APPLICATION OF LINKER LENGTH AND LINKER LENGTH DEPENDENCY IN IDENTIFICATION OF PROTEIN DOMAINS

by

Ling Zhang

A DISSERTATION

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Doctor of Philosophy

Major: Statistics

Under the Supervision of Professors Stephen Kachman and Etsuko Moriyama

Lincoln, Nebraska

November, 2016

APPLICATION OF LINKER LENGTH AND LINKER LENGTH DEPENDENCY IN IDENTIFICATION OF PROTEIN DOMAINS

Ling Zhang, Ph.D.

University of Nebraska, 2016

Advisers: Stephen Kachman and Etsuko Moriyama

In protein sequences, domains are identified as conserved unit of structure, function and evolution. Identification of protein domains is important for the functional analysis of proteins. To achieve more sensitive and accurate domain discovery, we developed novel probabilistic modeling of multi-domain protein architectures. In our hidden Markov model (HMM) and Double-chain Markov model (DCMM), we incorporate not only domain dependency but also inter-domain linker information. The HMM using domain dependency with linker lengths (HMM-DL) successfully harnesses domain dependency and inter-domain linker lengths observed in the training dataset to predict divergent and non-overlapping domains on protein sequences. Moreover, a simulation procedure has been developed, which allows us to estimate false discovery rates and false positive rates to assess our approaches. We also present DCMM using domain dependency with linker lengths and linker-length dependency (DCMM-DLL) for the predictions of domains. By using DCMM, which has not been used in the field of bioinformatics, we are able to remove the limitation of the conditional independence assumption between observations and improve domain discovery performance. To increase the number of correct domain identifications, HMM-DL and DCMM-DLL were also extended to allow some overlapping domain identifications.

ACKNOWLEGEMENTS

I would first like to gratefully and sincerely thank my advisers: Dr. Steve Kachman, Dr. Shunpu Zhang and Dr. Etsuko Moriyama, for their guidance, understanding, patience and support. This dissertation could not have been finished without the help and support from them. I would like to express my sincere appreciation to my graduate committee, Dr. Bo Deng and Dr. Qi Zhang, for their generous suggestions and insightful comments. Especially, I would like to thank Dr. Anne Parkhurst for her support and research opportunity.

Thanks to all members of my research group, Dr. Steve Dunbar, Dr. Stephen Hartke, Dr. Brittney Keel, Neethu Shah, Ximeng Zheng, Yixiang Zhang, Na Li and Guofei Feng for helpful discussions about this work.

I would express my gratitude to my parents and parents-in-law whose support and constant encouragement helped me through the hard times of this program.

Finally, and most importantly, I am especially thankful to my son, Ryan, for being the ultimate reason for finishing this program, and to my wife, Wei, for her support, encouragement, quiet patience and love.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Biological background

The rapid development of high-throughput sequencing technologies has led to an overwhelming increase in sequence data. Proteins coded by genes are polymers composed of amino acids linked together through peptide bonds. They play a variety of critical roles in living cells (*e.g.*, antibody, enzyme, messenger, structural component and transport). However, understanding the specific biological functions of these proteins remains a challenge. Experimentally validating protein functions takes time and is highly expensive. In bacteria, for example, approximately 30% of genes lacks functional annotation (Meier *et al.*, 2013). In eukaryotes, over 40% of proteins encoded in their genomes are not assigned functions (Peña-Castillo and Hughes, 2007; Dhanyalakshmi *et al.*, 2016). Therefore, development of reliable and efficient computational approaches to infer protein functions from protein sequences is needed.

1.1.1 Protein domains

In protein sequences, domains are identified as conserved sequence regions, and are units of structure, function and evolution (Vogel, Bashton, *et al.*, 2004). Proteins typically consist of one or more domains. Each domain can fold independently to form a stable

