# Chapter 1

# Introduction

Furthering our understanding of protein domains is fundamental to our knowledge of both the function of proteins and the evolutionary pressures which have shaped them. Protein domains are the basic structural and functional repertoire from which evolution produces novel function through new combinations as well as modifications within a protein domain. Thus protein domains form an important 'evolutionary crane'[1] [Den95] which have been used during the evolution of complexity.

There has been a recent explosion in the amount of molecular sequence data. As of July 2004, the Integr8 project (`http://www.ebi.ac.uk/integr8/`) contains information from 13 sequenced eukaryotic, 19 archaeal and 150 bacterial genomes. With more genomes in the sequencing pipeline, as well as environmental genome shotgun sequencing [Ven04], the growth of molecular sequence data is set to increase. The number of protein structures is also steadily increasing. Thus, there is increasing amounts of data upon which to build a firmer understanding of protein domain evolution.

Molecular sequence data, used in conjunction with structural data, has already proved to be central in furthering the goal of understanding protein function and evolution. Sequence data has been used for many diverse purposes, including but not limited to:

- detection of evolutionary relationships between proteins;
- clustering proteins into families on the basis of homology;

[1] According to Dennett, an evolutionary crane is a piece of machinery which has itself been generated via evolution, but once created has sped up evolution considerably.

- inferring detailed evolutionary relationships between homologous proteins by reconstructing phylogenetic trees, and describing orthology and paralogy relationships between proteins;
- detecting conserved regions, and inferring potential structural domains;
- inferring branchings in the tree of life;
- detecting highly conserved, functionally important residues;
- detecting correlated sites and inferring interactions;
- detecting pseudogenes;
- detecting positive selection;
- detecting recombination.

Moreover, protein sequence data has led to a better understanding of the evolutionary process both at the molecular level in terms of the mutational and insertion/deletion (indel) processes and at the domain and whole-protein level. Increasingly, with population sequence data becoming available, an understanding of population processes within species is emerging. This includes an understanding of processes responsible for disease and cancer, via studies of somatic mutations and translocations [FCM$^+$04], as well as an understanding of mutational processes which occur in germline cells but are not fixed due to non-viability.

Many of these applications rely on likelihood-based inference and probabilistic models. A probabilistic model parameterises a probability distribution over the space of all possible datasets, and thus assigns a probability (termed likelihood when the data is fixed and we are interested in the probability assigned to it by a particular model) for the particular dataset under investigation. The likelihood-based approach to inference is to construct several probabilistic models, and by comparing the likelihood of the data under one model vs another decide which best explains the data. One example of this approach is phylogenetic tree reconstruction, where each possible tree topology (together with a fixed model of molecular evolution) parameterises a different probabilistic model over alignments. An advantage of likelihood based inference is its flexibility with regard to unknown parameters of the model,

such as branch lengths in the case of tree reconstruction. Unknown parameters can be estimated as those which maximise the likelihood of the data (the maximum likelihood approach). Different models (in this case different tree topologies) can be compared on the basis of the maximum likelihood values. One pitfall of this approach is that adding parameters to a model will never decrease the maximum likelihood value and in most cases increase it, so that models with many parameters have a higher maximum likelihood than those with fewer parameters. A related problem, sometimes called the many parameter trap, occurs when parameters are introduced in relation to the size of the dataset, for instance site-specific parameters for an alignment. The Bayesian approach [Jay03], deals with these problems by comparing models via the integral of the likelihood with respect to a prior probability distribution over parameter space.

In this introductory chapter I will review some of the literature concerning protein domains, introduce mathematical models which are used to identify protein domains, as well as review the literature on parameterising evolutionary models. Accurate protein alignments and trees are fundamental to the approaches taken in this thesis; however, I will not review the vast literature surrounding these two topics, as no new techniques for either alignment or tree reconstruction will be presented in this thesis.

## 1.1 Protein Domains

The strict definition of a protein domain is a distinct structural and evolutionary unit of a protein. This can include the entire protein, in the case of a single domain protein, or it can include a fragment of a multi-domain protein which is observed to also occur in different contexts on different proteins. This definition assumes that the protein domain has a unique stable three dimensional structure, so the definition excludes natively disordered regions, although these regions may still be of functional importance – it has been shown that eukaryotes have 3-5 times more proteins with long regions of no secondary structure than other kingdoms [Ros02]. Not all evolutionary units of proteins have had their structures elucidated, and so in this thesis the definition will be weakened to include evolutionarily conserved protein fragments which are not homologous to an entry in the protein data bank (PDB). A domain architecture is defined as in [VBB$^+$04] as the linear arrangement of protein domains – two proteins have the same architecture if and only if they have the same domain composition,

and the domains are arranged from N-terminal to C-terminal in the same way, with linker regions of up to 30 residues allowed between domains. Inserts can be accommodated in this definition by requiring the same nesting pattern. The term *superfamily* will be used as defined in SCOP [HMBC97], which is as a collection of all domains which are related by descent. *Fold* will also be used as in SCOP to denote the collection of domains which have structural similarity, but may not be evolutionarily related.

A hypothesis forwarded in [LPR01, SL03] is that domains are the result of fusions of *antecedent domain sequences* (ADSs) which are ancient peptide sequences. The authors argue that this hypothesis explains the similarity of motifs present in different protein folds better than the alternative hypothesis of convergent evolution. Under the ADS hypothesis, motifs rather than domains are the atomic units of protein evolution. This hypothesis remains unproven and controversial. It is not considered further in this thesis.

A compelling view to emerge from several structural biology studies is that multi-domain proteins can be described at a high level by their protein domain architectures, and that most proteins are multi-domain (two-thirds in prokaryotes [TPC98], more in eukaryotes [Ger98]). Transferring functional annotation between single-domain proteins from the same superfamily can be done with 68% accuracy, whereas for multi-domain proteins sharing a domain combination the accuracy is 81% and for multi-domain proteins with almost complete residue coverage and identical domain architectures the accuracy rises to 91% [HG01]. These authors also found that ignoring the multi-domain structure of a multi-domain protein is particularly perilous for functional transfer: only 35% of multi-domain protein pairs sharing a single domain have the same function. Aloy and co-workers have shown that the geometry of interaction is generally conserved for homologues with sequence identity above 30% but is not conserved between members of the same fold without evidence of shared ancestry, although they also provide examples of very close homologues not preserving the interaction, and distant homologues which strongly preserve the geometry of interaction [ACSR03]. Thus, it appears that a substantial amount of protein function can be understood via an understanding of the structure and function of representative multi-domain proteins. On the basis of this principle, protein targets have been prioritised for structure determination in structural genomics projects [Bre00, AHT03, VBB$^+$04]. As well as predicting function, domain architectures can be used to assist cellular localization prediction [MSBP02].

Novel proteins are formed during evolution by duplication and recombination. Duplication gives rise to proteins which are freer to diverge and evolve new functions. While this process often leads to the formation of a pseudogene, it is also the main source for the creation of new genes [PP02]. It has been observed that the degree to which different domain superfamilies have been duplicated and subsequently maintained in the genome varies substantially, and this results in a power law distribution of domain superfamily occurrence[QLG01]. Recombination can lead to the formation of novel protein domain architectures, by either fusing genes or by shuffling exons via intronic recombination, leading to domain shuffling [KZNL02]. Insertions of one domain into another account for 9% of non-redundant domain architectures in the PDB – a small but likely significant subset of protein structures [ASHS04].

