\mathbf{SS}_{x,T,M}(\mathbf{D}^i) = H(\mathbf{d}_i) + \max_{e_j<s_i}\{\mathbf{SS}_{x,T,M}(\mathbf{D}^j) + C(\mathbf{d}_i|\mathbf{d}_j)\}, \tag{2.22}$$

where the condition $e_j < s_i$ ensures that the maximising sentence does not contain domain overlaps. Then set

$$\mathbf{D}^i = \{\mathbf{D}^j, \mathbf{d}_i\} \tag{2.23}$$

where $\mathbf{D}^j$ maximises eq. 2.22. Repeated application of eq. 2.22 and eq. 2.23 for $i = 1 \ldots m+1$ gives the maximising sentence $\mathbf{D} = \mathbf{D}^{m+1}$ required by eq. 2.19 (again, I use the convention that $\mathbf{d}_{m+1}$ is the end state, so that $\mathbf{D}^{m+1}$ is interpreted as the maximising sentence ending with the end state).

The assumption that the Markov model $\mathbf{M}$ is first order is now relaxed, and $C(\mathbf{D}_i|\mathbf{D}_{i-1})$ is replaced with $C(\mathbf{D}_i|\mathbf{D}_{i-1} \ldots \mathbf{D}_{i-k})$. Equation eq. 2.22 now becomes

$$\begin{aligned}\mathbf{SS}_{x,T,M}(\mathbf{D}^i) = H(\mathbf{d}_i) + \\ \max_{e_{j_1}<s_{j_2}<e_{j_2}<\ldots<s_{j_k}}\{\mathbf{SS}_{x,T,M}(\mathbf{D}^{j_1,\ldots j_k}) + C(\mathbf{d}_i|\mathbf{d}_{j_k}, \ldots \mathbf{d}_{j_1})\},\end{aligned} \tag{2.24}$$

and so the strategy outlined above is no longer guaranteed to return the highest scoring sequence under the language model. However, this strategy is still used in this case, and has been found to still work well in practice.

2.3.5 Incorporating the sequence score threshold

As mentioned above, Pfam uses a *sequence score* threshold in addition to the domain score threshold given in eq. 2.11. This thresholding is equivalent to a threshold on the sum of log-odds scores contributed by all instances of a particular domain type on a protein (for instance the sum of all of the zf-C2H2 domain scores). As the method applies Pfam thresholds, it must also apply a sequence score filter as a post-processing step to retain consistency with Pfam.

In order to do this, the maximising domain sentence is obtained as before. The total score for the maximising sentence comprises the sum of HMMER scores (left-hand bracket of eq. 2.10) and the context score (the right-hand bracket in eq. 2.10). As before, the total HMMER score for each type of domain on the maximising sentence is summed to give a sequence score for that domain type. Now, the context component of the score is distributed amongst each of the sequence scores such that as many domain types score above the sequence threshold as possible. To do this, assuming a positive context score, simply order the domain types according to sequence score and allocate to the first sub-threshold domain type as much context score is required to meet the sequence score threshold. Repeat this step until the context score has been completely distributed.

2.3.6 Variable length Markov model

The fixed-order Markov model has a significant drawback: the lengths of commonly occurring domain architectures are not fixed; some patterns are first order (CBS domains often occur in pairs), while many patterns have a higher order (the group of RNA polymerase RBP1 domains commonly occur in groups of seven). Restricting to a fixed order Markov model will degrade the ability of the model to recognize patterns of arbitrary length. Instead, for each proposed context $\mathbf{D}^j$ from eq. 2.22 in the dynamic programming algorithm, a different order k for M is chosen which is the maximum order which is observed in the training database. More precisely, labelling $\mathbf{D}^j = \mathbf{D}^j_1 \dots \mathbf{D}^j_{n_j}$ the order k is chosen to be the largest order with non-zero training set count $\mathbf{N}(\mathbf{D}^j_{n_j - k} \dots \mathbf{D}^j_{n_j})$. As this does not depend on the current domain $\mathbf{d}_i$, eq. 2.12 still defines a consistent probability distribution over domains. In practice, however, to cut down on memory requirements for storing counts of arbitrary length, I restrict $k \leq 4$.

This approach is an example of decision tree modelling which is commonly used in language modelling. Decision trees partition domain histories $\mathbf{D}^j$ into equivalence classes $\Phi_1 \dots \Phi_M$ with a corresponding probability distribution $P(\mathbf{D}_i|\Phi_l)$. My approach partitions on the basis of the longest domain context which has been observed in the training set. It is straightforward to develop more complicated decision rules, and this remains a basis for further investigation. My approach is also similar to the interpolated Markov chain approach used by Salzberg [SPD$^+$99] in gene prediction.

2.3.7 Incorporating Pfam clans

The Pfam project groups together closely related Pfam families into Pfam clans. Pfam enforces an overlap rule: the Pfam threshold must be set to ensure that no distinct significant Pfam family matches overlap. Clans were created to relax this rule – that is, two families from the same clan are allowed to have significant matches which are overlapping, and the family which scores highest above its own threshold is annotated as the matching Pfam family. From the point of view of language modelling of domains clans can be seen as variants of a single domain (in much the same way that different phonetic representations of a word are the same word). I have taken the approach that from a language modelling point of view, Pfam families from different clans are considered to be from the same family, and hence their counts are aggregated. This only applies for training and scoring the the transition scores $C(\mathbf{D}_i|\mathbf{D}_{i-1}, \mathbf{D}_{i-k})$ but the HMMER component $H(\mathbf{D}_i)$ remains specific to the domain which is being scored. Importantly, the threshold remains domain (not clan) dependent, as thresholds may still vary substantially within a clan (particularly if one clan member is a fragment of another).

Clans and context modelling have had a mutually beneficial existence in Pfam. Pfam annotators use context domain hits to guide their decisions about new clans to build, and grouping Pfam families into clans means that context modelling has more information (as more patterns are observed) with which to score domain architectures.

2.3.8 Significance scores

The Pfam database maintains for each domain hit an e-value score as well as a log-odds score. The e-value score for a domain is the number of hits which would be expected to have a score greater than or equal to the score of the domain in a random database of the same size. It is calculated for each Pfam family by fitting an extreme value distribution (EVD) to the bit scores of hits of that family against a set of randomly generated proteins, as implemented in the *hmmcalibrate* program of the HMMER package. The e-value score does not directly affect the assignment of domains in Pfam as manually created thresholds are used instead. However, the significance of domain matches is important to consider as it is used by end users when evaluating marginal hits. Moreover, significance scores can be used to compare the reliability of hits from different Pfam families, whereas log-odds scores cannot. Significance values are required in the SCOP test to generate aggregate ranked lists of domain matches. Thus it is important to consider the effect of language modelling on significance scores.

One possibility is to use the unmodified EVD parameters calculated by *hmmcalibrate* to calculate the significance of HMMER+context scores. This is the approach pursued in the SCOP test in section 2.4.1. An alternative strategy is to score the HMMER+context model on randomly generated proteins in order to generate a modified EVD. As the significance score relates to a particular domain rather than the entire domain sentence, the method described in section 2.3.3 is used to calculate the HMMER+context score for the domain as the difference of the HMMER+context score of the maximising sentence with and without the domain in question. As in *hmmcalibrate* the HMM is required to pass through the given domain at least once. Note that in almost all cases, the language model uses a start → domain → end architecture as it finds no other domains with scores above threshold to include in the calculation. In this case, all of the start to domain and domain to end transition scores will be attributed to the domain.

This process is demonstrated on two Pfam families, WD40 and pkinase as shown in fig. 2.6. Two different types of behaviour are observed. In one case, pkinase commonly occurs by itself on a protein, and hence hits to random proteins typically have their scores enhanced slightly by the language model, so that the EVD shifts to the right. However, real hits also have their scores enhanced. Furthermore, in the case of a single domain protein, the increase will be the same as the shift in the EVD, so that the significance of the hit remains unchanged. In contrast, hits to the pkinase domain in atypical contexts will not have their scores enhanced, and so their significance will decrease. The other example, WD40 commonly occurs in repeats of 5-8 units; so that individual random hits are penalised under the language model (by about 4 bits) and so the EVD shifts to the left. The language model enhances the score of real hits (as they do occur in the appropriate repeating pattern), thus providing the compound effect of increasing the score of real hits and increasing the significance of hits at a given score. To summarise, the effect of language modelling on significance scores appears to be either neutral, in the case in which the scores of random and real hits are shifted by the same amount, or more discriminatory, in the case of decreasing random scores and increasing real scores.