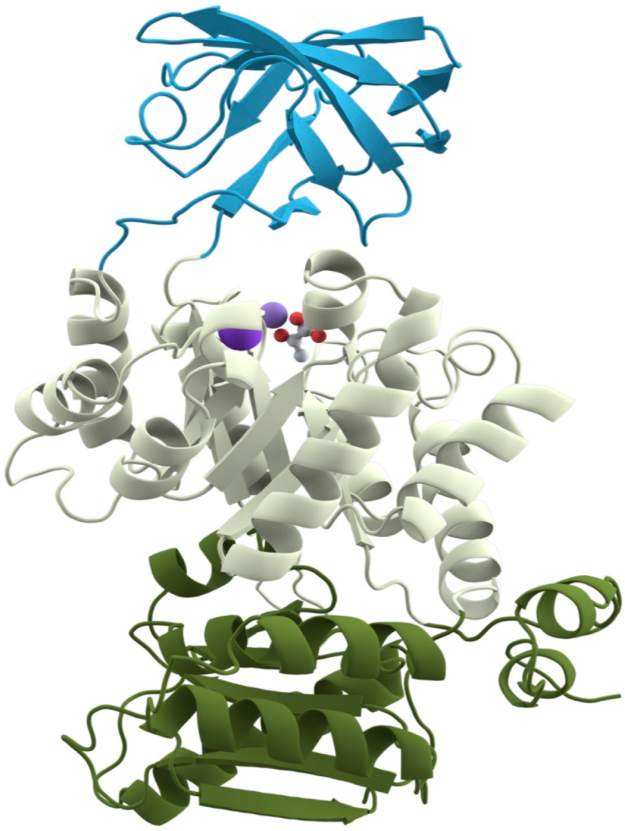

Figure 1.1: **The three domains of the rabbit pyruvate kinase (PDB: 1PKN)** (Rose *et al.*, 2015)**.** Each domain forms a compact three-dimensional structure and often can be independently folded (Larsen *et al.*, 2002).

three-dimensional structure (Fig. 1.1) (Kelley and Sternberg, 2015). A group of domains that have similar sequences, usually by descent from a common ancestral sequence, is known as a domain family (Punta *et al.*, 2011; Wilson *et al.*, 2009; Letunic *et al.*, 2015). Compared to protein databases, domain family databases grow more slowly (Ochoa, 2013). Thus, identification of protein domains can be efficiently used for functional classification and annotation.

In eukaryotes, 70% or more of proteins contain two or more domains (Apic *et al.*, 2001; Chothia *et al.*, 2003; Vogel, Bashton, *et al.*, 2004; Marsden *et al.*, 2006; Chothia and

Gough, 2009; Levitt, 2009). Such multi-domain proteins are thought to have evolved from a limited set of simpler single-domain proteins by combination of events such as duplication, divergence and recombination (Vogel, Bashton, *et al.*, 2004). Among multi-domain proteins, not only domain composition but also their orders in the protein sequences are often conserved (Vogel, Berzuini, *et al.*, 2004; Apic *et al.*, 2001). Therefore, understanding of the domain content and arrangement in proteins is very important for functional prediction and studies of evolution of protein functions.

1.1.2 Inter-domain linkers

In multi-domain proteins, the neighboring domains are connected by inter-domain linkers (Fig. 1.2). The inter-domain linkers are often unstructured. They play important roles in inter-domain interactions, functional regulation of proteins, protein stability, folding rates and domain-domain orientation (George and Heringa, 2002; Zhang *et al.*, 2009). Several properties of inter-domain linkers, such as the length, amino acid composition, hydrophobicity and glycosylation status, have been shown to affect protein stability and function (van Leeuwen *et al.*, 1997; Robinson and Sauer, 1998; Gustavsson *et al.*, 2001; Arai *et al.*, 2001; Zhang *et al.*, 2009; George and Heringa, 2002; Chen *et al.*, 2013).

1.1.3 Identification of protein domains

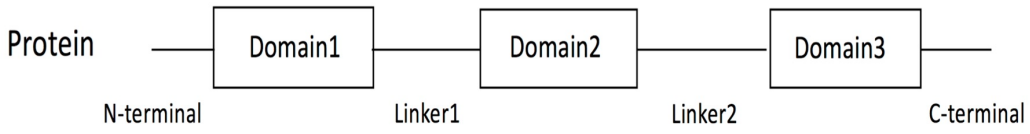

Figure 1.2: **Multi-domain protein architecture.** In this example, co-occurring domains 1, 2, and 3 are joined via linkers 1 and 2.

There are two types of widely used approaches to identify protein domains: structure-based approaches and sequence-based approaches. Structure-based approaches, such as SCOP (Fox *et al.*, 2014) and CATH (Sillitoe *et al.*, 2015), classify domains and proteins based on their 3D structures.