Apic et al. have demonstrated in [AGT01b, AGT01a, AHT03] that the observed pattern of domain combinations is highly non-random. In fact, a few domain superfamilies are highly versatile in forming multi-domain proteins with a variety of other protein domains, while most have only a single partner. A random model of recombination would predict a much flatter distribution. This suggests that the protein domain combinations which are observed are strongly selected. The authors also showed that multicellular organisms have more sequences and more domain families participating in tandem repeats.

A study of the geometry of domain combinations of Rossman domains [BC02], which are highly versatile in forming multi-domain proteins, has demonstrated that proteins which have the same domain architecture have evolved from the same ancestor. The authors confirmed the observation in [AGT01b] that superfamily combinations almost always occur in the same sequential order, and identified only 2% of cases in which both sequential orders of a domain pair occur. Moreover the authors discovered no structural reason for a particular order, and conclude the observed order is due to the single recombination event which occurred to create the combination. The authors also found extensive conservation of the relative geometry of the domain pair provided the order was conserved, and not otherwise.

Vogel et al. showed in [VBB$^+$04] that some domain combinations occur in many different domain contexts, while preserving the spatial relationships and the linear order of the combination. Such a combination is called a *supra-domain*. Two particular types of supra-domains, were identified based on the geometry of the interaction: *interface* supra-domains have an interface which is critical to the biochemical activity of the protein, whereas the

domains in *separate* supra-domains have biochemically separate but complementary activities. An example provided by the authors of a separate supra-domain is the P-loop nucleotide triphosphate hydrolase domain, which binds and hydrolyses GTP in order to drive a conformational change that is transmitted to its supra-domain partner. In the example provided for the interface supra-domain, both partners of the supra-domain are directly involved in the same cofactor binding interactions. As with domains, some supra-domains have been duplicated substantially, while others have only a few copies. A few supra-domains are very versatile with respect to other domain contexts, while most occur in only a few domain contexts. Vogel et al. note that the majority (64%) of single SCOP domains occur in all three kingdoms of life, whereas most two-domain combinations (96%) and most duplet supra-domains (85%) do not occur in at least one kingdom. Moreover, it was observed in [AGT01b] that of those superfamilies which do participate in kingdom specific domain combinations, significantly more are from all three kingdoms than not. Thus, while domains are in general ancient and common to the last common ancestor of three kingdoms of life, domain combinations have occurred largely within the evolution of specific kingdoms.

Superfamilies often display a wide diversity of function. 25% of CATH superfamilies contain members of different enzyme types [TOT01]. A recent evolutionary study into how evolution generates functional diversity from similar structures demonstrates the economy of nature: structurally conserved residues are kept intact, including residues important for cofactor binding [BBT03]. As noted in [VBK$^+$04], these studies have to a large extent focussed on residue changes within the protein domain structure, and not investigated the effect of domain context in modulating the behaviour of component protein domains. As shown in [HG01] context is of vital importance in correctly annotating domain function. Many examples of context-modulated function have been observed. One example given in [VBK$^+$04] is the winged helix domain, which is typically a DNA binding domain, and in many cases is combined with a regulatory domain, but can also be combined with a catalytic domain so that the protein function, while still acting on DNA, is changed. This is termed a *syntactic* change in [VBK$^+$04]. A more radical modulation of behaviour is observed in cases where the winged helix domain no longer has any DNA binding activity, but instead acts as a substrate specificity pocket, which the authors term a *semantic* change. Elucidating the range and extent of context-dependent domain functional change is an important area for future research

in structural biology.

## 1.2 Sequence based protein domain detection

An important problem is to identify the protein domain architecture of novel protein sequences, for example from genome sequencing projects. Once the domain architecture is determined, it may be possible to transfer functional annotation from biochemically and genetically characterised homologues, as well as to infer structurally important residues as well as regions of interactions with other proteins.

One potential approach is to use pairwise comparison techniques, such as BLAST [AMS+97], and to consider pairwise similarity scores with all members of a domain family. However, methods which use a profile are more sensitive than methods which look for pairwise homology [PKB+98] . A profile summarises the site-specific residue frequencies of a multiple alignment of known members of a domain family, termed the *seed* alignment. The simplest profile method is the position specific scoring matrix (PSSM) which constructs a probability distribution at each of the $m$ sites in the alignment and does not allow gaps. A novel sequence is scanned by the PSSM by calculating at each site the probability that the next $m$ residues in the sequence have been emitted by the corresponding distributions in the PSSM, and the highest score is taken as the overall score.

Profile hidden Markov models (HMMs) formalise PSSMs as probabilistic models and improve its sensitivity by allowing insertions and deletions relative to the profile. A profile HMM (labelled $\mathbf{D}$) is a probabilistic model which parameterises a probability distribution over all possible sequences (labelled $x = x_1 x_2 \dots x_n$). The basic idea is that the profile HMM constructed for a domain family assigns high probability to sequences which are homologous to the domain family (or more strictly contain a homologous fragment), and a low probability to non-homologous sequences. One problem with using the probability of the sequence as a score, regardless of the precise details of the profile HMM, is that long sequences (above a certain length threshold) will inevitably have lower probability than shorter sequences. From the point of view of Bayesian inference another problem is that the correct probability upon which to base the inference is the posterior probability

$$P(\mathbf{D}|x) = \frac{P(x|\mathbf{D})P(\mathbf{D})}{\sum_{\mathbf{D}'} P(x|\mathbf{D}')P(\mathbf{D}')}, \tag{1.1}$$

where the denominator is a sum of the likelihood under all possible domain models $\mathbf{D}'$ multiplied by the prior probability of that model, which is expensive to calculate. Both problems can be overcome by introducing as an alternative hypothesis a background probability distribution $\mathbf{R}$ over sequence space, and calculating the ratio of posterior probabilities, in which case the term involving the sum in the previous equation cancels out. It will be convenient to work with log probabilities

$$\log \frac{P(\mathbf{D}|x)}{P(\mathbf{R}|x)} = \log \frac{P(x|\mathbf{D})}{P(x|\mathbf{R})} + \log \frac{P(\mathbf{D})}{P(\mathbf{R})} \tag{1.2}$$

where the first term is called the log-odds score, and the second term is the log ratio of the prior probabilities of the models, and can be thought of as a threshold on the log-odds score. As long as the background model has a similar distribution over protein lengths, the scores should be normalised with respect to protein length.

The log-odds score is used to rank sequences and apply a threshold cut-off such that all sequences scoring above a threshold are taken to be members of the family. It is a useful measure for inferring relative similarities of sequences to the protein domain, but does not provide similarity scores in absolute terms, or at a particular level of significance. Empirical significance values can be obtained by calculating the log-odds scores for sequences randomly sampled from the background model. In this way a distribution of scores for random 'proteins' is obtained, and the significance level of a sequence log-odds score can be obtained by counting the fraction of random sequences which score higher. However, to get an accurate significance value for high scoring sequences in this way many hundreds of thousands of random sequences need to be scored[2]. To get around this problem it has been observed that the distribution of random scores from an profile HMM follows an extreme value distribution, which can be successfully parameterised with much less data (HMMER uses 5000 sequences)[DEKM98].

It is useful at this point to parameterise the profile HMM. I start with a description of Markov models, which will be useful at other points in this thesis.

### Discrete Markov Models

Let $\Sigma$ denote a state space, $\{Y_i : [0,1] \to \Sigma\}_{i=1,2\ldots}$ denote a series of random variables each of which takes values in the state space $\Sigma$. Let $\mu$ denote the uniform probability distribution

[2]particularly if the significance value is to be ascertained to the level required for annotation in Pfam, which is a significance of less than 1/N where N is the number of protein sequences scored, currently around 1.5m.

on the interval $[0,1]$ so that

$$P(Y_i = y) = \mu(\{r \in [0,1] : Y_i(r) = y\})$$

defines a probability distribution on $\Sigma$.