A weakness of this approach to calculating significance scores is that it considers random

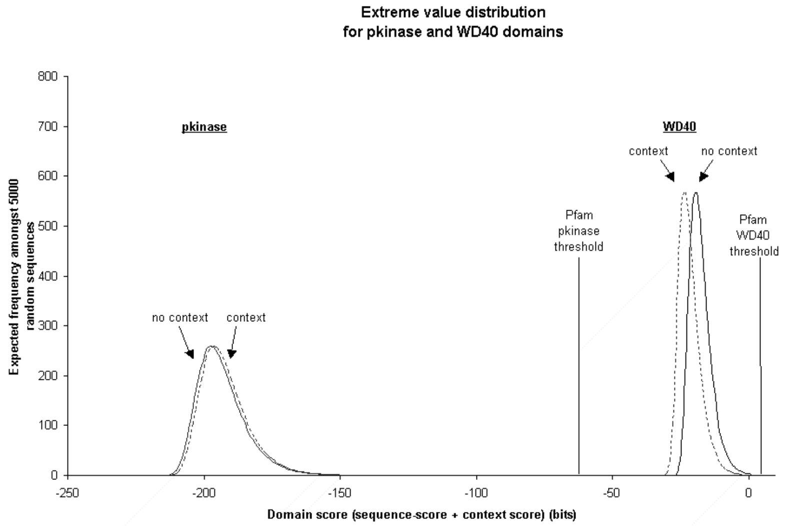


Figure 2.6: Extreme Value Distribution (EVD) curves calculated for pkinase and WD40 Pfam domains. The solid lines are the standard EVD curves calculated using HMMER. The dashed lines use the language modelling method, and hence take contextual information into account. For almost all sequences, this results in a domain sentence consisting of a BEGIN state followed by the given domain and ending in an END state. WD40 is commonly found in groups of 5 to 8 tandem repeats, so that single random WD40 hits are penalised by the language model. The WD40 EVD shifts 4.0 bits to the left. On the other hand pkinase often occurs by itself on a protein, and hence random single pkinase repeats gain slightly under the language model. The pkinase EVD shifts 1.1 bits to the right.

proteins for calculating the language model component of the score, whereas false positive hits in real proteins do not have random protein as context. This has not been further investigated.

2.3.9 Implementation

A major implementation challenge was to store efficiently in memory the counts of occurrence patterns of domains and species used in eqs. 2.13, 2.14, 2.15. These counts are central to the dynamic programming algorithm described above, and speed of accessing these counts is critical. Note that the counts are stored, rather than the smoothed probabilities, as the space of possible domain and taxonomy combinations is much vaster than the space of observed combinations. A context map is stored, which contains as keys every observed 1-mer, 2-mer .. k-mer observed in Pfam (with k normally set at 4). These keys map to a secondary map, in which each observed taxonomy from the reduced taxonomy tree maps to the number of observations of the given domain sequence k-mer in proteins of this taxonomy or with the taxonomy as ancestor. In order to facilitate rapid access to the counts to compute the smoothed probabilities, the context map is stored as a red-black tree [CLRS01]. The smoothing equations eq. 2.13 recursively interpolate from higher-order to lower-order contexts. However the counts are stored and accessed in the reverse order, progressively narrowing down from general to more and more specific contexts. This is achieved by first constructing an ordering of Pfam domains. This ordering is used to infer an ordering on domain sequences working from right to left – that is for two given domain sequences the final position is first compared, then, if this is equal the penultimate position is compared, etc. If the two domain sequences are equal in all positions, but one is shorter than the other, then the shorter sequence is ordered ahead of the longer sequence. This ordering is used to create the red-black map. Now consider eq. 2.13. The successive numerators require the counts $\mathbf{N}(T_j, \mathbf{D}_{i-k}, \ldots \mathbf{D}_i)$, $\mathbf{N}(T_j, \mathbf{D}_{i-k+1}, \ldots \mathbf{D}_i)$, ... $\mathbf{N}(T_j, \mathbf{D}_i)$, which are obtained in reverse order. Firstly the node in the red-black map is found below which all domain sequences end in $\mathbf{D}_i$, as all subsequent counts will be from this sub-tree. The first position in this sub-map is the last of the counts required. This process is continued, progressively narrowing down the sub-tree of counts. For successive denominators, the same strategy can be pursued, but starting with all domain sequences ending in $\mathbf{D}_{i-1}$. The counts over all ancestral T_j are all collected at the same stage. This search is not optimized in the same way as the maps are much smaller, so a standard hashing strategy is sufficient.

As an example, consider scoring an example transition $C(\mathbf{D}_i|\mathbf{D}_{i-k},\ldots\mathbf{D}_i)$. Consider the domain sequence $\mathbf{D}_0$ =BEGIN, $\mathbf{D}_1$ = C2-set, $\mathbf{D}_2$ = ig and I will show how to calculate eq. 2.21. Let the taxonomy of the protein be (Eutheria, Coelomata, Eukaryota, root). I assume $\alpha = 0.7$ and $\beta = 0.35$. The array of counts required for smoothing is given as follows:

$$\begin{array}{lcccc} & T_0 & T_1 & T_2 & T_3 \\ \mathbf{N}(T_j,\mathbf{D}_0,\mathbf{D}_1) & 3224 & 43394 & 5029 & 5210 \\ \mathbf{N}(T_j,\mathbf{D}_1) & 17894 & 255714 & 29972 & 30256 \\ \mathbf{N}(T_j) & 379460 & 618859 & 1376701 & 3005810 \end{array} \tag{2.25}$$

$$\begin{array}{lcccc} & T_0 & T_1 & T_2 & T_3 \\ \mathbf{N}(T_j,\mathbf{D}_0,\mathbf{D}_1,\mathbf{D}_2) & 2428 & 3355 & 3946 & 4011 \\ \mathbf{N}(T_j,\mathbf{D}_1,\mathbf{D}_2) & 12132 & 17657 & 21018 & 21119 \\ \mathbf{N}(T_j,\mathbf{D}_2) & 17894 & 25571 & 29972 & 30256 \end{array} \tag{2.26}$$

From these counts, I calculate the probabilities, and finally log-odds scores.

$$\begin{array}{lcccc} & T_0 & T_1 & T_2 & T_3 \\ \frac{\mathbf{N}(T_j,\mathbf{D}_0,\mathbf{D}_1,\mathbf{D}_2)}{\mathbf{N}(T_j,\mathbf{D}_0,\mathbf{D}_1)} & 0.75 & 0.77 & 0.78 & 0.77 \\ \frac{\mathbf{N}(T_j,\mathbf{D}_1,\mathbf{D}_2)}{\mathbf{N}(T_j,\mathbf{D}_1)} & 0.68 & 0.69 & 0.70 & 0.70 \\ \frac{\mathbf{N}(T_j,\mathbf{D}_2)}{\mathbf{N}(T_j)} & 0.05 & 0.04 & 0.02 & 0.01 \\ P(\mathbf{D}_2|T_j,\mathbf{D}_0,\mathbf{D}_1) & 0.39 & 0.40 & 0.39 & 0.38 \\ P(\mathbf{D}_2) & & & & 0.01 \\ C(\mathbf{D}_2|\mathbf{D}_0,\mathbf{D}_1) & & & & 5.24 \end{array} \tag{2.27}$$

So the transition score is 5.24 bits. In other words, in eutherian mammals it is $2^{5.24} = 38$ times more likely to see a ig as the second domain in a protein following a C2-set domain than it is in a random protein.

