BIOINFORMATICS ORIGINAL PAPER

Vol. 31 no. 1 2015, pages 40–47
doi:10.1093/bioinformatics/btu556

Sequence analysis

Advance Access publication August 31, 2014

Prediction of Indel flanking regions in protein sequences using a variable-order Markov model

Mufleh Al-Shatnawi*, M. Omair Ahmad* and M.N.S. Swamy*

Department of Electrical and Computer Engineering, Concordia University, QC H3G 2W1, Canada

Associate Editor: John Hancock

ABSTRACT

Motivation: Insertion/deletion (indel) and amino acid substitution are two common events that lead to the evolution of and variations in protein sequences. Further, many of the human diseases and functional divergence between homologous proteins are more related to indel mutations, even though they occur less often than the substitution mutations do. A reliable identification of indels and their flanking regions is a major challenge in research related to protein evolution, structures and functions.

Results: In this article, we propose a novel scheme to predict indel flanking regions in a protein sequence for a given protein fold, based on a variable-order Markov model. The proposed indel flanking region (IndelFR) predictors are designed based on prediction by partial match (PPM) and probabilistic suffix tree (PST), which are referred to as the PPM IndelFR and PST IndelFR predictors, respectively. The overall performance evaluation results show that the proposed predictors are able to predict IndelFRs in the protein sequences with a high accuracy and F_1 measure. In addition, the results show that if one is interested only in predicting IndelFRs in protein sequences, it would be preferable to use the proposed predictors instead of HMMER 3.0 in view of the substantially superior performance of the former.

Contact: m_alshat@ece.concordia.ca or omair@ece.concordia.ca or swamy@ece.concordia.ca.

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on March 23, 2014; revised on July 23, 2014; accepted on August 12, 2014

1 INTRODUCTION

As new protein sequences are discovered on an everyday basis and protein databases continue to grow exponentially with time, analysis of protein families, understanding their evolutionary trends and detection of remote homologues have become extremely important. The proteins that evolve from the same ancestor protein are called *homologous proteins*. A protein molecule is created in a cell as a chain of amino acids, called the polypeptide chain. A polypeptide chain can be represented as a string of characters by using the letter code of each amino acid. This string of characters is called the *primary structure* of a protein.

In addition to the primary structure, a protein has secondary and tertiary structures (Yona, 2011). *Secondary structure* of a protein refers to well-determined local sequence elements, such as an *alpha helix*, a *beta strand* or any other local sequence element that is neither a helix nor a strand. These other local sequences, usually called *loops* or *coils*, may have a large variety of shapes. These secondary structure elements of a protein can be combined together to create a motif, which is a simple combination of a few consecutive secondary structure elements with a specific geometric arrangement, such as *helix-loop-helix* or *strand-loop-helix*. Some, but not all, motifs are associated with specific biological functions. The *tertiary structure* of a protein refers to the 3-dimensional structure of the protein, where the secondary structure elements form the physical core of the 3-dimensional structure, and loops are located on the surface of the tertiary structure. A domain refers to a combination of several secondary elements and motifs, which may not necessarily be contiguous and which are usually packed in a compact structure. A protein may contain a single domain or several different domains, or several copies of the same domain.

Normally, the proteins are classified into families based on the existence of a specific motif or domain in their structure, where the existence of such a motif or domain has a major indication about the biological role of the protein. The structural classification of proteins (SCOP) database is a comprehensive ordering of all the proteins of the known structures, according to their evolutionary and structural relationships (Andreeva *et al.*, 2008), where the fundamental unit of this classification is a domain. In this database, the proteins have been classified into families, superfamilies, common fold and, finally, into classes at the top level of the structural hierarchy.

It is known that new proteins have evolved mainly through indel mutations (Grishin, 2001; Zhang *et al.*, 2010). Indel mutations have been found to occur more often in the loop regions (Benner *et al.*, 1993; Hsing and Cherkasov, 2008), and mainly in essential proteins and in those proteins that interact highly with others (Chan *et al.*, 2007). The functional divergence between homologous proteins may also be caused by indel mutations that occur in the regions between secondary structures of a protein (Jiang and Blouin, 2007). Further, it has been found that differences among species, as well as many of the human diseases, are related to indel mutations, which occur less often than substitution mutations do (Britten *et al.*, 2003; Chen *et al.*, 2009; Duval and Hamelin, 2002).

Sequence alignment is one of the most commonly performed tasks in bioinformatics, and has been used in many applications, including sequence annotation, phylogenetic tree estimation, evolutionary analysis, secondary structure prediction and database search (Notredame, 2002). In recent years, considerable effort has been devoted to the development of protein alignment algorithms that can efficiently detect mutations, and infer

* To whom correspondence should be addressed.

structural and functional relationships among the aligned protein sequences. By using position-variant probability to score indel mutations, the profile hidden Markov model (pHMM; Eddy, 1998; Krogh *et al*., 1994) is able to use the fact that indel mutations occur more frequently in some parts of a protein more than other parts (e.g. in the loop regions) (Benner *et al*., 1993; Hsing and Cherkasov, 2008). Several software packages, such as HMMER (Finn *et al*., 2011) and SAM (Hughey and Krogh, 1996; Karplus *et al*., 1998), have implemented the pHMM-based alignment algorithms. Among these packages, HMMER is the most-used software package in protein database search and comparison. A collection of pHMMs covering many protein families have been generated using HMMER, and they are available in the Pfam database (Punta *et al*., 2012). The disadvantage of using pHMM-based alignments for detecting mutations would be that the corresponding alignment algorithms assume the occurrence of mutations in the protein sequence to follow a fixed first-order Markov chain.

When a pair of protein sequences has been aligned, an indel region is defined as a gap in any of the two sequences. Segments of protein sequence immediately before and after an indel region are called flanking regions (see Supplementary Fig. S3). The database called IndelFR (Indel Flanking Region) database contains sequence and structure information of 2 925 017 indels with their flanking regions, including their positions, length, amino acid composition and secondary structure (Zhang *et al*., 2012). In the IndelFR database, the structure-based sequence alignment program PDBeFold (Krissinel and Henrick, 2004) has been used to align homologous non-redundant proteins obtained from the ASTRAL95 database (Chandonia *et al*., 2004), which in turn has used the non-redundant protein domains from the SCOP database that have similarity levels as high as 95%. It has been found that there exists a strong relationship between indels and their flanking regions (Chen *et al*., 2009; Tian *et al*., 2008; Zhang *et al*., 2011; Zhu *et al*., 2009).