A $k^{th}$ order Markov model is a probabilistic model with the property that the state at position $i$ is only dependent on the preceding $k$ states:

$$P(Y_i = y|Y_{i-1} = y_1, \ldots, Y_1 = y_{i-1}) = P(Y_i = y|Y_{i-1} = y_1, \ldots, Y_{i-k} = y_k). \quad (1.3)$$

In the simplest cases of a homogeneous Markov model these probabilities are independent of position in the chain

$$P(Y_i = y|Y_{i-1} = y_1, \ldots Y_{i-k} = y_k) = P(Y_{i'} = y|Y_{i'-1} = y_1, \ldots Y_{i'-k} = y_k),$$
$$\forall 1 \leq i, i' \leq n, \text{ and } y_1 \ldots y_k \in \Sigma. \quad (1.4)$$

So all that is needed to specify a Markov model is to specify the states, and the transition probabilities $P(y|y_1, \ldots y_k)$ between states. In the case of a first order Markov model, I will also write $P(y_1 \rightarrow y)$ for the transition probability. It will be useful to include a special state $S$ in which the model starts and one for which it terminates, $T$. The probability distribution parameterised by a Markov model is over chains of states, which will be of finite but unbounded length provided there is a path with non-zero transition probability from every state in the model to the end state.

A hidden Markov model is a Markov model in which some states $y \in \Sigma$ themselves are allowed to be random variables, $y : [0,1] \rightarrow \Upsilon$, taking values in the space $\Upsilon$. These are termed emission states. It is also useful to allow states which are not random variables, which includes the start and terminate states. The emission probability distribution for emission state $y$ over $\Upsilon$ is then defined as

$$P(u|y) = \mu(\{r \in [0,1] : y(r) = u\}).$$

In the case of a profile HMM, the state space $\Upsilon$ will be amino-acids, codons or nucleotides. These states are hidden in the sense that they are not observed in the data, but are internal states of the overall probabilistic model. They are introduced in order to provide flexibility in parameterising an appropriate probability distribution over sequence space. So, in order to

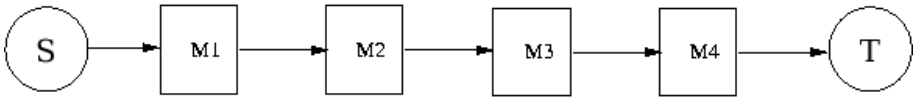


Figure 1.1: Basic architecture for profile hidden Markov model for an alignment with four amino acids and without gaps. Each $M_i$ corresponds to a column in the multiple alignment, and emits over a distribution of amino acids. $B, E$ correspond to begin and end states.

specify a Hidden Markov model, the hidden states $\Sigma$, transition probabilities $P(y|y_1, \ldots y_k)$ and emission probabilities $P(u|y)$ must be specified.

The PSSM reframed in this framework is shown in figure 1.1. The state space $\Sigma$ consists of match states $M_j$ for each column in the seed alignment as well as the begin and end states. Each match state emits in the space $\Upsilon$ of amino-acids.

A profile HMM from HMMER is shown in figure 1.2, taken from [Edd03]. In fact this comprises two HMMs – the domain model HMM and the null model HMM. Both HMMs are first order HMMs. For an alignment consisting of $m$ conserved columns (which can be defined as columns with less than 50% gaps) the domain model state space $\Sigma$ includes $m$ match and insert emission states $M_j$ and $I_j$ as well as $m$ non-emission states $D_j$. $\Sigma$ also includes an N-terminal, C-terminal and inter-domain emission state $N, C, J$ respectively as well as domain begin and end non-emission states $B, E$. The $M_j$ emit residues according to a probability distribution estimated from the counts observed in a particular conserved column of the alignment. The insert states $I_j$ emission probability is calculated from all insert states in the Pfam database. The match to insert transitions specify the 'cost' of opening a gap relative to the protein domain, and the insert to insert transitions specify the cost to maintain the gap, which is an affine gap scoring scheme. The delete states allow for domain states to be skipped, with a penalty controlled by match to delete and delete to match transitions. The $N, C, J$ states allow the model to score full length proteins by allowing for N- C-terminal and inter domain regions respectively. These states emit according to a background model of residue usage in proteins. The null model HMM consists of a single emission state, which emits according to the background distribution of protein residues.

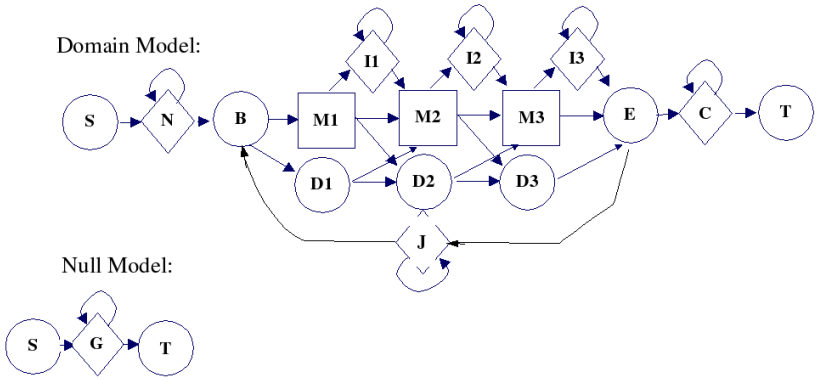


Figure 1.2: Diagram of profile Hidden Markov Model. States which emit symbols are shown as squares or diamonds; circles do not emit symbols. The core model consists of match states – which model conserved residues of a protein family; insert states – which allow for segments of the query sequence not present in the protein family; and delete states – which allow for deletions of conserved residues in the protein family from the query sequence. The model consists of several flanking states, which allow for local matches and multiple hits. The transition to the J state allows for multiple hits of the model to a single query sequence. The N, J and C states are analogous to insert states, but occur before, between and after the model hit respectively. The B and T states are states used to begin and terminate a hit to the query, while S and E states are formally required as overall start and end states. To obtain the log-odds score we also require a null model. The null model consists of an null emission state $G$ which emits according to a background distribution, and can loop back to itself, or transition to the end state. Effectively the transitions of the null model act to negate the otherwise intrinsic penalty for scoring longer query sequences.

**Calculating the likelihood of a profile HMM**

The forward algorithm can be used to calculate the likelihood of the sequence given the profile HMM. Note that the profile HMM can generate a particular sequence with many different paths through the HMM architecture, although only a few will have high posterior probability[DEKM98]. The forward algorithm naturally sums over all possible paths, in contrast to the Viterbi algorithm (which will not be used, and so not described in more detail, but see [DEKM98]) which calculates the probability of the mostly likely path to have generated the sequence. The Viterbi algorithm is employed by HMMER, and has the advantage that all calculations can be done in log probability space and that only summation is required. The forward algorithm, on the other hand requires working in probability space with multiplication, which can lead to underflow errors if an adaptive scaling algorithm is not employed.