2.4 Results

Figure 2.7 shows the processes carried out this chapter. The results are split into two sections, the SCOP test and the Pfam scan. The training set for the language model consisted of Pfam release 15 and proteins from the Uniprot [ABW$^+$04] database consisting of Swissprot release

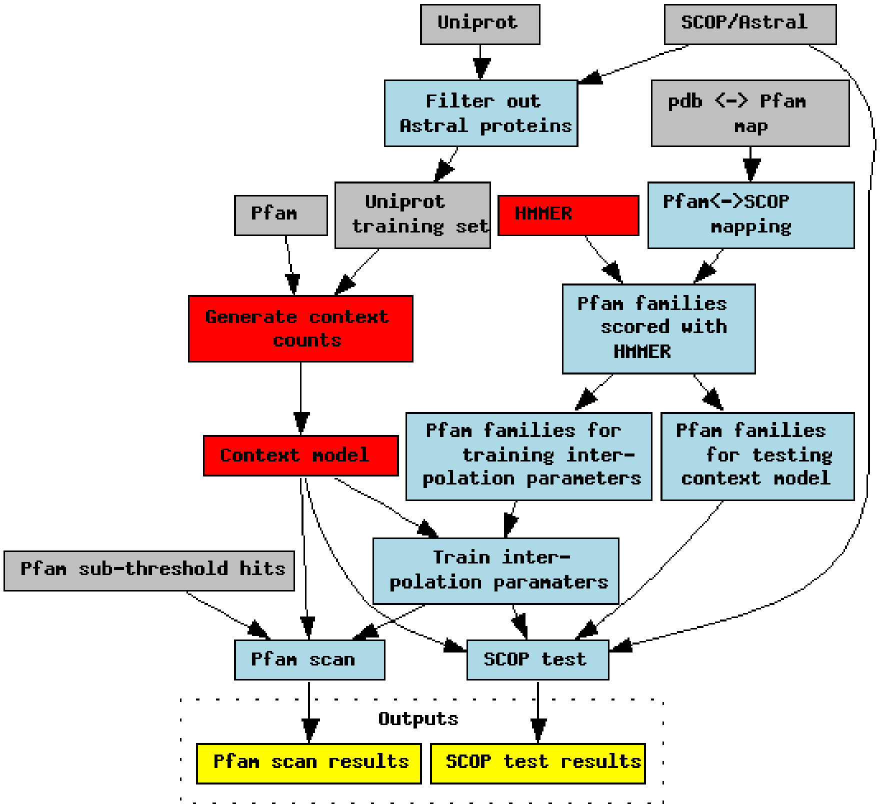


Figure 2.7: Conceptual diagram of processes and data in this chapter. Inputs are shown in grey, and outputs in light blue, with intermediate steps in yellow, and software steps in red.

44.0 and SP-TrEMBL27.0, with all proteins which match proteins from the ASTRAL protein set (filtered to a maximum of 40% identity between any two proteins in the set) removed. This set consisted of 982, 523 proteins, which only includes those proteins which have at least one annotated Pfam domain.

2.4.1 SCOP test

In order to test objectively the ability of the language model to detect protein domains, I use the SCOP test, initially developed by Brenner et al. [BCH98] and subsequently used by many authors to evaluate homology prediction algorithms (e.g. [MG02]). The SCOP database classifies all proteins of known structure [HMBC97] in terms of protein domains. Multi-domain proteins are split into component protein domains, which are classified hierarchically in four levels: family, superfamily, fold and class. Sequences belonging to the same family share sequence similarity, suggesting a common function and implying a clear common evolutionary origin; families are clustered into superfamilies on the basis of structural similarity suggesting a probable common evolutionary origin; superfamilies are grouped into folds on the basis of similar secondary structure topology. ASTRAL is a database of protein sequence fragments of known structure, annotated with SCOP family classifications [CWC$^+$02]. ASTRAL provides protein sequences filtered to various levels of sequence similarity. The SCOP test works by running a given algorithm and domain model over all proteins classified by SCOP, and comparing domain family predictions with the known structural class. In this way it is possible to independently identify proteins homologous to the given domain family (all proteins belonging to the same SCOP superfamily) and proteins which are non-homologous (all proteins belonging to a different SCOP fold). Proteins belonging to the same fold but different superfamily are not classified as homologous or non-homologous.

The SCOP test was modified in order to apply it to the domain models in Pfam. Using the file *pdbmap* (available at ftp://ftp.sanger.ac.uk/pub/databases/Pfam/pdbmap) I obtain a list of all proteins in which a Pfam domain annotation overlaps a PDB structure . The coordinates (with respect to the Uniprot protein sequence) of both the PDB structure and the Pfam domain are provided in this file. The PDB structure is classified by SCOP. Providing a given Pfam domain overlaps one and only one SCOP superfamily, I classify all SCOP proteins in this superfamily as homologous to the Pfam domain, and all proteins outside the fold to which this superfamily belongs as non-homologous. I identify 1970 Pfam families in Pfam release 15 which satisfy this criteria. Of these, I use 500 to train the interpolation parameters of the context models. The remaining 1470 form the test set of Pfam families for the SCOP test.

For a given algorithm and a given Pfam family, the SCOP test proceeds by scoring every protein in the ASTRAL filtered sequence set (to a maximum of 40% shared identity in this case), and generating a list of proteins ranked according to model log-odds score. The ultimate goal of homology detection is to score all homologous proteins above all non-homologous proteins. One simple measure of relative success is the number of true homologies scored above the highest scoring non-homologous sequence, which I shall refer to as the 'over the top' score (OTT). An alternative is a *coverage vs error* curve which plots at each point in the ranked list the total number of homologous proteins (true-positive) above this point on the y-axis vs the number of non-homologous (false positive) above this point on the x-axis. A randomly ranked list would give on average an equal proportion of homologous and non-homologous sequences identified. For a given error rate, a higher curve is a more effective classifier of homologous proteins, and the area under the curve is another measure of overall success. The minimum error rate (MER), which is the minimum of the sum of number of homologous sequences classified as non-homologous and non-homologous sequences classified as homologous, can also be used.

If instead of ranking according to model score the list is ranked according to e-value significance, then it is possible to generate an aggregated ranked list of significance across multiple domain models. From this list a score representing the effectiveness of the algorithm across all domain families can be obtained, using either the OTT, MER or area under the coverage versus error curve. The ranking by significance is necessary as the log-odds scores between models are not comparable.

The SCOP test was carried out on the following variants of the context models described in the previous section: HMMER alone; HMMER with a digram language model, denoted HMMER+2gram (which implies that a single domain is considered as context); HMMER+3gram; HMMER+4gram; HMMER with taxonomy context (denoted HMMER+taxonomy) and HMMER+4gram+taxonomy. It can be seen from the following results that the 4gram model is a small improvement on the 3gram model. A HMMER+7gram+species model was tested to observe the effect of longer context, but it was not found to improve results beyond the HMMER+4gram+context model.

In order to apply the language models, it was necessary to identify the protein sequence in Uniprot which matched each of the protein fragments in the ASTRAL set, so that I could use Pfam to assign the domain context and also obtain the taxonomic position of the protein from the NCBI taxonomic code assigned by Uniprot to each protein. As noted above, ASTRAL contains protein fragments, so it is also necessary to assign the correct position of the protein fragment on the Uniprot protein. This is achieved with the *pdbmap* file discussed above. The HMMER+context model score for a particular Pfam domain was obtained as the difference between the context score of the full domain sequence including the Pfam domain and the context score excluding the context domain, as given by equation eq. 2.18.

The interpolation parameters were trained on 500 of the 1970 Pfam families with the remainder forming the test set of Pfam families. The sum of individual family OTT scores was used as the objective function to train the taxonomy and domain context interpolation parameters. This score was chosen as it replicates most closely the objective of improving Pfam annotation, for which a threshold is manually curated for each family with the aim that there are no false positives. The optimal parameters from this set were $\alpha = 0.7$ for domain context, and $\beta = 0.35$ for taxonomic context.