More often domains are identified based on sequence similarities. The simplest and the most used similarity search method is the Basic Local Alignment Search Tool (BLAST) (Altschul *et al.*, 1997; Camacho *et al.*, 2009; Altschul *et al.*, 1990). BLAST uses an approximation of the Smith-Waterman algorithm that attempts to determine local matches between the query and each of the database sequences (Altschul *et al.*, 1990). However, it is difficult to identify similar protein sequences using BLAST when sequence identities become below 30% (Park *et al.*, 1998; Skewes-Cox *et al.*, 2014). To detect such distantly related proteins, a profile was introduced to capture the position-specific information to descript the consensus of a multiple sequence alignment (Li *et al.*, 2012; Eddy, 1998; Madera and Gough, 2002; Altschul *et al.*, 1997; Gribskov *et al.*, 1987). The profile specifies the frequency of each amino acid in each column of a multiple sequence alignment (MSA) and sets positon-specific penalties for gaps in MSA. A successful

application of profile is PSI-BLAST, a position-specific-iterated version of BLAST. PSI-BLAST iteratively uses an position-specific score matrix (PSSM) to search similar proteins in database (Altschul *et al.*, 1997). Another widely used profile method is profile hidden Markov models (profile HMMs), full probabilistic representations of multiple alignments (Durbin *et al.*, 1998; Eddy, 1998). Performance of profile HMMs is typically better than PSI-BLAST in detecting distantly related sequences (Park *et al.*, 1998; Madera and Gough, 2002). Profile HMMs have been used to identify many protein families and domains. Examples of profile-HMM based protein family and domain databases include: Pfam (Finn *et al.*, 2013), SMART (Letunic *et al.*, 2015), SUPERFAMILY (Oates et al. 2015), InterPro (Mitchell *et al.*, 2014), PANTHER (Mi *et al.*, 2013), PRODOM (Bru *et al.*, 2005), and Gene3D (Lees *et al.*, 2014). More details on profile HMM will be given in section 1.2.2.

1.1.4 Gene Ontology (GO) terms

The key objective of this dissertation is to develop sensitive domain prediction approaches to improve functional annotations of proteins. The Gene Ontology (GO) initiative describes gene product attributes across all organisms by using a consistent and computable vocabularies (Ashburner *et al.*, 2000). Three sub-ontologies are defined by the GO for describing the properties of gene products: molecular function, cellular component and biological process. Molecular function describes the activities of a gene product at molecular level, cellular component describes the locations of a gene product or as a subcomponent of cellular component, and biological process describes sets of

molecular events or functions (Harris *et al.*, 2004). The GO is structured by a directed acyclic graph (DAG) to describe the parenthood relationship between terms. The children of any term represent more specific functions than the parents.

Several attempts have been made to associate domains or domain architectures with GO terms. Interpro2GO (including Pfam2GO, which is derived from InterPro2GO) manually maps the domains to appropriate GO terms (Burge *et al.*, 2012; Mitchell *et al.*, 2014). Several approaches have been developed to assign functions for domains or domain architectures automatically. Schug *et al* (2002) developed rule-based method for function-domain associations based on the intersection of GO terms assigned to proteins that contain domains at different similarity levels. GOTrees employed decision tree to associate the GO terms with Pfam domains (Hayete and Bienkowska, 2005). MultiPfam2GO uses a naïve Bayesian network to assign GO terms to domain sets (Forslund and Sonnhammer, 2008).

1.2 Probabilistic modelling of biological sequences

1.2.1 Hidden Markov model

A hidden Markov model (HMM) is a probabilistic model that describes a series of observations by unobserved (hidden) states (Fig. 1.3) (Rabiner, 1989). In computational biology, HMMs have been extensively employed, where the observations are strings of nucleotides forming DNA (or RNA) sequence or amino acids forming the primary sequence of a protein (Krogh *et al.*, 1994; Yoon, 2009). To simplify calculation of an

HMM, two Markov assumptions are applied: 1) the probability of a state depends only on the previous state and 2) the observation depends on the state that produced the observation and not on other observations or states (Jurafsky and Martin, 2014). These are called the first-order hidden Markov assumptions. A first-order HMM can be specified by following components (Rabiner, 1989; Ramage, 2007; Jurafsky and Martin, 2014):

1. The observed sequence $X = \{x_1, x_2, x_3 \ldots x_T\},\ x_t \in V$,

   where $V = \{v_1, v_2, \ldots v_n\}$ is a set of observed symbols and $t = 1 \ldots T$

2. The state sequence $Z = \{z_1, z_2, z_3 \ldots z_T\},\ z_t \in S$

   where, $S = \{s_1, s_2, \ldots s_m\}$ is a set of states and $t = 1 \ldots T$

3. The transition matrix $A\{a_{ij}\}$,

   where $a_{ij} = P(Z_{t+1} = s_j \mid z_t = s_i),\ 1 \le i, j \le m$

4. The emission probability $B = \{B_i(k)\}$,

   where $B_i(k) = P[v_k \text{ at time t} \mid z_t = s_i], 1 \le i \le m \text{ and } 1 \le k \le n$