The forward algorithm proceeds by iteratively filling in eight matrices $P(x_1 \dots x_i|S \dots M_j)$, $P(x_1 \dots x_i|S \dots I_j)$, $P(x_1 \dots x_i|S \dots D_j)$, $P(x_1 \dots x_i|S \dots C)$, $P(x_1 \dots x_i|S \dots J)$, $P(x_1 \dots x_i|S \dots N)$, $P(x_1 \dots x_i|S \dots B)$, $P(x_1 \dots x_i|S \dots E)$ which are the partial probabilities of the HMM emitting subsequence up to and including the $i^{th}$ residue and the ending in the $j^{th}$ match, delete, insert or the $C, J, N, B, E$ states respectively. Let $\psi_i$ denote the state which emitted residue $x_i$. If the domain begin state $B$ is interpreted as also being $M_0$, these scores can be calculated

recursively using

$$
\begin{aligned}
P(x_1 \dots x_i | S \dots M_j) &= P(x_i|\psi_i = M_j) \cdot \begin{pmatrix} P(x_1 \dots x_{i-1}|S \dots M_{j-1}) \cdot P(M_{j-1} \to M_j) \\ + P(x_1 \dots x_{i-1}|S \dots I_{j-1}) \cdot P(I_{j-1} \to M_j) \\ + P(x_1 \dots x_i|S \dots D_{j-1}) \cdot P(D_{j-1} \to M_j) \end{pmatrix} \\
P(x_1 \dots x_i | S \dots D_j) &= \begin{pmatrix} P(x_1 \dots x_i|S \dots M_{j-1}) \cdot P(M_{j-1} \to D_j) \\ + P(x_1 \dots x_i|S \dots D_{j-1}) \cdot P(D_{j-1} \to D_j) \end{pmatrix} \\
P(x_1 \dots x_i | S \dots I_j) &= P(x_i|\psi_i = I_j) \cdot \begin{pmatrix} P(x_1 \dots x_{i-1}|S \dots M_j) \cdot P(M_j \to I_j) \\ + P(x_1 \dots x_{i-1}|S \dots I_j) \cdot P(I_j \to I_j) \end{pmatrix} \\
P(x_1 \dots x_i | S \dots B) &= \begin{pmatrix} P(x_1 \dots x_{i-1}|S \dots C) \cdot P(C \to B) \\ + P(x_1 \dots x_{i-1}|S \dots J) \cdot P(J \to B) \end{pmatrix} \\
P(x_1 \dots x_i | S \dots E) &= \begin{pmatrix} P(x_1 \dots x_{i-1}|S \dots M_m) \cdot P(M_m \to E) \\ + P(x_1 \dots x_{i-1}|S \dots I_m) \cdot P(I_m \to E) \\ + P(x_1 \dots x_i|S \dots D_m) \cdot P(D_m \to E) \end{pmatrix}
\end{aligned}
\tag{1.5}
$$

$$
\begin{aligned}
P(x_1 \dots x_i | S \dots C) &= P(x_i|\psi_i = C) \cdot \begin{pmatrix} P(S \to C) \text{ if } i = 1 \\ P(x_1 \dots x_{i-1}|S \dots C) \cdot P(C \to C) \end{pmatrix} \\
P(x_1 \dots x_i | S \dots J) &= P(x_i|\psi_i = J) \cdot \begin{pmatrix} P(x_1 \dots x_{i-1}|S \dots E) \cdot P(E \to J) \\ + P(x_1 \dots x_{i-1}|S \dots J) \cdot P(J \to J) \end{pmatrix} \\
P(x_1 \dots x_i | S \dots N) &= P(x_i|\psi_i = N) \cdot \begin{pmatrix} P(x_1 \dots x_{i-1}|S \dots N) \cdot P(N \to N) \\ + P(x_1 \dots x_{i-1}|S \dots E) \cdot P(E \to N) \end{pmatrix}
\end{aligned}
\tag{1.6}
$$

The overall likelihood of the domain matching the sequence is equal to the score for the terminal state: $P(x|\mathbf{D}) = P(x_1 \dots x_n|S \dots C)P(C \to T)$. The probability of the sequence being emitted by the null model can be calculated as

$$P(x|\mathbf{R}) = P(G \to G)^n \cdot P(G \to T) \cdot \prod_i P(x_i|G).$$

The overall log-odds score is then calculated as $\log P(x|\mathbf{D}) - \log P(x|\mathbf{R})$.

It is also possible to calculate the backward partial scores $P(x_{i+1} \dots x_n|M_j \dots T)$, $P(x_{i+1} \dots x_n|I_j \dots T)$, $P(x_{i+1} \dots x_n|D_j \dots T)$, $P(x_{i+1} \dots x_n|C \dots T)$, $P(x_{i+1} \dots x_n|J \dots T)$,

$P(x_{i+1}\dots x_n|N\dots T)$, $P(x_{i+1}\dots x_n|B\dots T)$, $P(x_{i+1}\dots x_n|E\dots T)$ which are the partial probability of the HMM emitting the subsequence from residue $i+1$ to $n$ and coming from $j^{th}$ match, delete, insert or the $C, J, N, B, E$ states respectively. The backward algorithm proceeds iteratively from the C-terminal to N-terminal end of the sequence (equations only shown for $P(x_{i+1}\dots x_n|M_j\dots T)$). Again to simplify the equations, the $E$ state is interpreted to be the same state as $M_{m+1}$ (where $m$ is the number of match states).

$$P(x_{i+1}\dots x_n|M_j\dots T) = \begin{pmatrix} P(x_{i+1}|\psi_{i+1}=M_{j+1})\cdot P(x_{i+2}\dots x_n|M_{j+1}\dots T)\cdot P(M_j\to M_{j+1}) \\ +P(x_{i+1}|\psi_{i+1}=I_j)\cdot P(x_{i+2}\dots x_n|I_j\dots T)\cdot P(M_j\to I_j) \\ +P(x_{i+1}\dots x_n|D_{j+1}\dots T)\cdot P(M_j\to D_{j+1}) \end{pmatrix} \tag{1.7}$$

Using the definition of the partial forward and backward scores

$$P(x|\psi_i=M_j) = P(x_1\dots x_i|S\dots M_j)P(x_{i+1}\dots x_n|M_j\dots T)$$

and hence, using Bayes' theorem

$$P(\psi_i=M_j|x) = \frac{P(x_1\dots x_i|S\dots M_j)\cdot P(x_{i+1}\dots x_n|M_j\dots T)}{P(x)} \tag{1.8}$$

This provides a way of calculating the posterior probabilities.

**Building the profile HMM**

A profile HMM is constructed from a seed alignment of homologous sequences. The conserved columns in the alignment correspond to match states, and the other columns correspond to insert states. Which columns to label as match states and which to label as insert states can be either resolved heuristically (by labelling all columns with greater than 50% gaps as insert states), or using the maximum *a posteriori* architecture algorithm to find the profile HMM which optimises the likelihood of the seed alignment (see [DEKM98]).

The observed residues in a conserved column are used to estimate the emission probability distribution for that column, and the observed transitions are used to estimate the transition probabilities. The most straightforward approach is to use the observed frequencies as the emission and transition probabilities, however this assumes that the rows are independently sampled from the target probability distribution, which is not true. Some sequences

are closely related to each other whereas some are more distantly related. To get around this problem, various weighting schemes have been proposed [SA90, HH92, GSC94, KM95, HH96], which all in effect try to adjust the weights of the rows included in the counts so that sequences from parts of sequence space which are not well sampled in the seed contribute more, and sequences from parts of sequence space which are well sampled each contribute less to the overall estimation of emission and transition probabilities. Even after re-weighting the sequences to remove sampling biases, another problem is that the space of possible domain family members has been inadequately sampled, particularly for small seed alignments. This problem has been addressed using a mixture of Dirichlet priors, in which the emission probability is taken to be the mixture of $k$ posterior probabilities, each of which are calculated as the posterior probability of the residue frequencies given the column and one of $k$ Dirichlet priors. The mixture co-efficients are calculated as the posterior probability of each mixture component given the observed counts in the column. See [SKB$^+$96] for more details. This problem can also be addressed using the tree HMM introduced in the next section.