The objective of this article is, therefore, to propose a novel scheme to predict indel flanking regions in a protein sequence, based on a variable-order Markov model (VOMM) of the flanking regions. We propose two indel flanking region (IndelFR) predictors: one predictor using prediction by partial match (PPM) (Cleary and Witten, 1984) and other using probabilistic suffix tree (PST) (Ron *et al*., 1996).

2 METHODS

2.1 VOMM for flanking regions

In the IndelFR database, indels and their flanking regions are extracted from alignments by dividing equally the region between two adjacent indels, and by taking 10 amino acids as the upper limit for the flanking regions. It has been shown by Zhang *et al*. (2011) that the impact of an indel on its flanking regions reduces dramatically as we move away from the indel, and this impact is negligible after 10 amino acids. In this study, we classify the indel regions stored in the IndelFR database according to the number of amino acids in the flanking regions as follows:

(i) If the number of amino acids between two indels is >20, then we consider each of the two flanking regions between them to have exactly 10 amino acids.

(ii) If the number of amino acids between two indels is ≤ 20, but ≥ 2, we still consider these two indels as two separate indels, and the region between them split equally or as equally as possible to define the flanking regions between the two indels.

(iii) If the number of amino acids between two adjacent indels in the same sequence is unity, then we combine the two indels along with the single amino acid in between to treat the combination as a single indel.

(iv) If the number of amino acids between two adjacent indels that are not in the same sequence is unity, then we treat these two indels as distinct. Thus, the right (left) flanking region of one of the indels and the left (right) flanking region of the other indel would each have only one amino acid.

We refer to an indel along with its left and right flanking regions as an *indel flanking region* (*IndelFR*). Figure 1 shows an example illustrating each of the above situations. It is noted that there are three IndelFRs for each of the two alignments shown in Figure 1.

It is to be recalled that in the IndelFR database, a given protein sequence has been aligned with a large number of protein sequences that belong to the same superfamily. For example, consider the protein *d1allb_*. In the IndelFR database, 87 pairwise alignments have been carried out for this protein (see Supplementary Table S2). From these alignments, we now identify all the IndelFRs for the protein *d1allb_* and mark off *IndelFR segments*, which are the segments of the protein sequence to which all the identified IndelFRs collectively belong to. For the protein sequence *d1allb_*, these segments are observed to be from position 88 to 97 and from 100 to 115, and no indel is located outside these segments. This process can be applied to any of the protein sequences available in the IndelFR database to obtain its IndelFR segments. Figure 2 shows such segments for some of the protein sequences selected from the *Globin-like* superfamily, the segments being marked by thick lines.

The above results strongly suggest that the IndelFRs for a given protein sequence are conserved within only the IndelFR segments. This is a significant finding, which we will use later in this section in training the model for the proposed IndelFR predictor.

We now present a technique to build an IndelFR predictor for a given protein fold. In this study, the protein folds are selected from the following protein classes: *All-α proteins*, *All-β proteins* and *α and β proteins* (*a*/*b*) (see Supplementary Tables S3–S5). It is noted that each protein fold contains one or more superfamilies and each superfamily contains one or more protein families. Therefore, the proposed IndelFR predictor for a given protein fold can be used to predict IndelFRs in protein sequences that belong to different protein families within the same fold.

Because we already know how to obtain the locations of the IndelFR segments for a protein sequence, we build our proposed model for the IndelFR predictor by confining only to the IndelFR segments of each of the protein sequences in a given fold. We extract the flanking regions of all the sequences in the fold and divide them into two sets, the left and right sets. The left set contains all the left flanking regions, whereas the

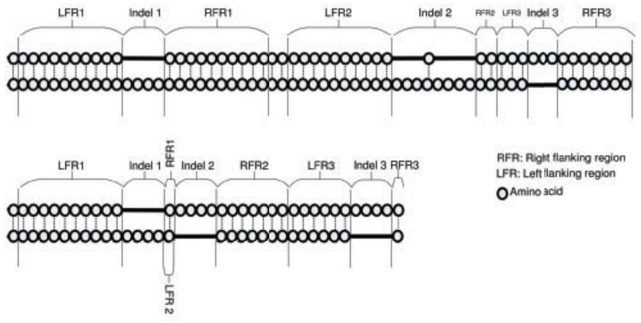

Fig. 1. Indel regions classified according to the number of amino acids in the flanking regions

right set contains all the right flanking regions. The flanking regions in either of the two sets have, in general, different lengths, as these lengths, according to our earlier assumption, can vary between 1 and 10.

The proposed IndelFR predictor for a given protein fold contains two VOMMs (Buhlmann and Wyner, 1999; Ron *et al.*, 1996): one for the left set and the other for the right set. These models learn the conditional probability $P_k(\sigma|s_{i-k}\ldots s_{i-1})$ of observing a particular amino acid $\sigma \in \Psi_{protein}$ at position *i*, given a context $\mathbf{s}=s_{i-k}\ldots s_{i-1}$ of length *k*, where the context $\mathbf{s}=s_{i-k}\ldots s_{i-1}$ represents all the previously observed amino acids before σ is observed at position *i*, each s_j, $j \in \{i-1,\ldots,i-k\}$, representing one of the possible 20 amino acids. The context length *k* could vary depending on the size and nature of the string of amino acids in a flanking region, and $\Psi_{protein}$ is the alphabet set containing all the amino acid symbols. We select a VOMM instead of a fixed-order one for the following reasons: (i) the chosen model should take into consideration varying sizes of the flanking regions in a set, and (ii) it should take care of situations where a flanking region in which a particular amino acid $\sigma \in \Psi_{protein}$ does not exist for a given context of length *m*. In the latter case, a VOMM would allow us to reduce the length of the context to be less than *m*.

Over the years, many VOMM structures, such as Lampel-Ziv compression (Nisenson *et al.*, 2003), context tree weighting (Willems *et al.*, 1995), PPM (Cleary and Witten, 1984) and PST (Ron *et al.*, 1996), have been proposed. We will briefly discuss here only two of them, namely, PPM and PST, which are among the most commonly used structures in prediction applications. The effectiveness of PPM and PST for prediction of sequences in various applications has been examined by Begleiter *et al.* (2004). PST has also been used to model DNA sequences by Ron *et al.* (1996). Further, it has been used in modeling and prediction of protein families by Bejerano and Yona (2001).