5. The initial state $\pi_i = P(z_1 = s_i), 1 \le i \le m$

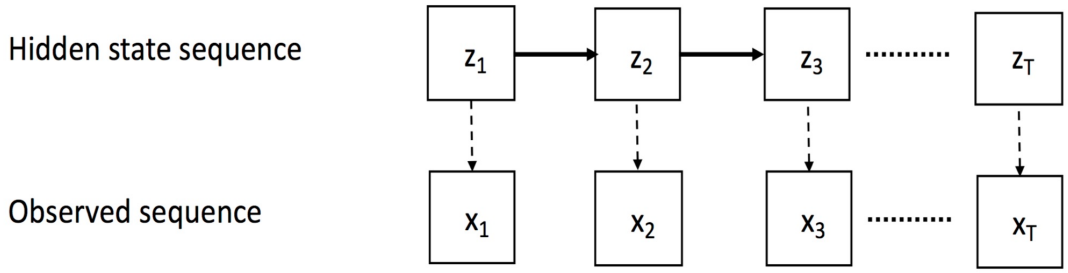

Figure 1.3: **A structure of a hidden Markov model.**

HMM $\lambda=(A,B,\pi)$ can be characterized by the following three fundamental problems (Rabiner, 1989; Jurafsky and Martin, 2014):

Likelihood: what is the probability of the observed sequence $x_1,x_2,x_3\ldots x_T$ given the model $\lambda=(A,B,\pi)$? That is, calculate $P(X|\lambda)$.

Decoding: given observation sequence X and HMM $\lambda=(A,B,\pi)$, what sequence of states has the largest probability. That is, find the state sequence Z^* such that $P(Z^*|X,\lambda)$ is maximized.

Learning: given some data, how do we "learn" a good HMM to describe the data? That is, given the topology of a HMM, and observed data, how we find the model which maximizes P(X).

The likelihood can be computed by the forward procedure (Rabiner, 1989). Let the forward variable to be

$$\alpha_t(i)=P(x_1,x_2,x_3\ldots x_t|z_t=s_i,\lambda) \tag{1.1}$$

$\alpha_t(i)$ can be solved as follows:

1. Initialization:

$$\alpha_1(i) = \pi_i B_i(x_1), 1 \le i \le m \tag{1.2}$$

2. Induction:

$$\alpha_t(j) = B_j(x_t) \sum_{i=1}^{m} \alpha_{t-1}(i) a_{ij}, \ 1 \le j \le m \quad 1 \le t \le T \tag{1.3}$$

3. Termination:

$$L = \sum_{i=1}^{N} \alpha_T(i) \tag{1.4}$$

In a similar way, we can define the backward variable as

$$\beta_t(i) = P(x_{t+1}, x_{t+2}, \ldots x_T \mid z_t = s_i, \lambda) \tag{1.5}$$

$\beta_t(i)$ can be solved inductively as follows:

1. Initialization:

$$\beta_T(j) = 1, \ 1 \le j \le m \tag{1.6}$$

2. Induction:

$$\beta_t(i) = \sum_{j=1}^{m} a_{ij} B_j(x_{t+1}) \beta_{t+1}(j) \tag{1.7}$$

The likelihood of observed sequence can be calculated by the forward-backward procedure:

$$L = \sum_{j=1}^{m} \alpha_t(j) \beta_t(j), \ t = 1, \ldots, T \tag{1.8}$$

For decoding question, the Viterbi algorithm is used. To describe the algorithm, the variable $\delta_t(i)$ is defined as follows(Jurafsky and Martin, 2014; Rabiner, 1989):

$$\delta_t(i) = \max_{z_1, z_2, \ldots z_{t-1}} P(z_1, z_2, \ldots z_{t-1}, z_t = s_i, x_1, x_2, \ldots x_t \mid \lambda) \tag{1.9}$$

For a given state s_j at time t+1, $\delta_{t+1}(j)$ is computed as

$$\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] B_j(x_{t+1}) \tag{1.10}$$

We use $\psi_t(j)$ as an array to track the best sequence path. The Viterbi recursion as follows is used to find the most likely state sequence as follows:

1. Initialization:

$$\delta_1(i) = \pi_i B_i(x_1), 1 \le i \le m \tag{1.11}$$

$$\psi_1(i) = 0 \tag{1.12}$$

2. Recursion:

$$\delta_t(j) = \max_{1 \le i \le m} [\delta_{t-1}(i) a_{ij}] B_j(x_t), \quad 2 \le t \le T \quad and \quad 1 \le j \le m \tag{1.13}$$

$$\psi_t(j) = \arg\max_{1 \le i \le m} [\delta_{t-1}(i) a_{ij}] B_j(x_t), \quad 2 \le t \le T \quad and \quad 1 \le j \le m \tag{1.14}$$

3. Termination:

   The best score:

$$P^* = \max_{1 \le i \le m} (\delta_T(i)) \tag{1.15}$$

   The start of backtrack:

$$z_T^* = \arg\max_{1 \le i \le m} (\delta_T(i)) \tag{1.16}$$

To solve learning problem, the variable is defined as follows:

$$\gamma_t(j) = P(z_t = s_i \mid X, \lambda) \tag{1.17}$$

Eq. 1.17 can be written in terms of the forward and backward variables:

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{m} \alpha_t(j)\beta_t(j)} \tag{1.18}$$

Let us define the probability $\xi_t(i,j)$ to be $P(z_t = s_i, z_{t+1} = s_j \mid X, \lambda)$.