#### Other methods and extensions

One improvement made to profile HMM methods in recent years has been the introduction of iterations, whereby an initial sequence is used to build a profile HMM which is searched against a database and significant hits are used to rebuild the profile HMM. This process is then repeated until no further hits are found. This is the strategy used in SAM [HK96] and PSI-blast [AMS$^+$97].

Several profile-profile comparison techniques have been proposed in recent years [Pie96, YL02, SBG03, Sd04]. The motivation for these methods is that profile-sequence comparisons are more sensitive than sequence-sequence comparisons and so profile-profile comparisons might be expected to be even more sensitive in detecting weak homology. Indeed, these methods appear to be more sensitive than profile HMMs. The method proposed in [Pie96] was developed for the comparison or conserved ungapped alignments from the BLOCKS [HHP99] database and so does not allow gaps. PROF_SIM [YL02] and COMPASS [SBG03] both use allow gaps via a Smith-Waterman local alignment algorithm with column similarity scores based on Jensen-Shannon entropy and a symmetric log-odds ratios respectively. Söding [Sd04] generalizes the profile HMM framework to compare two profiles.

Several discriminative support vector machine (SVM) approaches have been applied to homology detection. Jaakkola et al. [JDH00] proposed a method for using profile HMMs to derive a kernel function in a SVM classifier. The motivation for this approach is that profile HMMs are trained using positive training examples only. A discriminative model, which takes both positive and negative training examples, should perform better. A support vector machine is a discriminative model which can be thought of as a classifier which can be trained to discriminate points in a high dimensional space. A support vector machine relies on a kernel function $K(x, x^k)$ which can be thought of as a measure of similarity between a sequence $x$ and training example $x^k$, which can be either positive or negative. Considering the profile HMM as a likelihood function over sequence space, Jaakkola et al. define a vector $U_x$ called a Fisher score, which is the partial derivative of the log-likelihood score at the sequence $X$ with respect each of the parameters of the profile HMM. The vectors $U_x$ and $U_{x^k}$ are then used to derive the kernel function via a formula presented in [JDH00]. Leslie [LEC+04] et al. have proposed a string kernel for protein classification which maps a protein to a vector $U_x$ called its 'k-spectrum' which is the set of all k-mers contained in the protein. The kernel function is then a vector function of $U_x$ and $U_{x^k}$ as before.

## 1.3 Models of sequence evolution

The genomic sequence of cellular organisms is in constant flux. During cell division replication introduces copying errors of which some fraction remain uncorrected. Recombination leads to exchange of genomic material between alleles in the case of eukaryotes, and between different species in bacteria. Processes such as non-allelic homologous recombination lead to genomic rearrangements including deletion, inversion, translocation and duplication. Retroviral elements are integrated into genomic DNA. Certain proteins promote genomic mutation via processes such as class switch recombination and somatic hypermutation, particularly in certain cell types such as germinal centre B cells where mutation is required in order to generate a diverse set of antibodies. External factors such as radiation also lead to genomic mutation. These mutations can occur either in somatic cells, in which case they are not passed to the next generation, or in germline cells. Most germline cell mutations are hypothesised to be neutral [Kim83], however some will be deleterious and therefore not survive. Rarely, mutations will be advantageous and selected for, resulting in a selective sweep through the

population.

The DNA or RNA sequence of viruses is typically under an even higher rate of flux than for cellular organisms due to copying errors during replication in the host cell as well as processes such as host-mediated hypermutation. In many cases these errors are not repaired by the host cell DNA repair machinery and so the rate at which mutation occurs is significantly higher. Retrotransposition is a particularly error prone step leading to high rates of mutation in retroviruses. Viruses often have particular features which enhance the rate of mutation.

Thus genomic sequences change over time and these changes can be modelled at different levels: within a single cell during the cell's lifetime; progressively during transmission from parent to daughter cells; within a population of cells (for example during the progression of a tumour, or a bacterial culture); transmission from a multicellular parent to offspring organism; within a population of multi-cellular organisms; or between different species of organisms. Each of these levels requires a different level of resolution. For instance when modelling difference between species, differences within a population will typically be ignored and the most frequent allele will be taken as representative for that species. Due to duplication, different segments of genomic DNA will be related to each other via descent, and so sequence evolution can be modelled within a single genome.

### 1.3.1 Probabilistic models of sequence evolution

Let $\Upsilon$ describe a state-space, which initially is taken to be all of sequence space, and let $u, v \in \Upsilon$ be elements of this state space. Let $|u|$ denote the length of a sequence. A probabilistic model of sequence evolution, denoted by $\mathcal{E}$, is a model which describes a probability distribution $P_{\mathcal{E}}(x^t = u)$ over sequences at each time $t \geq 0$. This can be used to describe the transition probability $P_{\mathcal{E}}(x^{t+\triangle t} = v | x^t = u)$ of observing a sequence $v$ at time $t + \triangle t$ given that $u$ was observed at time $t$. A general probabilistic model of evolution would need take into account all of the mutational processes described above, including point mutation, insertion, deletion, recombination, gene conversion and translocation. Moreover, such a model would also need to describe how the rates of each of these processes change with respect to position in the genome and time. This is clearly a very challenging task.

Given the complexity of the task, why bother constructing probabilistic models of evolution? The answer principally lies in the usefulness of the likelihood $P(\{x^k\}|\mathcal{E}, T)$ of a cluster

of homologous sequences $\{x^k\}$. If $\mathcal{E}$ is fixed, the likelihood can be used as a criterion for evaluating how well the tree fits the data, for finding optimal branch lengths, and for parameterising a posterior distribution over all possible trees. This approach can also be used to find evidence for recombination [HW01]. If $T$ is fixed, the likelihood can be used to compare different evolutionary models, and so gain quantifiable insight into the evolutionary process itself. This approach is the basis for tests for pseudogenes and positive selection, which will be further described in section 4.1 and 4.2. Probabilistic models of evolution can also be used to align sequences [MD95, HB01, Hol03, MLH04].

**Whole sequence evolutionary models**

The first standard simplifying assumption is that $\mathcal{E}$ is a continuous-time Markov process over the state space $\Upsilon$. This corresponds to assuming that the transition probability, $P_{\mathcal{E}}(x^{t+\triangle t} = u|x^t = v)$ is independent of $t$, or that evolution is homogeneous with respect to time. This simplifying assumption is clearly violated in many circumstances. One example is if a functional gene becomes non-functional, in which case the evolutionary constraints on the sequence change. One way to improve the realism of models with respect to this assumption is to have different models on different parts of the tree, as in Chapter 4. If the process $\mathcal{E}$ is assumed to be Markov, then the time evolution of the probability distributions $P_{\mathcal{E}}(t)$ is described by the differential equation

$$\frac{dP_{\mathcal{E}}(t)}{dt} = P_{\mathcal{E}}(t)\mathbf{Q}r \tag{1.9}$$

where $\mathbf{Q}$ is a fixed rate matrix describing the instantaneous transition rate between states in the state space so that $\mathbf{Q}_{u,v}$ the instantaneous rate of transition between states $u$ and $v$, and satisfies

$$Q_{u,u} = -\sum_{v\in\Upsilon, v\neq u} Q_{u,v},\ \forall u \in \Upsilon. \tag{1.10}$$