In PPM, to build a VOMM for the left (right) set of flanking regions, we start by analyzing a subset of the left (right) flanking regions as the training set and counting the number of occurrences of the amino acid σ immediately after the context **s**, that is, counting the number of occurrences of the pattern $\mathbf{s}\sigma$ in the training set for each amino acid $\sigma \in \Psi_{protein}$ and for each context $\mathbf{s}=s_{i-k}\ldots s_{i-1}$ of length *k*, where each s_j, $j \in \{i-1,\ldots,i-k\}$, represents one of the possible 20 amino acids. The context length *k* varies from zero to *D*, where *D* is the memory length of the VOMM. Hence, for each value of *k*, we can compute the conditional empirical probability $\tilde{P}_k(\sigma|\mathbf{s})$ (Yona, 2011) as

$$\tilde{P}_k(\sigma|\mathbf{s}) = \frac{N_{\mathbf{s}\sigma}}{\sum_{s_j\in\Psi_{protein}} N_{\mathbf{s}s_j}} \tag{1}$$

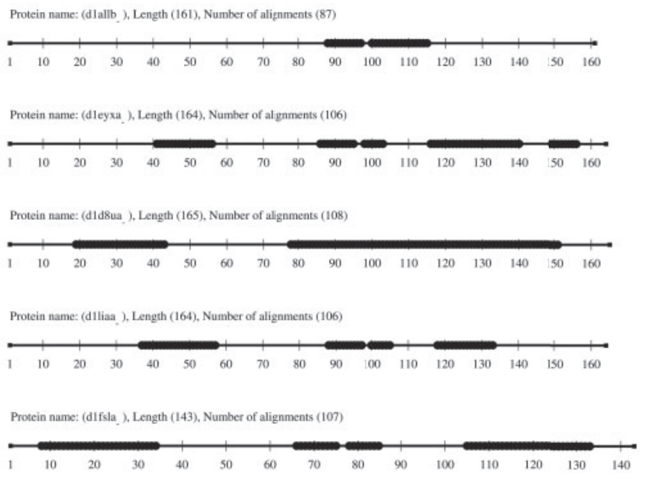

Fig. 2. IndelFR segments where flanking regions may exist for some selected protein sequences. The segments are indicated by thick lines

where $N_{\mathbf{s}\sigma}$ is the number of occurrences of the pattern $\mathbf{s}\sigma$ in the training set. For $k = 0$, we can calculate the conditional empirical probability, $\tilde{P}(\sigma|\epsilon)$, where ϵ represents an empty context. PPM handles the zero frequency problem by going through the mechanisms of *escape* and *exclusion* (Yona, 2011). In the escape mechanism, for each context $\mathbf{s}=s_{i-k}\ldots s_{i-1}$ of length *k*, we make use of a probability mass $P_k(escape|\mathbf{s})$ for all the amino acids that do not appear after the context $\mathbf{s}=s_{i-k}\ldots s_{i-1}$ in the training set. There are different ways of defining the escape probabilities for a context. These definitions are generally based on intuition and experience, and not on any underlying theory. For example, in Moffat (1990), this escape probability has been defined as

$$P_k(escape|\mathbf{s}) = \frac{|\Psi_{\mathbf{s}}|}{|\Psi_{\mathbf{s}}| + \sum_{s_j\in\Psi_{\mathbf{s}}} N_{\mathbf{s}s_j}} \tag{2}$$

where $\Psi_{\mathbf{s}}$ is a set of amino acids appearing after the context $\mathbf{s}=s_{i-k}\ldots s_{i-1}$, i.e. $\Psi_{\mathbf{s}}=\{\sigma : N_{\mathbf{s}\sigma}>0\}$, and $|\Psi_{\mathbf{s}}|$ denotes the number of elements in $\Psi_{\mathbf{s}}$. Accordingly, the conditional probability is modified as

$$P_k(\sigma|\mathbf{s}) = \begin{cases} P_k(\sigma|\mathbf{s}) & \text{if } \sigma \in \Psi_{\mathbf{s}} \\ P_k(escape|\mathbf{s})P_{k-1}(\sigma|s_{i-k+1}\ldots s_{i-1}) & \textit{otherwise} \end{cases} \tag{3}$$

where

$$P_k(\sigma|\mathbf{s}) = \frac{N_{\mathbf{s}\sigma}}{|\Psi_{\mathbf{s}}| + \sum_{s_j\in\Psi_{protein}} N_{\mathbf{s}s_j}} \tag{4}$$

In the above equation, if $P_{k-1}(\sigma|s_{i-k+1}\ldots s_{i-1})$ is zero, then Equation (3) is recursively modified by using contexts of shorter lengths as shown below:

$$P_{k-1}(\sigma|s_{i-k+1}\ldots s_{i-1}) = \Big(P_{k-1}(escape|s_{i-k+1}\ldots s_{i-1}) \; P_{k-2}(\sigma|s_{i-k+2}\ldots s_{i-1})\Big) \tag{5}$$

In the exclusion mechanism, if a prediction fails for a certain context, then the unseen amino acid cannot be one of the amino acids that has been observed after that context, and the relevant alphabet set for all the shorter contexts should be reduced by eliminating these observed amino acids. Hence, every amino acid $\sigma \in \Psi_{\mathbf{s}}$ observed after context $\mathbf{s}=s_{i-k}\ldots s_{i-1}$ is excluded, when we calculate the conditional probability for all contexts shorter than $\mathbf{s}=s_{i-k}\ldots s_{i-1}$. An example of the PPM structure of VOMM is given in the Supplementary Material Section 1. In the PST structure, a single tree with depth *D* is constructed to represent a VOMM of memory length *D*. The nodes in the tree have different degrees varying from zero (for leaves) to the size of the alphabet set $\Psi_{protein}$ (for the internal nodes and the root). Each edge in the tree is labeled by a single amino acid from the set $\Psi_{protein}$. Each node in the tree is labeled by a unique context $\mathbf{s}=s_{i-k}\ldots s_{i-1}$, where the context length varies from zero (for the root) to *D*. Also, each node is assigned a conditional probability $P_k(\sigma|\mathbf{s})$, where $\sigma \in \Psi_{protein}$. It should be noted that the context $\mathbf{s}=s_{i-k}\ldots s_{i-1}$ is generated by moving from the node to the root (i.e. in PST, the father of the node labeled by $s_1s_2s_3$ is the node s_2s_3, and not s_1s_2 as in a regular suffix tree).