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}B_j(x_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_t(i)a_{ij}B_j(x_{t+1})\beta_{t+1}(j)} \tag{1.19}$$

The transition probabilities can be estimated as follows:

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(j)} \tag{1.20}$$

The formula for computing emission probabilities is

$$\hat{B}_j(v_k) = \frac{\sum_{t=1 s.t. x_t = v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} \tag{1.21}$$

1.2.2 Profile hidden Markov model

Some positions of protein sequences in a domain family are more conserved, and other positions are more divergent. Thus simple pairwise alignments may not capture information required to find divergent query sequences (Durbin *et al.*, 1998; Eddy, 1998). It is thus desirable to use MSA to capture more positional conservation information (Fig. 1.4). A “profile”, a position-specific information from MSA, was introduced to represent

distantly related similar sequences (Eddy, 1998). HMMs provide a coherent theory to model profiles derived from MSAs. They are called profile HMMs (pHMMs).

When building a pHMM, a pHMM for MSA can be viewed first as HMM with one "match" state ("M") for each column (Fig. 1.4). Then the model can be extended to handle two types of gaps. One type of gap occurs when a portion of sequence that do not match anything in MSA, which is an insert state ("I"). The other type of gap is the region in MSA that is not present in the sequence, which is a delete state ("D"). Since no residue exists in the gap position in the sequence, a gap character '-' is used (Fig. 1.4). The full pHMM has the structure shown in Fig. 1.5 (Durbin *et al.*, 1998). Assigning columns to match states and insert states is needed. A simple rule is that when more than half of a column is occupied by gap characters, the position should be modeled as an insert state.

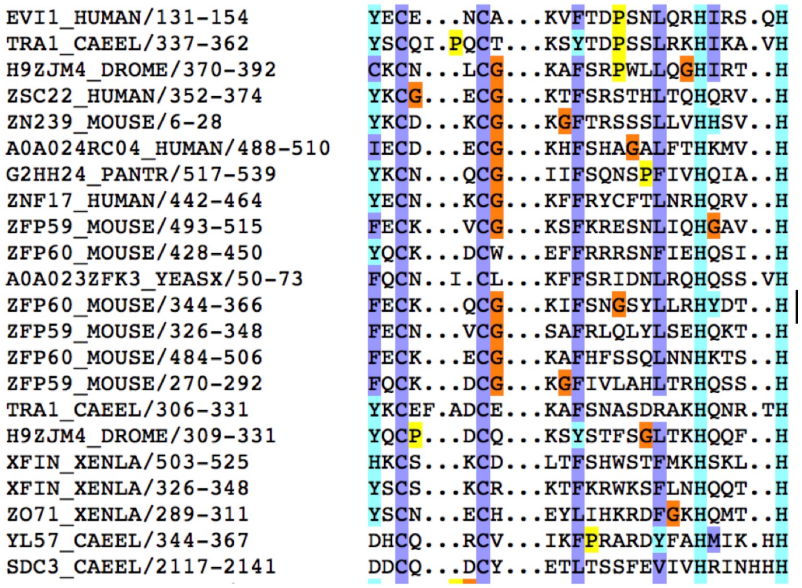

Figure 1.4: **Multiple sequence alignment of C2H2 zinc finger domains.**

In a full pHMM, the states are connected with arrows, representing transition probabilities (Fig. 1.5). For each match state M_i, it can transition to an insert state (I_i), a delete state (D_{i+1}), or the next match state (M_{i+1}). Each insert state I_i can transition to the next match state (M_{i+1}) or has a self-transition allowing multiple insertions of residues. The transitions between insert states and delete states are allowed although these situations are quite rare. Each delete state D_i can transition to an insert state (I_i), the next delete state (D_{i+1}), or the next match state (M_{i+1}).

Based on the probability of a given residue at the position in an MSA, there are emission probabilities associated with the match state in that position. The insert states also have emission distribution. However, the emission probabilities for insert states are set as the background probabilities. Since delete states do not emit any residues, delete states are silent states and have no emission probabilities.