An arbitrary scaling constant $r$ representing the rate of evolution has been included for future reference and can be assumed at this stage to be equal 1. This rate matrix is scaled so that the average rate of substitution at equilibrium is 1:

$$-\sum_{u\in\Upsilon} \pi_u \mathbf{Q}_{u,u} = 1 \tag{1.11}$$

which reduces by 1 the number of parameters required to specify a rate matrix and implies that $rt$ is measured in units of expected substitutions per site. The solution to the differential

equation is given by

$$P_{\mathcal{E}}(t) = P_{\mathcal{E}}(0)e^{\mathbf{Q}rt}, \tag{1.12}$$

where $e^{\mathbf{Q}rt}$ is the matrix exponential. The transition probabilities are given by

$$P_{\mathcal{E}}(x^{t+\triangle t} = v | x^t = u) = \left[e^{\mathbf{Q}r\triangle t}\right]_{u,v}. \tag{1.13}$$

The rate matrix $\mathbf{Q}$ is assumed to be irreducible, which requires that the there is a non-zero probability of transitioning over some time $\triangle t > 0$ between any two states $u$ and $v$, and recurrent, which requires that the probability of visiting each state at least $N$ times in an infinite amount of time is equal to 1 for all positive integers $N$. A stationary distribution $\pi$ of $\mathbf{Q}$ is a distribution for which $\pi\mathbf{Q} = 0$. For a recurrent, irreducible rate matrix $\mathbf{Q}$ a stationary distribution $\pi$ exists and is unique up to scalar multiplication (see [Nor97] for further details). Thus it makes sense to talk about the stationary probability distribution of $\mathcal{E}$, and so I will write $\mathcal{E} = (\pi, \mathbf{Q})$. Another common simplifying assumption is reversibility, which implies that the instantaneous flux between residues is the same in both directions

$$\pi_u \mathbf{Q}_{u,v} = \pi_v \mathbf{Q}_{v,u}. \tag{1.14}$$

This halves the number of parameters required to estimate the rate matrix $\mathbf{Q}$. There is no a priori reason to expect that evolution is reversible, although there is some evidence [AB97] that DNA evolution in many cases is close to reversible. As observed in [HD98], insertion events may be short and frequent, while deletion events long and rare, which would lead to a violation of the reversibility assumption. Observe that if eq. 1.14 holds then

$$\mathbf{S}_{u,v}(f) = \pi_u^f \pi_v^{f-1} \mathbf{Q}_{u,v} \tag{1.15}$$

is symmetric, i.e. $\mathbf{S}_{u,v}(f) = \mathbf{S}_{v,u}(f)$. This defines a single parameter family of symmetric matrices for $\mathbf{Q}$. The parameter $f$, described in [GW02], is called the +gwF parameter. $\mathbf{S}(0)$ is referred to as an exchangeability matrix. A symmetric matrix can be expressed in the form

$$\mathbf{S}(f) = \mathbf{N}(f)\mathbf{D}(f)\mathbf{N}(f)^T \tag{1.16}$$

where $\mathbf{D}(f)$ is a diagonal matrix, $\mathbf{N}(f)$ is an orthonormal matrix and $\mathbf{N}(f)^T$ is the matrix transpose (see [Lay94] for further details). Let $\Pi$ be a diagonal matrix with entries $\Pi_{u,u} = \pi_u$.

If I restrict to $f = 1/2$

$$\mathbf{Q} = \Pi^{-1/2}\mathbf{S}(1/2)\Pi^{1/2} \tag{1.17}$$

$$= \mathbf{N}'(1/2)\mathbf{D}(1/2)\mathbf{N}'(1/2)^{-1} \tag{1.18}$$

where

$$\mathbf{N}'(1/2) = \Pi^{-1/2}\mathbf{N}(1/2). \tag{1.19}$$

Thus, $\mathbf{Q}$ is diagonalizable and the matrix exponential can be calculated as

$$e^{\mathbf{Q}r\triangle t} = \Pi^{-1/2}\mathbf{N}(1/2)e^{\mathbf{D}(1/2)r\triangle t}\mathbf{N}(1/2)^T\Pi^{1/2} \tag{1.20}$$

which provides a fast way to calculate the matrix exponential – first calculate the orthonormal decomposition of $\mathbf{S}(1/2)$ and then for all $t > 0$ the matrix exponential step just consists of exponentiating the diagonal entries of $\mathbf{D}(1/2)$ and two matrix multiplication steps. The columns of $\mathbf{N}'(1/2)$ are the eigenvectors of $\mathbf{Q}$ and can be interpreted as directions in state space in which information about the ancestral sequence is lost through evolution. The corresponding diagonal entries are the rate at which the information is lost.

Most methods also assume stationarity, which says that $P_{\mathcal{E}}(0) = \pi$, or that the system is at equilibrium at time 0. There is also no particular reason to expect stationarity to hold in general. In particular, a universal trend of amino acid loss and gain has been observed in all kingdoms of life [JKA$^+$], with Cys, Met, His, Ser, and Phe gaining and Pro, Ala, Glu, and Gly losing frequency. Moreover G+C content varies widely between genomes, again indicating the stationarity does not hold in general.

For proteins of known structure, Robinson et al. [RJK$^+$03] parameterise a whole sequence model for sequences evolving in such a way as to preserve this structure. The authors restrict their evolutionary model to DNA sequences of length $N$ and allow only one position in the sequence change in any given mutation event, so the rate matrix $\mathbf{Q}$ is of size $4^N$ x $4^N$ and each row has no more than $3N$ non-zero off-diagonal entries. The rate of amino-acid changing substitutions is based on the propensity of the mutation to change the structure using a sequence-structure compatibility score. Transition/transversion and non-synonymous/synonymous substitution rate ratios are used to determine the underlying DNA mutation rate within these constraints.

**Substitution, insertion, deletion models**

Most models of sequence evolution further assume that the evolution consists of two independent processes, namely a k-mer residue substitution process and an insertion/deletion (indel) process. The k-mer residue substitution process can itself be considered as a continuous time Markov process. Let $\dot{\Upsilon}$ and $\dot{\mathcal{E}} = (\dot{\mathbf{Q}}, \dot{\pi})$ denote the state space and substitution model respectively for a single residue substitution process, with the natural extension for 2-mer and 3-mer substitution processes. The symbols $u$, $v$ will be used to represent both arbitrary length sequences as well as single residues, but it will be clear from the context which is implied in each case.

Miklós et al. consider the class of evolutionary models which allow local point substitutions and multiple residue inserts and deletes (called SID models) [MLH04]. Let $\rho_I(u)$ be the context-independent rate of insertion of sequence $u$ between two residues in an ancestral sequence and let $\rho_D(u)$ be the context-independent rate of deletion of sequence $u$.

The simplest SID model disallows insertions and deletions in the evolutionary model, and treats gaps as either missing data (see section 1.3.3 for a discussion on how to accommodate missing data in the likelihood calculation), or as an extra residue character. This has the effect of not allowing the sequence length to change over time. The residue substitution process can be further simplified by assuming that sites evolve independently of one another according to a single residue model $\dot{\mathcal{E}}$. Site-specific residue models are discussed in more depth below.