To build the PST structure **T** for the left or for the right flanking region, we need to set the following four parameters:

(i) The memory length parameter *D* of PST.
(ii) The context threshold parameter N_T, where $N_T=c\cdot m$, c $(0<c<1)$ being a constant and *m* the total number of flanking regions in the left or right set. The parameter N_T determines which contexts would be included in building the PST structure. If the

number of occurrences of a context is less than N_T, then such a context is excluded in building the structure.

(iii) The parameter r is used to determine whether the context $\mathbf{s} = s_{i-k} \ldots s_{i-1}$ contributes additional information in predicting the amino acid σ relative to its 'parent' or 'suffix' context $s_{i-k+1} \ldots s_{i-1}$, denoted by *suf*(**s**). The ratio $P_k(\sigma|\mathbf{s})/P_{k-1}(\sigma|suf(\mathbf{s}))$ is chosen to be outside the interval $(1/r, r)$. To make the contribution of this context to be sensitive, r is chosen to be $1+\delta$, δ being a small quantity.

(iv) The parameter B_s is chosen to be $B_s = 5 \cdot |\Psi_s|$, as suggested by Henikoff and Henikoff (1996), to ensure that for a given context the probability of an amino acid $\sigma \in \Psi_{protein}$ does not become zero.

Let N_s be the number of occurrences of the context $\mathbf{s} = s_{i-k} \ldots s_{i-1}$ in the training set of the left (right) flanking regions, $N_{\mathbf{s}\sigma}$ the number of occurrences of the pattern $\mathbf{s}\sigma$ in the left (right) training set and $P_k(\sigma|\mathbf{s})$ the conditional probability associated with the node labeled by the context $\mathbf{s} = s_{i-k} \ldots s_{i-1}$. The various steps to build the PST structure for the left (right) training set are as follows.

Step 1: Create a tree **T** with a single root node labeled by an empty context ϵ, and create an empty set $\mathbf{set}_{ptr}$.

Step 2: Add to the set $\mathbf{set}_{ptr}$ all the contexts of length unity that have occurred more number of times than N_T, the context threshold (i.e. $N_s > N_T$).

$$\mathbf{set}_{ptr} \leftarrow \{\mathbf{s}|N_\mathbf{s} > N_T\}$$

Step 3: Select a context from $\mathbf{set}_{ptr}$.

Step 4: Test if there is an amino acid $\sigma \in \Psi_s$ that has a conditional empirical probability $\tilde{P}_k(\sigma|\mathbf{s})$ given by Equation (1) satisfying the following inequality:

$$\tilde{P}_k(\sigma|\mathbf{s}) > \frac{1}{|\Psi_{protein}|}$$

Step 5: Test if there is an amino acid σ (not necessarily the same amino acid as in Step 4) from alphabet Ψ_s satisfying the condition

$$\frac{\tilde{P}_k(\sigma|\mathbf{s})}{\tilde{P}_{k-1}(\sigma|suf(\mathbf{s}))} = \begin{cases} \geq r \\ or \\ \leq \dfrac{1}{r} \end{cases}$$

Step 6: If the conditions in Steps 4 and 5 are both satisfied, go to Step 7; otherwise (i.e. condition in Steps 4 or 5 is not satisfied), go to Step 10.

Step 7: Test if the parent node of the context **s**, labeled by *suf*(**s**), already exists in the tree **T**. If yes, add the node corresponding to this context **s** to the tree **T**.

Step 8: If the parent node for **s** in Step 7 does not exist, then create a node for this context **s** and for its parent node. If the parent node of the latter does not exist, then repeat this procedure until an existing parent node in the tree is reached.

For example, assume a node labeled by context $\mathbf{s} = C$ exists in the tree, and we are trying to add a node labeled by context $\mathbf{s} = ABDC$ to the tree. In PST, the parent node for the context $\mathbf{s} = ABDC$ is a node labeled by context BDC, which does not exist in the tree. Also, the parent node of the context BDC is a node labeled by context DC, which also does not exist in the tree. But the parent node of the context DC, namely, the node labeled C exists in the tree. Hence, we have to add two more nodes labeled BDC and DC, in addition to the node labeled $ABDC$ to the tree. This is illustrated in the Supplementary Figure S4.

Step 9: Adjust the conditional probability for each added node in Steps 7 or 8, so that the probability of an amino acid $\sigma \in \Psi_{protein}$ for a given context is given by

$$P_k(\sigma|\mathbf{s}) = \frac{N_{\mathbf{s}\sigma} + \left(\frac{1}{|\Psi_{protein}|}\right) B_s}{\sum_{s_j \in \Psi_s} N_{\mathbf{s}s_j} + B_s}$$

Step 10: If the length of $\mathbf{s} < D$, and there exists a pattern $\sigma\mathbf{s}$, which has occurred more number of times than N_T (i.e. $N_{\sigma\mathbf{s}} \geq N_T$), then add the pattern $\sigma\mathbf{s}$ to $\mathbf{set}_{ptr}$.

$$\mathbf{set}_{ptr} \leftarrow \{\sigma\mathbf{s}|\sigma \in \Psi_{protein} \textbf{ and } N_{\sigma\mathbf{s}} \geq N_T\}$$

Step 11: Remove the context **s** from $\mathbf{set}_{ptr}$, and repeat Steps 3–11 until $\mathbf{set}_{ptr}$ becomes empty.

An example of building a PST structure with the parameters $D = 2$, $N_T = 0.01$ and $r = 1.05$ is given in the Supplementary Material Section 2.

2.2 Prediction of indel flanking regions using VOMM

Given a test protein sequence $\mathbf{S}^n = s_1 s_2 s_3 \ldots s_n$ of length n, we scan it using a running window of length L moving it one amino acid at a time. To determine whether the string of amino acids within a window contains a flanking region, we compute the probability of this string using VOMM.