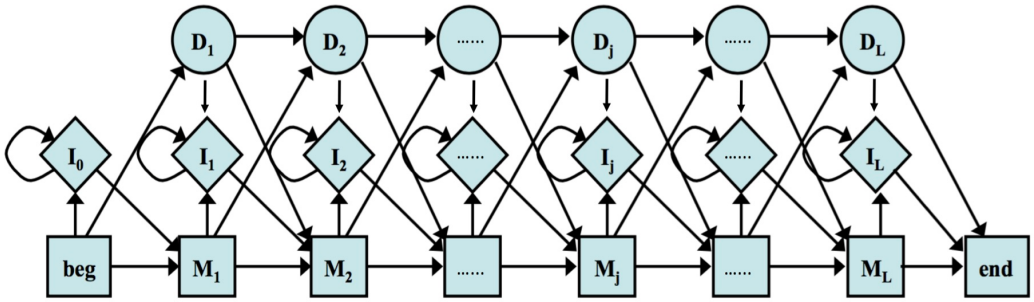

Figure 1.5: **The full structure of profile HMM(Eddy,2003).** M_i is the ith match state, I_i is the ith insert state, and D_i is the ith delete state. Delete states are silent and does not emit any residues.

Once pHMM is established, we need to estimate transitions probabilities between states, and emission probabilities for match and insert states. The transition probabilities are assigned as follows:

$$a_{kl} = \frac{A_{kl}}{\sum_{l'} A_{kl'}} \tag{1.22}$$

And the emission probabilities are computed as follows:

$$e_k(a) = \frac{E_k(a)}{\sum_{a'} E_k(a')} \tag{1.23}$$

It is possible that some transitions and emissions are never observed in an MSA. However, assigning these probabilities to be zero can be a problem in the future computation. For example, if only leucine is observed at a certain position of MSA, then the emission probability for leucine would be 1 and the probability would be zero for all other amino acids at this position. However, it is often found that valine tends to

substitute leucine without serious alteration of function. The probability of the query sequence with valine substituted for leucine at this position becomes zero (or minus infinity when using log-odds). Thus, this model cannot recognize any similar sequences if it does not have leucine at this position (Krogh, 1998). In order to avoid such over-fitting problems, a simple method is to use pseudocounts instead of the real counts (0). Simply using a constant as the pseudocount and add it to all the counts is based on a priori assumption that all amino acids are equally likely. However, it is not true since amino acids have different physical and chemical properties and some share similar properties. In order to include this prior knowledge, PSI-BLAST use substitution matrices (such as BLOSUM62) to calculate pseudocounts (J. G. Henikoff and S. Henikoff, 1996). However, there are two weakness in using substitution matrices to determine pseudocounts: 1) a substitution matrix does not deal with the context for each amino acid required for a specific position and 2) a substitution matrix treats amino acids the same when they have the same frequency, which ignores the actual number observed (Sjölander *et al.*, 1996). A nine-component Dirichlet mixtures can solve these problems by representing a variety of contexts and using actual number of observed residues to estimate probabilities of amino acids (for details see (Sjölander *et al.*, 1996)).

1.2.3 HMMER

One implementation of pHMM is HMMER, which is a program package used to build a pHMM from an MSA of protein or DNA sequences (Durbin *et al.*, 1998; Finn *et al.*, 2011; Eddy, 2010). The potential membership of a query sequence to a sequence family can be identified based on a significant match of a sequence to the pHMM. An example

of the C2H2 zinc finger domain that contains two highly conserved cysteines (C) and histidines (H) is shown in Fig. 1.4.

In HMMER, the log-odds score is used for asserting a sequence similarity. This is done by testing whether a sequence **x** is more likely to be a homolog of domain family (D) or more likely to be a random match (R). The log-odds score is calculated as follows (Eddy, 2003):

$$S = \log_2 \frac{P(x \mid D)}{P(x \mid R)} \tag{1.24}$$

The presence of a domain belonging to a domain family in the protein can be asserted if the log-odds score is above the given threshold (Finn *et al.*, 2011; Punta *et al.*, 2011). The thresholds used in HMMER are called the gathering thresholds (GAs). The gathering thresholds (GAs) are family-specific bit score, and they are empirically defined by the Pfam curators. GA is typically the higher of the lowest score obtained from the seed sequences (considered to be true positives) and the highest score of potential false (one of overlapping matches is false positive) positives (Finn *et al.*, 2011; WONG *et al.*, 2011; 2010; Srivastava *et al.*, 2007; Punta *et al.*, 2011). For each domain family, there are two GAs: a sequence GA and a domain GA. They are used to define the significance of a sequence and a domain hit, respectively. The domain bit score is the score computed by comparing the query protein sequence against a profile HMM. The sequence bit score is the sum of all bit scores contributed by all matches of a domain family on the protein query (Punta et al., 2011). A domain can be asserted on a protein if both its sequence and domain bit scores are equal to or larger than corresponding sequence and domain GAs,

respectively. Usually, the domain GAs are set to be lower than sequence GAs based on the assumption that a domain prediction is more likely to be correct when observing multiple copies of that domain in the same sequence. The GAs are conservative criterion due to the complete absence of false positives. Therefore, many domains can be still missed from highly divergent proteins.