The TFK91 links model [TKF91] is a SID model with an arbitrary point substitution matrix $(\dot{\mathbf{Q}}, \dot{\pi})$ and an indel process governed by

$$\rho_I(u) = \begin{cases} \lambda \dot{\pi}_{u_1} \text{ if } |u| = 1 \\ 0 \text{ otherwise} \end{cases} \tag{1.21}$$

$$\rho_D(u) = \begin{cases} \mu \text{ if } |u| = 1 \\ 0 \text{ otherwise,} \end{cases} \tag{1.22}$$

where $\lambda$ is the insert rate and $\mu$ is the deletion rate. The TFK92 model [TKF92] is an extension to this model but considers a sequence to consist of fixed-length indivisible fragments of variable length $k$. The substitution process for each of these k-mers is given by $\overset{k}{\mathcal{E}} = (\overset{k}{\mathbf{Q}}, \overset{k}{\pi})$

and the indel process is the same as for TFK91 but considered over arbitrary length k-mers. Each of the $\overset{k}{\mathcal{E}}$ can be parameterised as a product of $k$ independent single residue models.

Miklós et al. introduce a long-indel model [MLH04] which is parameterised over the state space $\Upsilon$ of sequences of arbitrary length by

$$\rho_I(u) = \lambda_{|u|} \prod_{k=1}^{|u|} \dot{\pi}_{u_k} \tag{1.23}$$

$$\rho_D(u) = \mu_{|u|} \tag{1.24}$$

where $\lambda_{|u|}$ and $\mu_{|u|}$ are the rate of deletion and insertion respectively of sequences of length $|u|$. Miklós et al. derive restrictions on the insertion and deletion rates in order to preserve reversibility. Alignment algorithms using this model are also presented.

Mitchison and Durbin [MD95] propose a tree HMM to model insertions and deletions. Under this model, there is a collection of $n$ match $\{M_j\}$ and $n$ delete $\{D_j\}$ states of a HMM, each of which will generate a column in a multiple sequence alignment with $n$ columns. The model does not allow insertions, so the maximum length $n$ of sequence generated by this model is pre-specified. The HMM architecture is shown in figure 1.3. The path through the HMM is evolved as well as the residues. Thus, the sequence $x^t$ is augmented with $\psi^t$ describing the path through the model at time $t$. Mitchison and Durbin propose that each transition in the model evolves independently according to continuous, stationary, time-reversible Markov process, denoted $\bar{\mathcal{E}} = (\bar{\mathbf{Q}}, \bar{\pi})$, over the state space of transitions, denoted by $\bar{\Upsilon}$, where

$$\bar{\Upsilon} = \{M_i \to M_{i+1}, M_i \to D_{i+1}, D_i \to D_{i+1}, D_i \to M_{i+1}\}, \tag{1.25}$$

provided $M_0$ is interpreted as the begin state and $M_{n+1}$ is interpreted as the end state. The rate matrix over transitions $\bar{\mathbf{Q}}$ is trained from a database of alignments. Note that the path $\psi^t$ is hidden, and so to use the tree HMM for inference of trees and evolutionary distances, it is necessary to sum over all possible hidden states, which is computationally expensive. The tree HMM does not attempt to model novel insertions – instead it models 're-insertion' of ancestral sequence which has been temporarily lost in a lineage. For practical purposes this is not a substantial drawback but it is unsatisfactory from a theoretical point of view.

### 1.3.2 Models of residue substitution

The most general non-reversible DNA model is the unrestricted model (UNR), which has 11 free parameters (12 off-diagonal elements minus 1 parameter for scaling). The general time

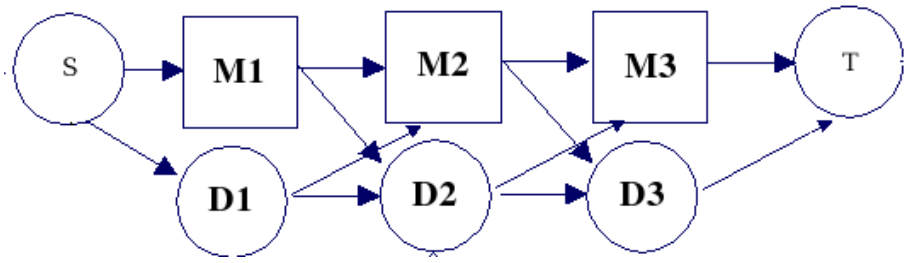


Figure 1.3: Basic architecture for tree HMM as defined in [MD95]. Each $M_i$ corresponds to a column in the multiple alignment, and emits over a distribution of amino acids. $S, T$ correspond to start and termination states.

reversible model (GTR) has 5 free parameters (6 upper diagonal elements minus 1 parameter for scaling). Other substitution models include the HKY model [HKY85], which has 4 free parameters and the F81 model [Fel81], which has 3 free parameters. These parameters are typically trained from an alignment with a given tree topology by jointly finding the parameters of the model and the branch lengths which maximise the likelihood of the data. Methods have been devised for simultaneously optimising over alternative tree topologies.

Goldman and Yang [GY94] and Muse and Gaut [MG94] have described models of codon evolution which take into account an underlying nucleotide model based on the HKY [HKY85] with transition/transversion ratio of $\kappa$ as well as a non-synonymous/synonymous rate $\omega$. These models were refined in [YN98]. Let $u = u_1u_2u_3$ and $v = v_1v_2v_3$ be one of the 61 non-termination codons. Then the codon rate matrix is parameterised as

$$\dddot{\mathbf{Q}}_{u,v} = \begin{cases} 0 \text{ if the codons differ at more than 1 position,} \\ \dddot{\pi}_v \text{ for a synonymous transversion,} \\ \kappa\dddot{\pi}_v \text{ for a synonymous transition,} \\ \omega\dddot{\pi}_v \text{ for a non-synonymous transversion,} \\ \omega\kappa\dddot{\pi}_v \text{ for a non-synonymous transition.} \end{cases} \tag{1.26}$$

In [GY94] codon models which take into account the chemical similarity of substituted amino acids are presented.

The codon and DNA models presented have a small number of parameters which can be trained by maximum likelihood (ML) on a given alignment. Reversible amino acid models, on the other hand, have 190 upper diagonal elements and so 189 free parameters after scaling as

in eq. 1.11. Reliable estimates of these parameters requires a significant amount of data. Thus amino-acid rate matrices are typically derived from databases of alignments, as is discussed in more detail below. However, different alignments and particularly different sites in an alignment have very different structural and functional constraints. Using a database derived rate matrix to describe all columns in an alignment leads to an inaccurate model of evolution at particular sites [Bru96, HB98]. One approach is to modify database derived rate matrix, labelled $\hat{\dot{\mathcal{E}}} = (\hat{\dot{\mathbf{Q}}}, \hat{\dot{\pi}})$ so that the stationary probabilities of the new rate matrix $\dot{\mathbf{Q}}$ are set to a given distribution $\dot{\pi}$ which is set (or trained) to reflect a particular alignment or site in an alignment. This can be achieved as described in [CAJ$^+$94, GW02] where the exchangeabilities matrix for the alignment is set to be equal to that estimated from a database, i.e. $\dot{\mathbf{S}}(f) = \hat{\dot{\mathbf{S}}}(f)$, where the equation for $\mathbf{S}(f)$ is given in eq. 1.15. This leads to the equation

$$\dot{\mathbf{Q}}_{uv} = \left(\frac{\dot{\pi}_v}{\hat{\dot{\pi}}_v}\right)^{1-f} \times \hat{\dot{\mathbf{Q}}}_{uv} \times \left(\frac{\hat{\dot{\pi}}_u}{\dot{\pi}_u}\right)^{f}. \tag{1.27}$$

The +gwF parameter $f$ is restricted to lie between 0 and 1 [GW02]), and can be thought of as the trade-off between frequencies in the equilibrium distribution resulting from pressure to mutate from ($f = 1$) and pressure to mutate towards ($f = 0$) a particular residue/base. The most common approach [CAJ$^+$94] is to set $f = 0$ which reduces equation 1.27 to

$$\dot{\mathbf{Q}}_{uv} = \left(\frac{\dot{\pi}_v}{\hat{\dot{\pi}}_v}\right) \hat{\dot{\mathbf{Q}}}_{uv} \tag{1.28}$$

**Accounting for variation between sites**

In many cases a single substitution model is used for every site in an alignment, in contrast to the profile HMM methods discussed in section 1.2 which have a different frequency distribution at each site. This is readily seen to be a drastic simplifying assumption for both protein and DNA alignments. Some regions in an alignment will be evolving slowly due to functional and/or structural reasons. Regions of DNA vary greatly in composition, for example in G+C content. Codon substitution patterns at neutrally evolving or positively selected sites will be different from those under purifying selection.