For a VOMM, we use $P(\mathbf{win}_i)$ to denote the probability of the string $\mathbf{seg}_i = s_i s_{i+1} \ldots s_{i+L-1}$ of length L. If VOMM has a memory length $D < L$, then $P(\mathbf{win}_i)$, which is also referred to as the likelihood of $\mathbf{win}_i$, is given by Yona (2011):

$$P(\mathbf{win}_i) = P_0(s_i)P_1(s_{i+1}|s_i)\ldots \\ P_D(s_{i+L-1}|s_{i+L-D}\ldots s_{i+L-2}) \tag{6}$$

We calculate the various probabilities on the right side of Equation (6) by using PPM or PST. If the probability $P_k(s_j|s_{j-k}s_{j-k+1}\ldots s_{j-1})$ for s_j, $(i \leq j \leq i+L-1)$ does not exist, we proceed as follows:

(a) In the case of PPM, we use the escape and exclusion mechanisms, in conjunction with Equation (3), to calculate each of the probabilities in Equation (6).

(b) In the case of PST, we find the longest suffix of the context $s_{j-k} s_{j-k+1} \ldots s_{j-1}$ that exists in the tree. Assuming the longest suffix of the context $s_{j-k}s_{j-k+1}\ldots s_{j-1}$ that exists in the tree to be $s_{j-t}s_{j-t+1}\ldots s_{j-1}$, $(0 \leq t < k)$, then

$$P_k(s_j|s_{j-k}s_{j-k+1}\ldots s_{j-1}) = P_t(s_j|s_{j-t}s_{j-t+1}\ldots s_{j-1}) \tag{7}$$

Maximizing the likelihood $P(\mathbf{win}_i)$ is equivalent to minimizing the *average log-loss function* in a lossless compression algorithm (Yona, 2011), where the average log-loss function is defined as

$$loglossP(\mathbf{win}_i) = -\frac{1}{L}(\log P_0(s_i) + \log P_1(s_{i+1}|s_i) + \\ \log P_2(s_{i+2}|s_1 s_{i+1}) + \ldots + \\ \log P_D(s_{i+L-1}|s_{i+L-1-D}\ldots s_{i+L-2})) \tag{8}$$

the logarithm being taken to base 2. It may be noted that in compression, the log-loss function represents the average number of bits per character, as the term $(-\log P_k(s_j|s_{j-k}s_{j-k+1}\ldots s_{j-1}))$ is the shortest code that can be assigned to the character s_j, given the conditional probability distribution $P_k(s_j|s_{j-k}s_{j-k+1}\ldots s_{j-1})$. Therefore, $\mathbf{win}_i$ contains a flanking region if it has a low average log-loss value compared with that of its neighboring windows.

An example illustrating the steps for calculating the probability of a particular segment using PPM or PST is given in the Supplementary Material Sections 1 and 2.

The proposed IndelFR predictor for a given protein fold can be built using PPM or PST. We build the left PPM (LPPM) and the left PST (LPST) for the left set, and build the right PPM (RPPM) and the right PST (RPST) for the right set. LPPM and RPPM are combined together to form a *PPM IndelFR predictor* for memory length D. Similarly, LPST and RPST are combined together to form a *PST IndelFR predictor* for memory length D. Such IndelFR predictors are built for various values of the memory length D, $D \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, so that we can determine the value of D that results in the best performance in predicting the locations of IndelFRs. The procedure to extract the predicted locations of these regions in the test protein sequence using PPM IndelFR predictor is given in Algorithm 1. A similar procedure is applied for the proposed PST IndelFR predictor.

Algorithm 1: Procedure to extract the predicted locations of IndelFRs in a test protein sequence using the proposed PPM IndelFR predictor with a memory length D.

Step 1: Scan the test protein sequence $\mathbf{S}^n = s_1 s_2 s_3 \ldots s_n$ of length n using a running window of length $L = 10$.

$$\mathbf{win}_i = s_i s_{i+1} \ldots s_{i+9} \quad 1 \le i \le (n-9)$$

Step 2: Compute and store the average log-loss values for each window using LPPM and RPPM with a memory length D using Equation (8).
Step 3: From the LPPM average log-loss values, choose the mean of these values as the threshold. Then, find the locations of the local minima that have values below the threshold.
Step 4: Repeat Step 3 using the RPPM log-loss values and find the locations of the local minima.
Step 5: Find the locations of IndelFRs in the test protein sequence by identifying each of the LPPM minimum locations that is immediately followed by an RPPM minimum location. The identified LPPM and the corresponding RPPM minimum locations represent the start locations of the predicted left and right flanking regions, respectively, and each flanking region (left or right) has a length of at most 10. For each selected minimum location at v, the predicted locations for this flanking region (left or right) are limited to $v, v+1, v+2, \ldots, v+9$.
Step 6: For the test protein sequence, use the actual locations of IndelFRs and the predicted locations to determine the *accuracy* and the *F_1 measure*.

3 RESULTS AND DISCUSSION

In this section, we evaluate the performance of the predictors proposed in Section 2. For this purpose, from the SCOP database, we select 11, 14 and 18 protein folds from different protein classes: *All-α proteins*, *All-β proteins* and *α and β proteins* (*a*/*b*), respectively (see Supplementary Tables S3–S5). This selection is confined to those protein folds that have indel flanking regions listed in the IndelFR database. We build the PPM and PST IndelFR predictors proposed in Section 2 for each of the selected protein folds. These predictors are built for various values of the memory length D, $D \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, so that we can determine the value of D that provides the best performance.

We use the k-fold cross-validation method for training and testing the proposed IndelFR predictors, where $k = 10$. Consequently, k iterations of training and testing are performed for each predictor. In the training phase, we train the IndelFR predictor for a given protein fold using the indel flanking regions listed in the IndelFR database. In the testing phase, we first test the trained predictors on the protein sequences from the same protein fold belonging to the IndelFR database and next, on the set of protein sequences from the same protein fold but belonging to the sequence alignment benchmark (SABmark 1.65) (Walle *et al.*, 2005). Finally, the performance of the two proposed predictors is compared with that using the latest version of the alignment software HMMER, HMMER 3.0 (Finn *et al.*, 2011).

We evaluate the performance of the proposed predictors using the measures of *accuracy* and *F_1 measure*, which are the commonly used metrics in the evaluation of the performance of prediction techniques in bioinformatics (Fawcett, 2006; Sonego *et al.*, 2008) (see Supplementary Material Section 3 for more details).