Figure 2.8 displays the coverage versus error curve over all Pfam domains tested (with the results ranked by significance). HMMER+4gram+taxonomy identifies 3% more homologous proteins at an error rate of 1000 proteins. Table 2.4.1 shows summary measures of the performance of each of the context models. From the point of view of using the method to improve Pfam domain annotation, the important measure is the sum of family OTT scores (column 4). HMMER+4gram+taxonomy improves this measure by 2.2%, implying that if the Pfam thresholds could be optimally selected, context models could increase the number of domains annotated by 2.2%. HMMER+4gram+taxonomy is substantially better under this metric than HMMER+4gram, indicating that taxonomy is useful in improving the context models. Taxonomy on its own generates a smaller improvement than the 4-gram but better than the 3-gram language model.

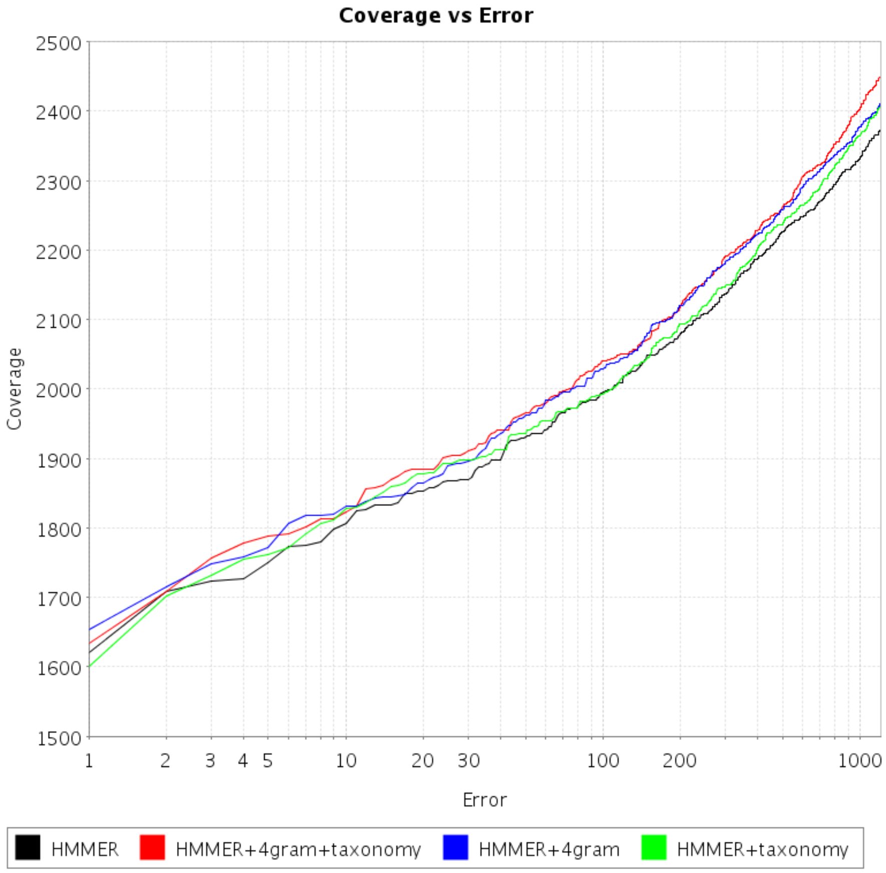


Figure 2.8: Coverage vs error curve for detection of remote homologies for aggregated results from 1470 Pfam families not used for training the interpolation parameters. The lines are black - HMMER score, green- HMMER+taxonomy , blue - HMMER+4gram, the red - HMMER+4gram+taxonomy. A higher line indicates a better classification of remote homologies. I display only up to 1000 false positives.

Method	# families with OTT		sum of family score		Aggregate score	
	Better	Worse	OTT	MER	OTT	MER
HMMER	-	-	3604	5092	1620	3692
+2gram	37	15	3638	5042	1650	3668
+3gram	46	20	3646	5041	1655	3662
+4gram	50	22	3657	5031	1654	3665
+taxonomy	53	34	3644	5064	1601	3687
+taxonomy +4gram	69	39	3682	5017	1634	3650

Table 2.1: Comparison of context models with HMMER, scored over the 1470 families not used for training the interpolation parameters.

For each method the number of false positive and false negative matches at a given e-value significance is plotted in figure 2.9. Context models improve error rates over a range of e-values less than 1.0 by reducing false negative matches with negligible impact on false positive matches. This demonstrates that at a given e-value threshold, HMMER+4gram+taxonomy has a lower error rate than HMMER alone. From the point of view of large scale classification of protein homology with profile HMMs this is an important result, as classification is often done on the basis of a global e-value threshold. This figure justifies to a certain extent the use of the same EVD on context adjusted scores, in that the false positive error curve is correctly calibrated with the HMMER false positive score. Note that no false positives are obtained with evalue of 10^{-3} or lower.

Figure 2.10 displays the domains which have the greatest increase and decrease in OTT score. In particular, C2-set gains 12 domains while Semialdehyde_dh loses 3 domains. In some cases the increase obtained by using a joint model is greater than the sum of the individual OTT score increases of the 4gram and context models (for example Laminin_EGF).

One family with significant improvement is the C2-set domain. C2-set is a member of the immunoglobulin superfamily clan in Pfam, and commonly co-occurs with other immunoglobulin superfamilies on a protein. HMMER alone scores 6 positive sequence from the

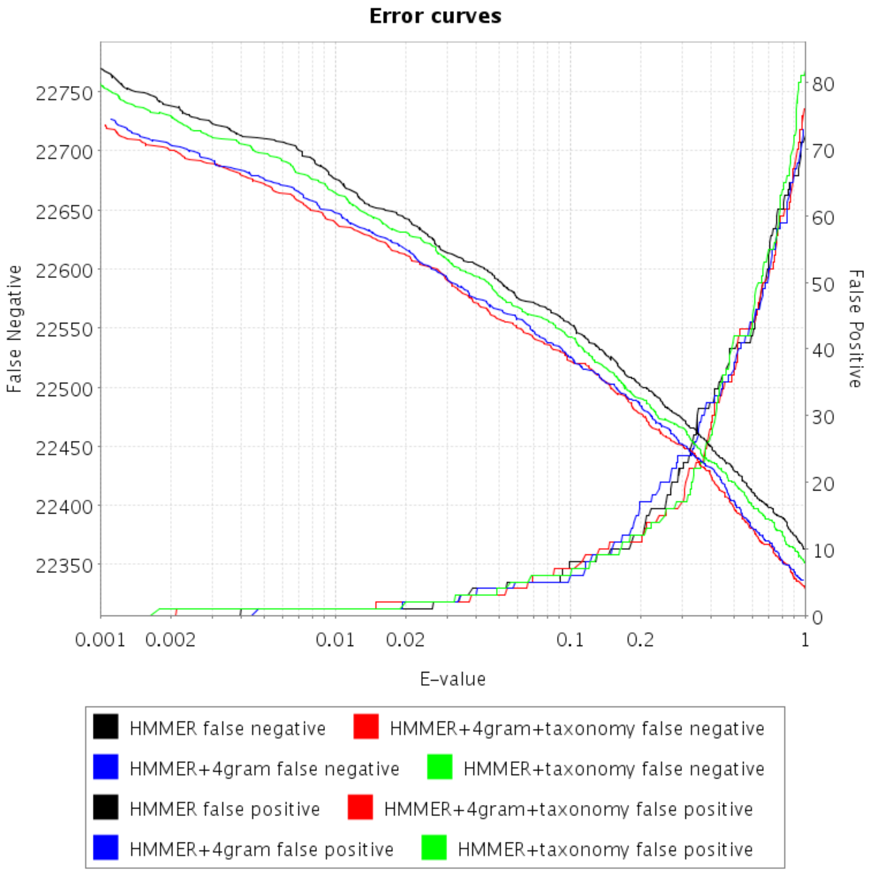


Figure 2.9: Number of false negative (upper six lines) and false positive (lower six lines) matches versus e-value threshold for HMMER (red lines) and context models. At a given e-value threshold, each of the models decreases false negative rates with negligible impact on false positive rates.