To assess the statistical significance of log-odds scores when searching a database, the expectation value (E-value) is calculated and tested if the score is higher than the one obtained by chance from the database (Barrett *et al.*, 1997). The E-value of HMMER is calculated based on the stochastic model of Karlin/Altschul (Eddy, 2008):

$$P(S \geq t) = 1 - \exp[-e^{-\lambda(t-\mu)}] \tag{1.25}$$

S is a bit score, which is calculated in (1.24) and (1.25) gives the probability of observing random sequences with a score $S >$ t, where t is the score threshold. This random distribution is fitted by estimating μ and λ. μ and λ are summary statistics depend on the length and composition of the sequences and on the scoring system. $\mu = \frac{\log KNL}{\lambda}$, here N and L are the query sequence length and the database length, respectively. K and λ are statistical parameters estimated from scoring matrix and the amino acid composition of sequences.

Although HMMER is more sensitive for identifying distantly related similar sequences compared with BLAST, the pHMM implementations were much slower than BLAST. A set of heuristic filters have been developed to accelerate pHMM search in the new

version of HMMER (HMMER 3) (Eddy, 2011; 2010). The heuristic filter pipeline consists of the Multiple ungapped Segment Viterbi (MSV), Viterbi and Forward filters. The MSV algorithm is the first filter and the main speed heuristic in HMMER. The MSV algorithm is a simplified version of the Viterbi algorithm where the delete and insert states in pHMM are ignored (Eddy, 2011). It is used to calculate an optimal sum of ungapped high scoring alignment segments by employing Single Instruction Multiple Data (SIMD) to decrease the time requirements (Eddy, 2011; 2010). If the MSV score of the sequence is above a given threshold, the entire sequence passes onto the next filtering process, the Viterbi filter. The Viterbi algorithm is used to calculate the score of gapped alignment, which is more sensitive than the MSV score (Eddy, 2010). The sequences passing through the Viterbi filter arrive at the third, and final, filtering process, the Forward filter. It calculates scores by summing over those of all possible alignments (Eddy and Birney, 2001). We ran *hmmscan* using options --F1, --F2 and --F3 are used to control thresholds for passing MSV, Viterbi and Forward filters, respectively (Eddy, 2010).

1.2.4 Double-chain Markov model

As introduced in Section 1.2.1, there is an observation independence assumption for Markov model (Jurafsky and Martin, 2014):

$$P(x_i \mid z_1,\ldots,z_i,\ldots,z_T,x_1,\ldots,x_i,\ldots,x_T) = P(x_i \mid z_i)$$

Instead of the assumption of observation independence in HMM, Berchtold presented a full Markov model called the Double-chain Markov model (DCMM), which assumes a

Markov dependency between successive observations under hidden states (Berchtold, 2009; 2007; Fang *et al.*, 2010). Fig. 1.6 presents a first-order DCMM with the hidden state and observation sequences. Since the first observation (x_1) depends on the previous observation, the initial observation (x_0) without a corresponding hidden state is included in the model.

Similar to HMM, the first-order DCMM can be described by the following elements (Berchtold, 2009):

1. The observed sequence $X = \{x_1, x_2, x_3 \ldots x_T\}$, $x_t \in V$

   where $V = \{v_1, v_2, \ldots v_n\}$ is a set of observed symbols and $t = 1 \ldots T$

2. The state sequence $Z = \{z_1, z_2, z_3, \ldots z_T\}, z_t \in S$

   where, $S = \{s_1, s_2, \ldots s_m\}$ is a set of states and $t = 1 \ldots T$

3. The transition matrix $A\{a_{ij}\}$

   where $a_{ij} = P(z_{t+1} = s_j \mid z_t = s_i), 1 \le i, j \le m$

4. The emission matrix $B = \{B_j(k,l)\}$

   where $B_j(k,l) = P[x_t = v_l \mid x_{t-1} = v_k, z_t = s_i],\ 1 \le i \le m \quad and \quad 1 \le k \le n$