Yang [Yan93] proposed the use of a mixture of substitution models of the form eq. 1.12 each with different rates $r$ chosen from a discretised gamma distribution. Yang [Yan95] as well as Felsenstein and Churchill [FC96], further proposed correlating rates at adjacent sites via a first order spatial Markov chain. In [Yan95] it is assumed that the rate at a site is drawn

from a different discretised gamma distribution, and the variance of the gamma distribution is the hidden parameter of the spatial HMM. In [FC96] the rate (drawn from a finite set of categories) is itself the hidden parameter of the HMM. In [SH04], context dependent site-specific substitution models are used as part of an HMM framework. Their model is context dependent on the previous column:

$$P_{\mathcal{E}}(x^{t+\triangle t} = u | x^t = v) = \\ P_{\mathcal{E}}(x_1^{t+\triangle t} = u_1 | x_1^t = v) \prod_{i=2}^{n} P_{\mathcal{E}}(x_i^{t+\triangle t} = u_i | x_i^t = v_i, x_{i-1}^t = v_{i-1}, x_{i-1}^{t+\triangle t} = u_{i-1}) \quad (1.29)$$

Yang and Nielsen [NY98, YN00] describe models of codon evolution where the ratio of the rates of non-synonymous and synonymous substitution are allowed to vary between sites. These models have been successful in detection positive selection, as discussed in section 4.2.

Bruno [Bru96, HB98] learn site-specific rate matrices from a given alignment, using an amino-acid model and a codon model respectively. In both cases the site-specific rate matrices are defined in terms of the site-specific residue frequencies $\dot{\pi}$. In [Bru96] the EM algorithm is used to find the residue frequencies which optimise the likelihood of the column. In [TGJ96] the authors introduce a model for amino-acid evolution which has rate matrices specific to particular secondary structure states. A spatial HMM is used to correlate the hidden structural states along the length of the sequence. The authors demonstrated a significantly better likelihood fit to the alignment data, and used the model to derive phylogenetic trees as well as to label sites in the alignment with secondary structure states. This method was extended to accommodate more states in [LGTJ98] and to model transmembrane proteins specifically in [LG99].

Lartillot and Hervet [LP04] have recently defined a Bayesian mixture model which allows each site in an alignment to evolve according to a mixture of $K$ distinct evolutionary models $\dot{\mathbf{Q}}^k$, where $K$ is itself a parameter of the model and the $\dot{\mathbf{Q}}^k$ are parameterised as in eq. 1.28. Thus each class is parameterised by a different stationary probability $\dot{\pi}^k$. The authors define appropriate priors over the model parameters as well as tree space and present a MCMC sampling technique for identifying the ML model parameterisation and tree. In this way they are able to learn the optimal number of rate matrix categories in the data.

**Database derived protein rate matrices**

The original attempts to estimate $\hat{\mathbf{Q}}$ used maximum parsimony (MP) rather than maximum likelihood. Dayhoff et al. [DSO78] and [JTT92] used MP to estimate both the trees and ancestral sequences for multiple protein families, and counted the observed amino acid replacements along the tree to estimate the PAM matrices. Jones et al. [JTT92] extended this technique and applied it to a much larger database of protein families. To avoid observing transitions which are the product of multiple steps and to avoid assigning an ancestral sequence, the authors counted transitions based on pairwise sequence comparisons (where each sequence is used in only one comparison) between sequences which are more than 85% identical.

The maximum likelihood approach has been applied to estimating amino acid replacement rates in [AH96, YN98, AWMH00, WG01]. The first three of these estimated amino acid replacement rates in vertebrate mitochondrial, mammalian mitochondrial and chloroplast sequences respectively. Whelan and Goldman [WG01] apply an approximate form of ML training on a larger database of globular protein sequences. Holmes and Rubin [HR02] use expectation maximisation (EM) [DLR77] to train substitution models from sequence alignments and phylogenetic trees. The EM algorithm is designed to maximise the likelihood of data where some of the data is missing. In this case the missing data corresponds to the precise substitution history of the sequence. The model can also accommodate finding a pre-defined number of hidden substitution rate matrices in the data.

### 1.3.3 Likelihood calculation

For a given tree $T$ with branch lengths specified and evolutionary model $\mathcal{E}$, it is desirable to calculate the likelihood of a cluster of sequences $\{x^k\}$, $P(\{x^k\}|T,\mathcal{E})$. This likelihood is useful for several purposes: to evaluate different evolutionary models on a fixed tree, with the aim of finding the model that best fits the data; or to evaluate different trees with a fixed evolutionary model, with the aim of finding the tree which best fits the data. The likelihood can be calculated efficiently using Felsenstein's algorithm [Fel81]. Felsenstein's algorithm allows the summation over unknown states at internal nodes of the tree, and is closely related to the forward algorithm for for HMMs. In fact, as several authors have noted, the algorithm also allows summation over unknown states at the leaves of the tree (which might occur, for

example if there is a gap present in the alignment). Let $p_{u,k}$ denote the partial likelihood of all sequences $\{x^{k'}\}$ below node $n_k$ given sequence $u$ in the ancestral sequence at $n_k$. This algorithm proceeds by calculating in post-order (i.e. working upwards from the leaves),

$$p_{u,k} = \begin{cases} 1 \text{ if } n_k \text{ is a leaf node and } u \text{ matches } x^k \\ 0 \text{ if } n_k \text{ is a leaf node and } u \text{ does not match } x^k \\ \prod_h \sum_v p_{v,kh} \cdot P_{\mathcal{E}}(x^{kh} = v | x_k = u) \text{ otherwise} \end{cases} \tag{1.30}$$

where $n_{k1}, n_{k2}, \ldots$ are the child nodes of $n_k$. The term 'matches' (following [SH04]) has been used to include cases where $x^k$ contains a gap but is otherwise equal to $u$. This is effectively the same as treating the gap as missing data.

## 1.4 Outline of thesis

In this thesis, I focus on probabilistic modelling of protein domain evolution. Protein domain databases, such as Pfam [BCD+04] provide a valuable resource for studying protein domain evolution. To demonstrate the volume of data amenable for probabilistic analysis of the type described above, Pfam release 16.0 contains 7677 protein families covering 1.1m protein sequences and 264m residues. In the next two chapters of this thesis I investigate ways to model protein domains in order to improve protein domain detection and to extend the coverage of protein domain databases. Looking for distant homologues is important beyond simply extending residue coverage of domain databases. Arguably the most divergent members of a particular domain family are the most interesting for identifying the range of potential functions and partners for a particular domain as well as identifying fast evolving proteins. The final chapter of the thesis concerns looking for such fast evolving proteins in order to identify pseudogenes, as well as proteins and sites under positive selection.