3.1 Prediction in IndelFR database

The average log-loss values for each test protein sequence are computed for each of the two proposed predictors. For the purpose of illustration, the average log-loss values using the two predictors for the protein sequence *d1liab_* are shown in Figure 3. It is seen from this figure that the average log-loss value around the flanking regions is much less than that around the other regions. It should be noted that for each of the test protein sequences, we follow the steps outlined in Algorithm 1 to extract the predicted locations of IndelFRs, and to calculate both the accuracy and the F_1 measure. As seen from Figure 3a, the PPM IndelFR predictor predicts the locations (A, B), (C, D), (E, F) and (G, H) as the start locations for IndelFRs (left and right, respectively), while this predictor ignores the RPPM minimum location (I), as it is not preceded by an LPPM minimum location. To compute the accuracy and the F_1 measure, we use the actual IndelFRs shown in Figure 3c taken from the IndelFR database, and the locations predicted by the proposed PPM IndelFR predictor. The accuracy and F_1 measure are found to be 81 and 77%, respectively. In a similar manner, using the results shown in Figure 3b, we

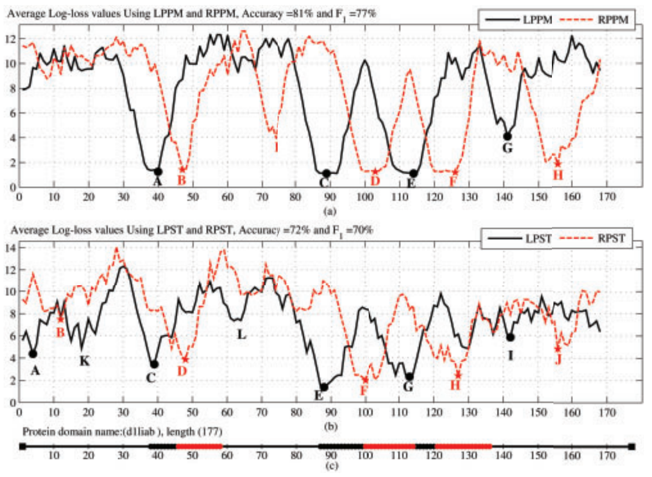

Fig. 3. Average log-loss values for the *d1liab_* protein sequence: (**a**) using PPM IndelFR predictor and (**b**) using PST IndelFR predictor. (**c**) Ground truth for the IndelFR taken from the IndelFR database (Zhang *et al.*, 2012). Solid dots represent the start locations of the predicted left flanking regions and the stars that of the predicted right flanking regions

determine the accuracy and F_1 measure for the PST IndelFR predictor to be 72 and 70%, respectively.

The average accuracy and F_1 measure values of the PPM predictor for various values of the memory length D, $D \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, are obtained for each of the 11 chosen protein folds from the *All-α protein* class, and are shown in Figure 4. It is observed from this figure that the best choice for the memory length D is 4. Further, the accuracy varies from 74 to 98% and the F_1 measure from 54 to 99% for the various folds. The average accuracy and F_1-measure values of the PPM predictor, with a memory length of 4, over the *All-α protein* class are 91 and 92%, respectively. In a similar manner, the average accuracy and F_1-measure values of the PST predictor for various values of the memory length D, $D \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, are obtained for each the 11 chosen protein folds from the *All-α protein* class (see Supplementary Fig S5 and S6). The results for the PST predictor strongly suggest that the best choice for the memory length D is again 4. Further, the accuracy varies from 63 to 96% and the F_1 measure from 54 to 97% for the various folds. The average accuracy and F_1-measure values of the PST predictor, with a memory length of 4, over the All-α protein class are 88 and 89%, respectively.

The average accuracy and F_1 measure values of the proposed PPM and PST predictors for various values of the memory length D, are obtained for each of the remaining 14 and 18 protein folds from the protein classes, *All-β protein* class and *α and β protein* (*a*/*b*) class, respectively (see Supplementary Tables S6–S13). These results show that the best choice for the memory length D is still 4 for all the selected protein folds. In addition, the results indicate that the proposed predictors perform better on those protein folds that have a large number of protein sequences. The average accuracy and F_1 measure values of the proposed PPM and PST predictors with $D = 4$, for the selected 11, 14 and 18 protein folds from the three protein classes are shown in Figure 5.

Average performances in terms of the accuracy and F_1 measure, over all the protein sequences contained in the 11, 14 and 18 protein folds of the IndelFR database belonging to the *All-α protein*, *All-β protein* and *α and β protein* (*a*/*b*) classes, respectively, for the two proposed predictors are given in Table 1. The table shows that the proposed PPM and PST predictors with $D = 4$ provide about the same average performance.

3.2 Prediction in SABmark 1.65

To have a more stringent assessment of the performance of the proposed predictors, we now test the two predictors with $D = 4$ on the sequence alignment benchmark (SABmark 1.65) (Walle *et al.*, 2005). It should be noted that the SABmark is generated from the SCOP database, and covers the entire known protein fold space with two sets, referred to as the *Superfamily set* and the *Twilight set*. The similarity level between any two protein sequences is <50% in the Superfamily set, while it is at most 25% in the Twilight set, in contrast to that in the IndelFR database, which contains protein sequences that have a similarity level that could be as high as 95%.

To evaluate the performance of the proposed PPM and PST predictors on the Superfamily and Twilight sets, we select protein sequences from the protein folds belonging to these sets; the folds chosen are only those for which the predictors have already been designed using the IndelFR database. The average accuracy and F_1 measure of the proposed predictors for each of the above protein folds are given in Supplementary Table S14. The average performance of the proposed predictors is also given in Table 1 for the Superfamily and Twilight sets. These results show that the proposed predictors are still able to predict the IndelFRs in the selected protein folds from both the sets with a high accuracy and F_1 measure, even though the similarity level between any two protein sequences is at most 50% in the case of the Superfamily set, and at most 25% in the case of the Twilight set. In addition, the results show that the performances of the two proposed predictors are almost the same for both sets, the average accuracies being ~75% and the average F_1 measures being ~79%. The average accuracy values and average F_1-measure values for the proposed IndelFR predictors for the selected protein folds from the three protein classes are shown in

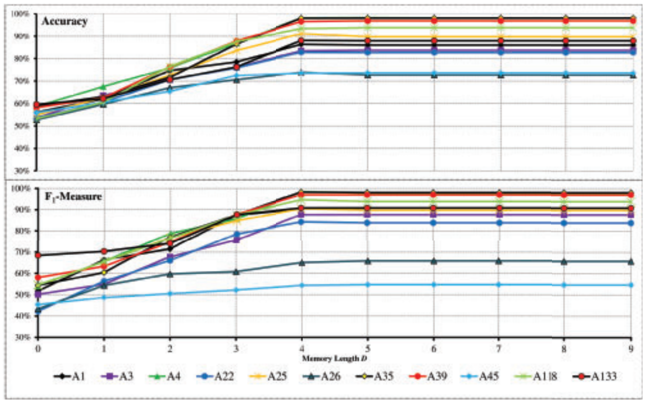

Fig. 4. Average accuracy and F_1-measure values for the PPM IndelFR predictor for different protein folds selected from the *All-α protein* class for various values of the memory length D, where A1, A3, A4, A22, A25, A26, A35, A39, A45, A118 and A133 are the protein folds (see Supplementary Table S3)