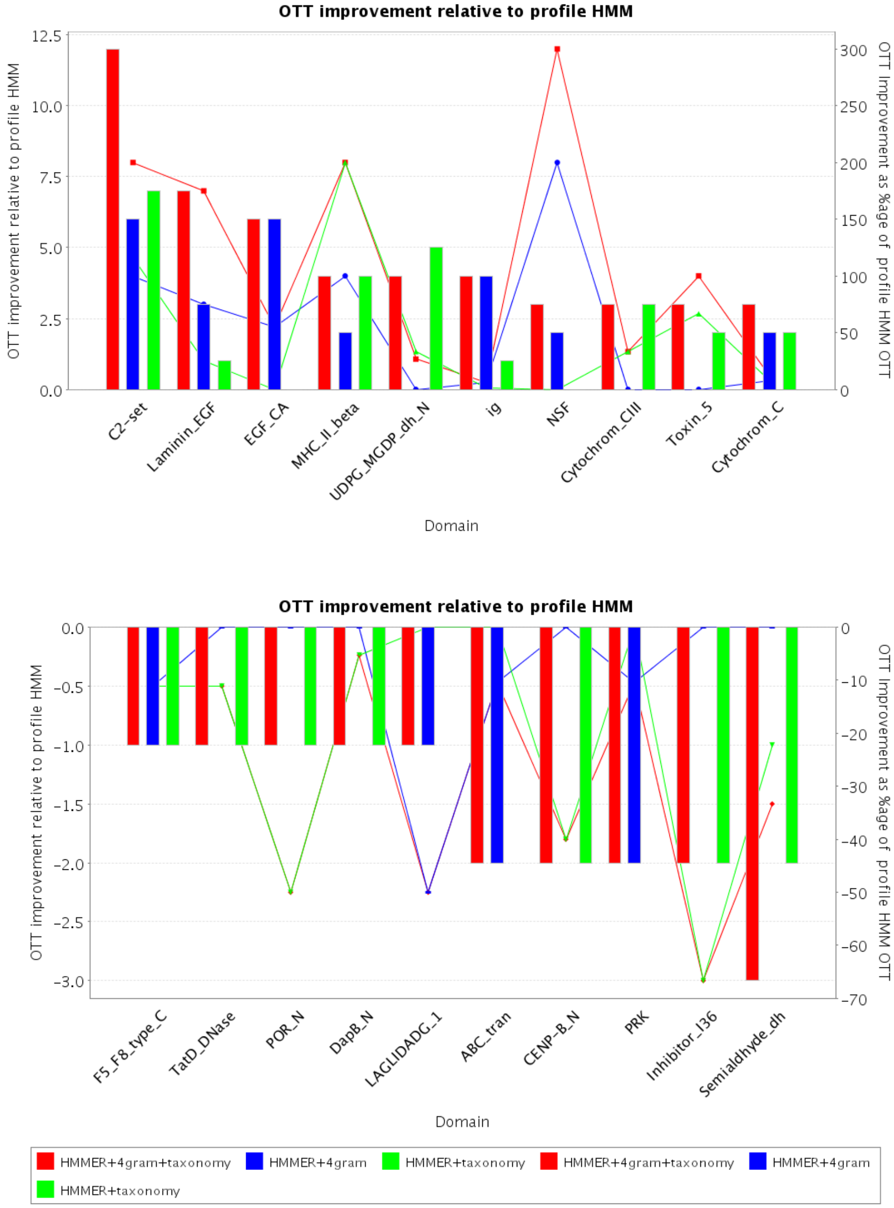


Figure 2.10: Pfam domains which have their OTT scores improved the most (upper graph) and decreased the most (lower graph), with OTT improvements relative to HMMER alone plotted for HMMER+taxonomy (blue), HMMER+4gram(red) and HMMER+4gram+taxonomy(black). The bars indicate absolute increase or decrease in OTT score, indicated on the left-hand y-axis. The lines indicate the percentage increase (or decrease) in OTT score, indicated on the right-hand y-axis.

ASTRAL test set above the first negative sequence, whereas HMMER+context+species scores 18 sequences above the first negative sequence. This improvement is obtained by increasing the significance of 12 homologous low significance scores and decreasing the significance of 3 non-homologous high significance scores. In the Pfam annotation, this domain is restricted to eutheria (placental mammals), however other members of immunoglobulin superfamily clan occur frequently in other verterbrata, and less commonly in other metazoa. The improvement in classification includes 11 vertebrate proteins and 1 insect protein. Figure 2.11 displays the significance scores for both HMMER and HMMER+4gram+taxonomy on this family.

2.4.2 Pfam scan

I scanned the Uniprot [ABW+04] database with all Pfam models to search for novel hits to these models. The same interpolation parameters were used as in the SCOP test. A HMMER+4gram+taxonomy language model was used, as the SCOP test demonstrated that this is the most sensitive of those context models tested.

The Pfam scan identifies 44792 new domain instances, which corresponds to 2.8% of the total number of domains previously scored as significant in Pfam under full-length models (Pfam also scores partial matches to Pfam domains). The new domain instances occur on 26458 proteins (which corresponds to 1.8% of the total number of proteins in Uniprot) and 3479 proteins which previously had no Pfam annotation (which corresponds to a 0.2% increase in sequence coverage). The new domain instances cover an additional 1.8m residues (Pfam full-length models previously covered 246m of 470m residues in Uniprot) which corresponds to a 0.38% increase in residue coverage. The new predictions are limited to 1245 domains, of which 344 domains contribute 95% of the new domain instances.

Figure 2.12 displays the families that the method detects. Figure 2.13 displays the length distribution of both new domains detected using context and the current Pfam annotation. Context domains have average length of 44 residues; the average length of Pfam domains is 183 residues. This is due to the over-representation of repeats in short Pfam families (and hence better contextual information) and a lower sequence-based signal-to-noise ratio for short families so that extra information is more likely to make a difference in detecting them.

Figure 2.14 shows how the impact of context varies across the taxonomic tree. In

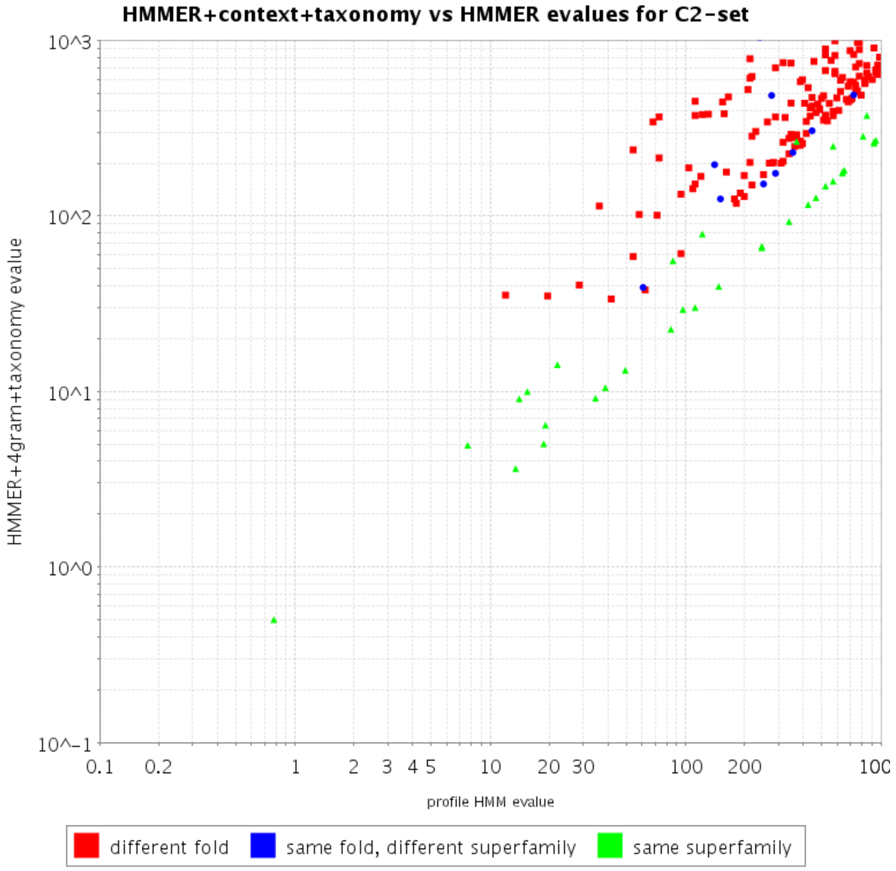


Figure 2.11: E-value significance scores for HMMER+4gram+taxonomy vs HMMER for C2-set domain, plotted on a log-log scale. The green dots represent sequences in the same SCOP superfamily (which are treated as homologous). The red dots represent sequences in different SCOP folds (which are treated as non-homologous). The blue dots represent sequences in the same SCOP fold but different superfamily (which are treated as neither homologous or non-homologous). Note that the four most significant matches (with e-value less than $1e-8$ under both HMMER and HMMER+4gram+taxonomy) are not shown. All 31 homologous sequences shown on this graph (green dots) fall below the $y=x$ line, and hence are more significant under HMMER+4gram+taxonomy than under HMMER.