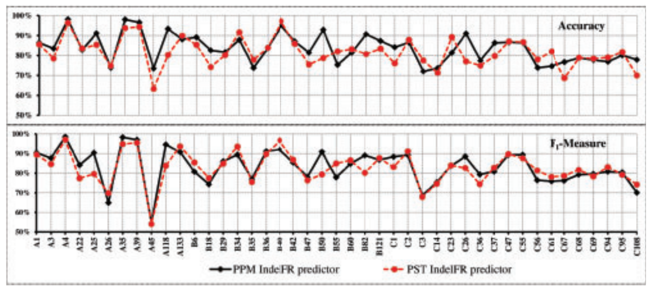

Fig. 5. Average accuracy and F_1-measure values for the proposed PPM and PST IndelFR predictors with $D = 4$ for different protein folds selected from the *All-α protein*, *All-β protein* and *α and β protein* (*a*/*b*) protein classes for memory length $D = 4$, where A1, A3, A4, A22, A25, A26, A35, A39, A45, A118, A133, B6, B18, B29, B34, B35, B36, B40, B42, B47, B50, B55, B60, B82, B121, C1, C2, C3, C14, C23, C26, C36, C37, C47, C55, C56, C61, C67, C68, C69, C94, C95 and C108 are the protein folds (see Supplementary Tables S3–S5)

Figures 6 and 7 for the Superfamily and Twilight sets, respectively.

3.3 Comparison with HMMER

The performance of the proposed predictors with $D = 4$ is now compared with that obtained using the latest version of the alignment software HMMER, HMMER 3.0 (Finn *et al.*, 2011). HMMER 3.0 implements the alignment of a protein sequence with the pHMM representing a particular protein family. A collection of pHMMs covering many protein families is available in the Pfam database (Punta *et al.*, 2012). To be able to make this comparison, it is necessary to find the Pfam pHMMs for all protein families that belong to a protein fold for which the PPM and PST predictors have already been designed using the IndelFR database. The protein families for the selected 11, 14 and 18 protein folds from the three protein classes are given in Supplementary Tables S3–S5. Prediction performance using HMMER 3.0 is obtained on the IndelFR database, as well as on the Superfamily and Twilight sets, and the results for the selected 11, 14 and 18 protein folds are given in Supplementary Tables S15–S20. The average performances obtained using HMMER 3.0 on the IndelFR database and on the Superfamily and Twilight sets, are also included in Table 1. The results indicate that the proposed predictors significantly outperform that obtained using HMMER 3.0 in terms of both accuracy and F_1 measure.

It should be noted that the proposed IndelFR predictors are more general than when HMMER 3.0 is used in that the proposed PPM or PST predictor for a given protein fold is capable of predicting the indel flanking regions for any protein sequence from any protein family in that fold, whereas HMMER 3.0 has to use different pHMMs depending on the family of the protein fold to which the protein sequence belongs. For instance, we have to design only one IndelFR predictor for the *Globin-like* fold, while HMMER 3.0 has to use five different pHMMs (see Supplementary Tables S3–S5).

4 CONCLUSION

In conclusion, the results show that if one is interested only in predicting the indel flanking regions in protein sequences, then it would be preferable to use the proposed predictors instead of using HMMER 3.0 in view of the substantially superior

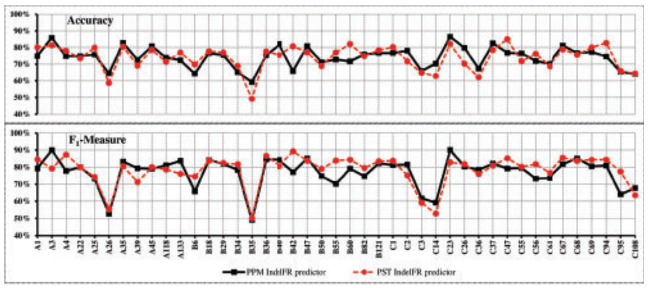

Fig. 6. Average accuracy and F_1-measure values of the proposed IndelFR predictors for the Superfamily set from the SABmark benchmark for the selected protein folds from different protein classes listed in the caption of Figure 5

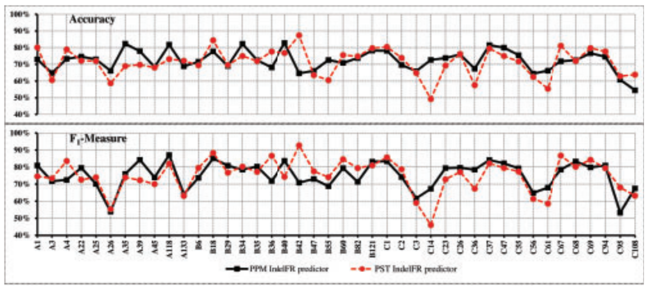

Fig. 7. Average accuracy and F_1-measure values of the proposed IndelFR predictors for the Twilight set from the SABmark benchmark for the selected protein folds from different protein classes listed in the caption of Figure 5

Table 1. Average accuracy and F_1 measure values for the proposed PPM and PST predictors with $D = 4$ and that obtained using HMMER 3.0 over all the protein sequences contained in the selected 11, 14 and 18 protein folds from the All-α protein, All-β protein and α and β protein (a/b) classes, respectively, for (a) IndelFR database, (b) SABmark-Superfamily set and (c) SABmark-Twilight set

		(a)		(b)		(c)	
	Protein classes	Accuracy (%)	F_1 (%)	Accuracy (%)	F_1 (%)	Accuracy (%)	F_1 (%)
PPM	All-α	91	92	76	79	74	76
IndelFR	All-β	86	85	74	80	75	78
Predictor	α and β	83	84	77	79	74	78
PST	All-α	88	89	76	79	71	75
IndelFR	All-β	84	86	76	83	75	81
Predictor	α and β	80	83	76	80	73	77
HMMER	All-α	43	49	57	64	59	66
Alignment	All-β	43	49	61	68	61	70
Software	α and β	46	56	59	69	58	68

performance of the former. It should be noted that if HMMER 3.0 is used for prediction, one would need as many pHMMs as the number of families in a given fold, while only one proposed predictor is needed for a given fold. Moreover, it should be noted that the proposed IndelFR predictors are built in a fully automated manner without using any prior assumption about the occurrence of mutations in the protein sequences, as in the case of scoring schemes. We anticipate that our study will not only enable future studies on the modeling of indel mutations, but will also open up new avenues for research concerning protein evolution, structures and functions.