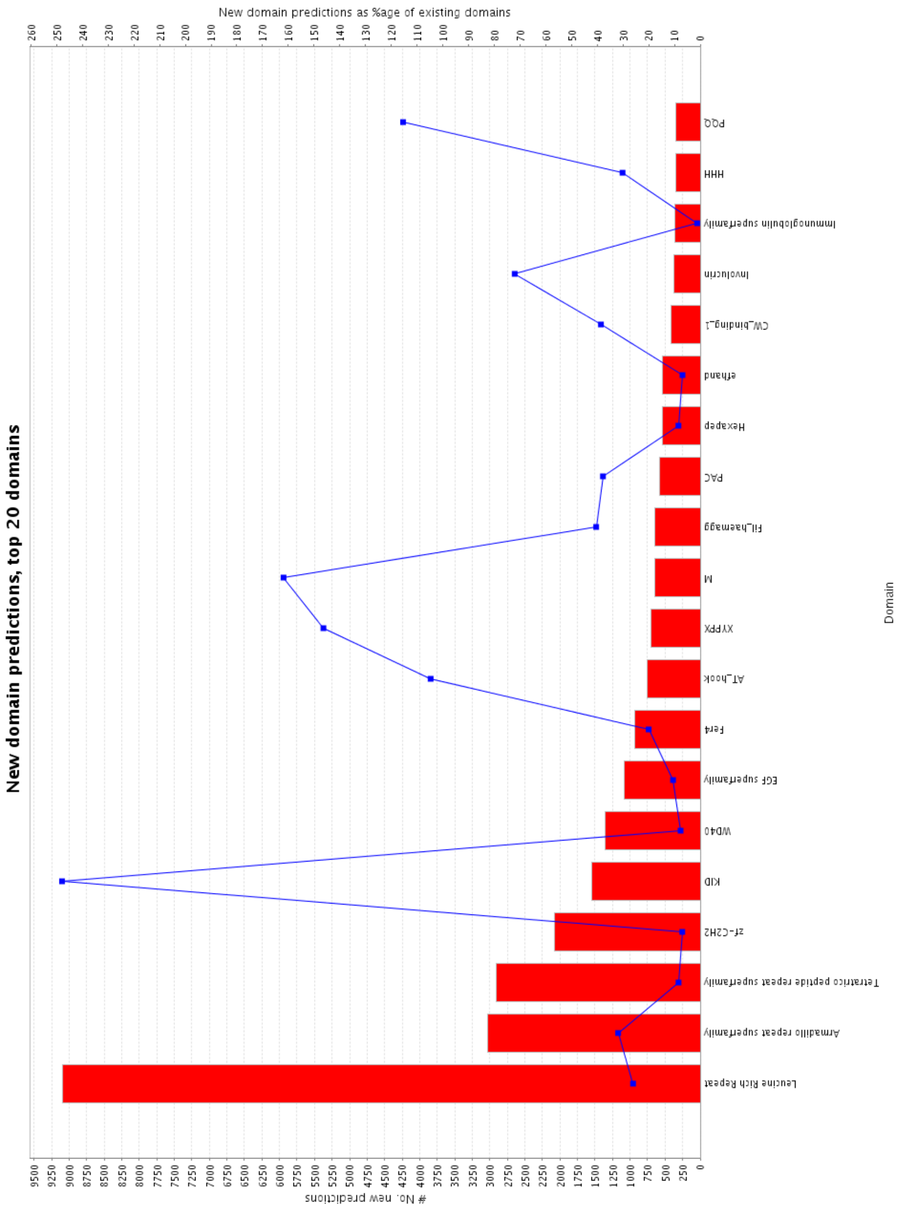


Figure 2.12: Domain occurrences amongst top 20 'context' families. The bars shows the absolute number of new predictions; the line line shows the percentage increase in that family.

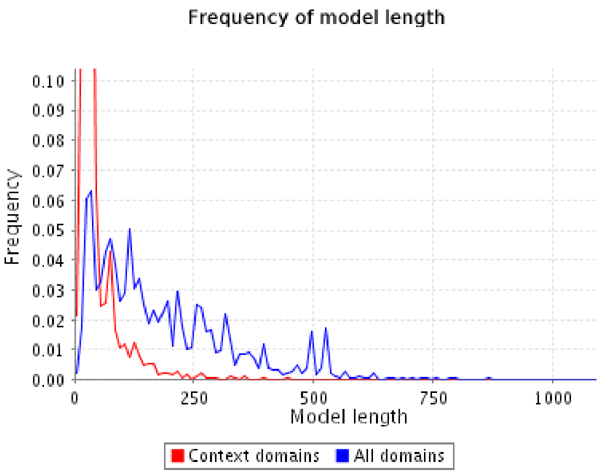


Figure 2.13: Length distribution for context domains(red) and Pfam domains (blue).

particular, context is not particularly effective in annotating Virus proteins. One possible explanation is that almost half the virus proteins in Uniprot are HIV proteins, and most of these are homologous proteins from different HIV strains, hence represents a much smaller pool of proteins with different domain architectures, each of which is already well understood. Context increases the number of Pfam annotations in bacteria and archaea by approximately 2% which is slightly below the average result. Context performs particularly well on eukaryotic proteins, increasing coverage by up to 6%. Table 2.2 suggests a weak relationship between the average number of domains per protein annotated with at least one Pfam domain and the increase in context domains.

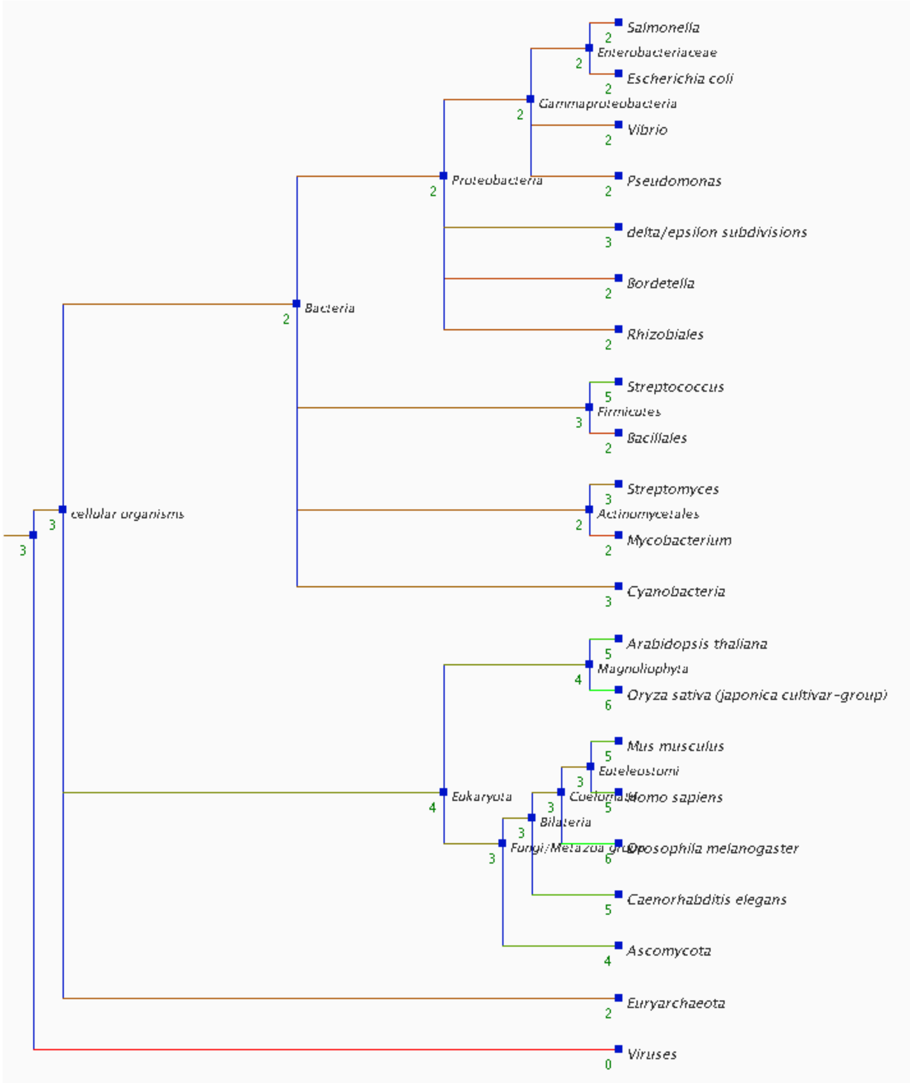


Figure 2.14: Percentage increase in domain occurrences by position in taxonomic tree. Each of the taxa displayed have more than 10,000 proteins in Uniprot (counting nodes which have the given node as ancestor). Nodes which have a single parent have been removed (for example HIV). Each node is annotated with the percentage increase in domain instances given by context at that level in the taxonomic tree. The branches above a given node are coloured according to the percentage increase, from green (high increase) to red (low increase).

Taxonomy	Percentage increase due to context domains	Average no. of domains per Pfam annotated protein
Drosophila melanogaster	6.4	2.5
Oryza sativa	6.4	2.0
Homo sapiens	4.6	2.6
Eukaryota	3.6	1.9
Bacteria	2.5	1.5
Archaea	2.4	1.5
Viruses	0.4	1.4

Table 2.2: Percentage increase in domain annotations due to context and average number of domains per protein annotated with at least one domain.

Figure 2.15 shows several examples of domains found by the context models without taxonomic context. Two TPR domains are found on the SR68_HUMAN protein, which has no TPR domains annotated in any of the protein databases. This protein is known to interact with SR72_HUMAN in the signal recognition particle [LPA$^+$93], which itself has a pair of annotated TPR domains. As TPRs are protein-protein interaction motifs, this suggests that the interaction between SR68 and SR72 may be mediated by this region. On the previously un-annotated E2BG_CAEEL protein I find an NTP_transferase domain, followed by three hexapep repeats, all raised above the noise by their mutual compatibility.

The method also predicts a previously un-annotated Tf_Otx domain in the cone rod homeobox protein (CRX), in *H. sapiens*, *R. norvegicus* and *M. musculus* (figure 2.15). CRX is a 299 amino acid homeodomain transcription factor which is primarily expressed in the rod and cone receptors of the retina [CWN$^+$97, FMC97]. CRX is highly conserved amongst mammalian species. CRX is known to share homology with Otx1 and Otx2, and contains a homeodomain near the N-terminus followed by a glutamine rich region, a basic region, a WSP motif, and an Otx-tail motif. The new Tf_Otx prediction extends over the un-annotated region: amino acids 164 to 250. This region encloses a valine to methionine mutation at position 242 associated with autosomal dominant cone rod dystrophy, which leads to early

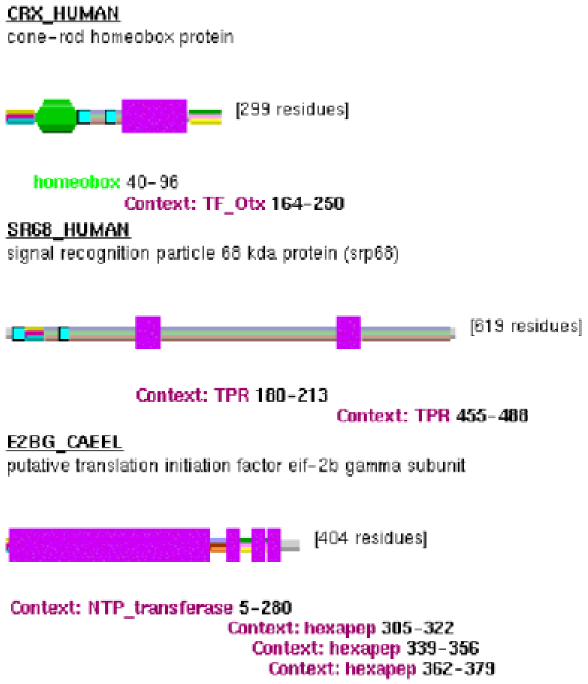
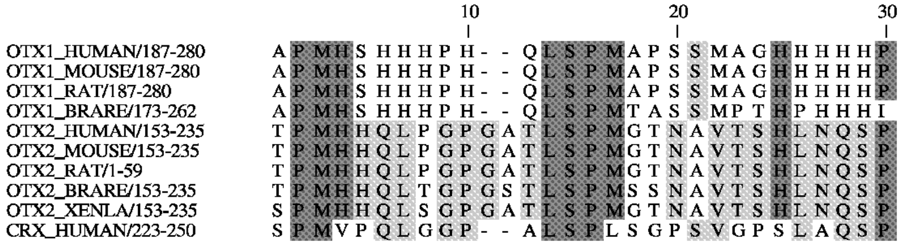


Figure 2.15: Examples of new context domains, indicated by rectangles. Standard Pfam domains are indicated by angled boxes. These domains can be identified using only a domain context model, without considering taxonomic context.



```
                                     10                  20                  30
                                      |                   |                   |
OTX1_HUMAN/187-280     A P M H S H H H P H - - Q L S P M A P S S M A G H H H H H P
OTX1_MOUSE/187-280     A P M H S H H H P H - - Q L S P M A P S S M A G H H H H H P
OTX1_RAT/187-280       A P M H S H H H P H - - Q L S P M A P S S M A G H H H H H P
OTX1_BRARE/173-262     A P M H S H H H P H - - Q L S P M T A S S M P T H P H H H I
OTX2_HUMAN/153-235     T P M H H Q L P G P G A T L S P M G T N A V T S H L N Q S P
OTX2_MOUSE/153-235     T P M H H Q L P G P G A T L S P M G T N A V T S H L N Q S P
OTX2_RAT/1-59          T P M H H Q L P G P G A T L S P M G T N A V T S H L N Q S P
OTX2_BRARE/153-235     T P M H H Q L T G P G S T L S P M S S N A V T S H L N Q S P
OTX2_XENLA/153-235     S P M H H Q L S G P G A T L S P M G T N A V T S H L N Q S P
CRX_HUMAN/223-250      S P M V P Q L G G P - - A L S P L S G P S V G P S L A Q S P
```

Figure 2.16: Part of multiple alignment of Tf_Otx domain in members of the Otx1 and Otx2 sub-families. Position 22 in this alignment - corresponds to position 242 on CRX_HUMAN. This position is methionine for all members of the Otx1 subfamily while it is valine for all members of the Otx2 sub-family.

blindness [SCW+97, RBD01]. Recent research demonstrates that a region coinciding with the new prediction (amino acids 200 to 284) is essential for transcriptional activation of the photo-receptor genes, and supports the hypothesis that the V242M mutation acts by impairing this transactivation process [CWX+02]. An analysis of the multiple alignment of the Tf_Otx domains (figure 2.16), demonstrates the existence of two sub-families of the domain, the first of which has a methionine at position 105 and contains all Otx1 proteins, the second of which has a valine at position 105 and contains all Otx2 proteins. Furthermore, the CRX V242M mutation aligns with this position and hence transfers the CRX Tf_Otx domain from the Otx2 to Otx1 sub-family. Finally, note that it has been demonstrated that both Otx2 and CRX transactivate the inter-photo receptor binding protein (IRBP) [BBI+99], while this has not been demonstrated for Otx1. This suggests that the V242M mutation loss of function is due to loss of IRBP transactivation ability, and conversely that position 105 in the Tf_Otx motif is critical for IRBP transactivation.