Funding: This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada and in part by the Regroupement Stratgique en Microlectronique du Qubec (ReSMiQ).

Conflict of Interest: none declared.

REFERENCES

Andreeva,A. *et al.* (2008) Data growth and its impact on the scop database: new developments. *Nucleic Acids Res.*, **36**, D419–D425.

Begleiter,R. *et al.* (2004) On prediction using variable order markov models. *J. Artif. Intell. Res.*, **22**, 385–421.

Bejerano,G. and Yona,G. (2001) Variations on probabilistic suffix trees: statistical modeling and prediction of protein families. *Bioinformatics*, **17**, 23–43.

Benner,S.A. *et al.* (1993) Empirical and structural models for insertions and deletions in the divergent evolution of proteins. *J. Mol. Biol.*, **229**, 1065–1082.

Britten,R.J. *et al.* (2003) Majority of divergence between closely related dna samples is due to indels. *Proc. Natl Acad. Sci.*, **100**, 4661–4665.

Buhlmann,P. and Wyner,A.J. (1999) Variable length Markov chains. *The Annals of Statistics*, **27**, 480–513.

Chan,S.K. *et al.* (2007) Relationship between insertion/deletion (indel) frequency of proteins and essentiality. *BMC bioinformatics*, **8**, 227.

Chandonia,J.M. *et al.* (2004) The ASTRAL compendium in 2004. *Nucleic Acids Res.*, **32**, D189–D192.

Chen,J.Q. *et al.* (2009) Variation in the ratio of nucleotide substitution and indel rates across genomes in mammals and bacteria. *Mol. Biol. Evol.*, **26**, 1523–1531.

Cleary,J. and Witten,I. (1984) Data compression using adaptive coding and partial string matching. *IEEE Trans. Commun.*, **32**, 396–402.

Duval,A. and Hamelin,R. (2002) Mutations at coding repeat sequences in mismatch repair-deficient human cancers: toward a new concept of target genes for instability. *Cancer Res.*, **62**, 2447–2454.

Eddy,S.R. (1998) Profile hidden Markov models. *Bioinformatics*, **14**, 755–763.

Fawcett,T. (2006) An introduction to ROC analysis. *Pattern Recogn. Lett.*, **27**, 861–874.

Finn,R.D. *et al.* (2011) HMMER web server: interactive sequence similarity searching. *Nucleic Acids Res.*, **39**, W29–W37.

Grishin,N.V. (2001) Fold change in evolution of protein structures. *J. Struct. Biol.*, **134**, 167–185.

Henikoff,J.G. and Henikoff,S. (1996) Using substitution probabilities to improve position-specific scoring matrices. *Comput. Appl. Biosci.*, **12**, 135–143.

Hsing,M. and Cherkasov,A. (2008) Indel PDB: a database of structural insertions and deletions derived from sequence alignments of closely related proteins. *BMC bioinformatics*, **9**, 293.

Hughey,R. and Krogh,A. (1996) Hidden Markov models for sequence analysis: extension and analysis of the basic method. *Comput. Appl. Biosci.*, **12**, 95–107.

Jiang,H. and Blouin,C. (2007) Insertions and the emergence of novel protein structure: a structure-based phylogenetic study of insertions. *BMC Bioinformatics*, **8**, 444.

Karplus,K. *et al.* (1998) Hidden Markov models for detecting remote protein homologies. *Bioinformatics*, **14**, 846–856.

Krissinel,E. and Henrick,K. (2004) Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. *Acta cryst.*, **D60**, 2256–2268.

Krogh,A. *et al.* (1994) Hidden Markov models in computational biology: applications to protein modeling. *J. Mol. Biol.*, **235**, 1501–1531.

Moffat,A. (1990) Implementing the PPM data compression scheme. *IEEE Trans. Commun.*, **38**, 1917–1921.

Nisenson,M. *et al.* (2003) Towards behaviometric security systems: learning to identify a typist. *Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databasess, Cavtat-Dubrovnik, Croatia.* Vol. 2838, pp. 363–374.

Notredame,C. (2002) Recent progress in multiple sequence alignment: a survey. *Pharmacogenomics*, **3**, 131–144.

Punta,M. *et al.* (2012) The Pfam protein families database. *Nucleic Acids Res.*, **40**, D290–D301.

Ron,D. *et al.* (1996) The power of amnesia: Learning probabilistic automata with variable memory length. *Mach. Learn.*, **25**, 117–149.

Sonego,P. *et al.* (2008) ROC analysis: applications to the classification of biological sequences and 3d structures. *Brief. Bioinform.*, **9**, 198–209.

Tian,D. *et al.* (2008) Single-nucleotide mutation rate increases close to insertions/deletions in eukaryotes. *Nature*, **455**, 105–108.

Walle,I.V. *et al.* (2005) SABmark benchmark for sequence alignment that covers the entire known fold space. *Bioinformatics*, **21**, 1267–1268.

Willems,F.M.J. *et al.* (1995) The context-tree weighting method: basic properties. *IEEE Trans. Inform. Theory*, **41**, 653–664.

Yona,G. (2011) *Introduction to Computational Proteomics*. CRC Press, Boca Raton, FL.

Zhang,Z. *et al.* (2010) The combined effects of amino acid substitutions and indels on the evolution of structure within protein families. *PloS One*, **5**, e14316.

Zhang,Z. *et al.* (2011) Impact of indels on the flanking regions in structural domains. *Mol. Biol. Evol.*, **28**, 291–301.

Zhang,Z. *et al.* (2012) IndelFR: a database of indels in protein structures and their flanking regions. *Nucleic Acids Res.*, **40**, D512–D518.

Zhu,L. *et al.* (2009) Genomewide association between insertions/deletions and the nucleotide diversity in bacteria. *Mol. Biol. Evol.*, **26**, 2353–2361.