Figure 2.17 shows further examples of new domain occurrences found by considering taxonomic context only. A pair of TPR repeats are found in Aspartyl (asparaginyl) beta-hydroxylase (Q9Y4J0). This protein has been shown to be over-expressed in an enzymatically active form in hepatocellular carcinoma and cholangiocarcinoma[LJN+96]. The enzyme acts by catalyzing post-translational hydroxylation of β carbons of aspartyl and asparaginyl residues in EGF-like domains with the appropriate consensus sequence. In particular, the Notch homologues – which are known to be involved in cell differentiation and have been shown to be oncogenic – have the appropriate consensus sequence. TPR domains are thought to be involved in protein-protein interactions[DCB98], and may therefore help to mediate this interaction.

The method also identifies novel antistasin domain on the theromin protein (THBI_THETS) in *Theromyzon tessulatum*, a leech. This protein has important medical applications as a potent thrombin inhibitor, and is found in the head of the leech [SCB+00]. The antistasin family is an inhibitor of trypsin-family proteases and is often found in anti-coagulants. Thus the function of the protein concurs with the novel domain occurrence. Taxonomic modelling also find a novel occurrence of the toxin_2, or scorpion short toxin domain on the ErgToxin protein (Q9GQ92) in *Centuroides noxius* (Mexican scorpion). The ErgToxin protein blocks the ERG-K+-channels of nerve, heart and endocrine cells [SBF+00]. Other members of the toxin_2 family also inhibit potassium channels.

Finally, in the fertilization 18kda protein (Q25063) in *Haliotis fulgens* (Green Abalone), a novel Egg_lysin domain is identified. Egg_lysin is found in other *Haliotidae*, as well as other *Archaeogastropoda*. The 18kda fertilization protein acts in conjunction with a paralogous 16kda lysin protein on the egg vitelline envelope. The 16kda protein creates a hole in the vitelline envelope. The 18kda protein is a potent fusagen of liposomes, and is thought to mediate membrane fusion between the gametes, a step in gamete recognition which is important in restricting heterospecific fertilization with other species [SV95]. These authors also found very high divergence amongst the group of orthologous 18kda proteins in California abalone; together with a high frequency of non-synonymous to synonymous substitution, indicating a high selective pressure toward differentiation between species and thus furthering the gamete recognition hypothesis. Furthermore, the 18kda protein exhibits a rate of evolution 2-3x that of the 16kda protein. The 18kda protein in *Haliotis fulgens* is the most distantly related of this group (with 27%–34% identity to the others), and hence standard profile methods fail to detect the similarity.

I had validated the predictions of an earlier version of this method using a Psi-blast [AMS+97] test (table 2.4.2). This test was performed on a set of new domain predictions using Pfam 7.7, and an earlier version of the language modelling software which did not take into account the taxonomic context of sequences. For each novel predicted domain occurrence, Psi-blast was used to generate a set of similar sequence fragments. These sequences were then searched for matches to Pfam families. For 30.7% of novel domain occurrences Psi-blast found matches that are annotated in Pfam. In 90.0% of these the majority of annotations matched the

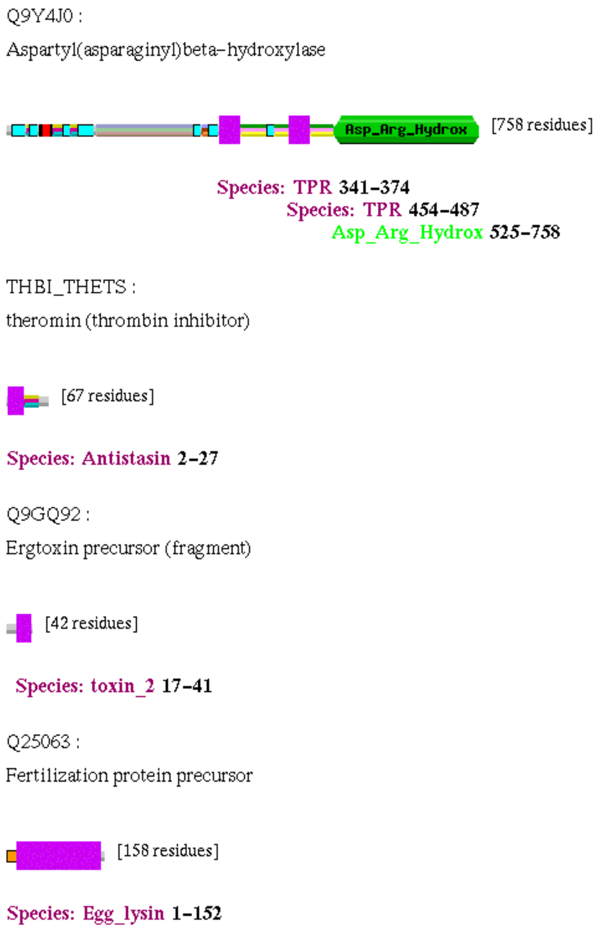


Figure 2.17: Emergence of new domains occurrences, identified using HMMER+taxonomy, indicated by magenta boxes and ‘Species:’ labels. Standard Pfam domains are indicated by angled boxes. These domains can be found by modelling taxonomic context without also considering domain context.

identified family; a further 7.6% had at least one match to the correct family; 0.8% matched a related family and for the remaining 1.5% all matches were to incorrect families. By inspection, the assignment due to the language modelling method of this paper appears to be correct for the overwhelming majority of the 7.6% and 0.8%, and many of the 1.5%. This suggests that the false positive rate is no more than a few percent. Since many of the 69.3% novel predictions for which Psi-blast does not find a match have higher scores than those for which it does, this also indicates the approach can detect matches which Psi-blast does not.

Psi-blast does not find match in Pfam Family		10,575	69.3%
Majority of matches to correct Pfam family		4,220	27.6%
Majority of matches to incorrect family	Has 1 match to correct family	358	2.3%
	Has matches to related family	38	0.3%
	All matches to unrelated families	72	0.5%

Table 2.3: Blast Results For New Positives Predicted By Model.

2.5 Discussion

I have demonstrated that significant improvement in protein domain detection is possible through modelling domain context using techniques inspired by speech recognition methodology. I have shown several examples in which the increased predictive power has discovered domains which further understanding of human disease and biology, and expect there will be many others. From a theoretical point of view, this method provides an integrated prediction of domain annotation for a given protein, evaluating in a strictly probabilistic fashion the appropriate trade-off between amino-acid signal strength and contextual information. Lastly, from a pragmatic perspective, the method significantly increases sequence coverage. The predictions of the method are available via the Pfam web-pages.

Further improvements to the language models are possible, motivated by similar techniques in speech recognition. Modifications to the decision trees used to classify domain contexts are possible, for example I could classify domain contexts on the basis of the longest potentially non-contiguous preceding subsequence which is also observed in the training database. Alternatively, standard classification techniques to learn optimal decision trees can be employed. Other annotated regions on the protein could be used in our search: for example regions of low complexity and transmembrane regions. Explicitly modelling the length distribution of spacers between domains may also increase sensitivity. Lastly, alternative classes of generative grammars may be used – although it remains unclear which level is appropriate for domain modelling. The language modelling could also be adapted to take into account nested domains.

An alternative approach to language modelling, such as the exponential model introduced in section 2.1 might provide more flexibility for modelling long-range domain interactions as well as providing an alternative method for integrating taxonomic information. This method is more computationally expensive but also more flexible with regard to modelling arbitrary features.

Extra information other than taxonomy regarding the protein may also prove a useful guide in domain annotation. For example the techniques used to incorporate taxonomic information can also be used to incorporate protein localisation or even functional information such as phenotype in a systematic RNAi screen.

This type of approach may also be applicable to the discovery of cis-regulatory modules (CRMs) and transcription factor (TF) binding sites. Identification of TF binding sites using weight matrices is difficult, as they can lie kilobases away from the transcription start-site, and the motifs occur at random throughout the genome. Several authors have built organizational models which take motif positioning and orientation into account [DSW01, PFL$^+$01], while others have attempted to identify functional motifs on the basis of high local density of potential binding sites [BNP$^+$02]. Language modelling is related to some of these methods, and may provide an alternative strategy.