5. The initial state $\pi_i = P(z_1 = s_i), 1 \le i \le m$

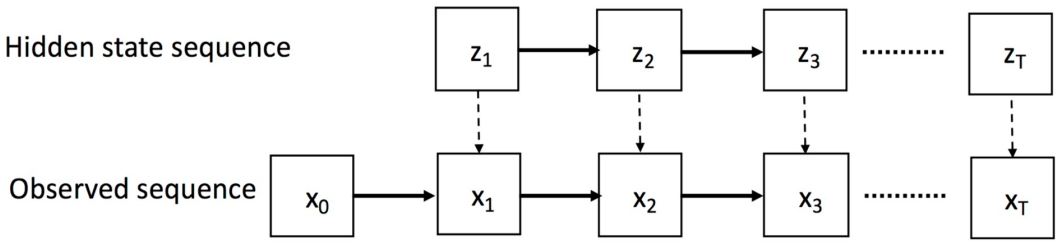

Figure 1.6: **The structure of a Double-chain Markov model.**

Three fundamental problems in section 1.3 can be applied for DCMM ($\lambda = (A, B, \pi)$) and be solved using similar algorithms (Berchtold, 2009).

The forward algorithm is used to answer the first question. The forward variable is defined as

$$\alpha_t(i) = P(x_1, x_2, x_3, \ldots x_t \mid z_t = s_i, \lambda) \tag{1.26}$$

$\alpha_t(i)$ can be solved as follows:

1. Initialization:

$$\alpha_1(i) = \pi_i B_i(x_0, x_1), 1 \le i \le m \tag{1.27}$$

2. Induction:

$$\alpha_t(j) = B_j(x_{t-1}, x_t) \sum_{i=1}^{m} \alpha_{t-1}(i) a_{ij}, \quad 1 \le j \le m \quad 1 \le t \le T \tag{1.28}$$

3. Termination:

$$L = \sum_{i=1}^{N} \alpha_T(i) \tag{1.29}$$

The backward variable used in the backward procedure is defined as

$$\beta_t(i) = P(x_{t+1}, x_{t+2}, \ldots x_T \mid x_t, z_t = s_i, \lambda) \tag{1.30}$$

For $t = T$, we have

$$\beta_T(j) = 1, 1 \le j \le m \tag{1.31}$$

For $t = 1,\ldots,T-1$,

$$\beta_t(i) = \sum_{j=1}^{m} a_{ij} B_j(x_t, x_{t+1}) \beta_{t+1}(j) \tag{1.32}$$

Thus, the likelihood of an observation given the DCMM can be written as

$$L = \sum_{j=1}^{m} \alpha_t(j) \beta_t(j), \;\; t = 1,\ldots,T \tag{1.33}$$

To answer second question, we use Viterbi algorithm. We define the variable:

$$\delta_t(i) = \max_{z_1, z_2, \ldots z_{t-1}} P(z_1, z_2, \ldots z_t = s_i, x_1, x_2, \ldots x_t \mid \lambda) \tag{1.34}$$

For a given state s_j at time $t+1$, $\delta_{t+1}(j)$ is computed as

$$\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] B_j(x_t, x_{t+1}) \tag{1.35}$$

We use $\psi_t(j)$ as an array to track the best sequence path. The Viterbi recursion is used to find the most likely state sequence as follows:

1. Initialization:

$$\delta_1(i) = \pi_i B_i(x_0, x_1), 1 \le i \le m \tag{1.36}$$

$$\psi_1(i) = 0 \tag{1.37}$$

2. Recursion:

$$\delta_t(j) = \max_{1 \le i \le m} [\delta_{t-1}(i) a_{ij}] B_j(x_{t-1}, x_t), \;\; 2 \le t \le T \quad and \quad 1 \le j \le m \tag{1.38}$$

$$\psi_t(j) = \arg\max_{1 \le i \le m} [\delta_{t-1}(i) a_{ij}] B_j(x_{t-1}, x_t), \;\; 2 \le t \le T \quad and \quad 1 \le j \le m \tag{1.39}$$

3. Termination:

The best score:

$$P^* = \max_{1 \le i \le m}(\delta_T(i)) \tag{1.40}$$

The start of backtrack:

$$z_T^* = \arg\max_{1 \le i \le m}(\delta_T(i)) \tag{1.41}$$

An Expectation-maximization (EM) algorithm, known as Baum-Welch algorithm, can be used to estimate the parameters (A, B, π) of DCMM. For $t = 1,\ldots T$, we have

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{m}\alpha_t(j)\beta_t(j)} \tag{1.42}$$

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}B_j(x_t, x_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_t(i)a_{ij}B_j(x_t, x_{t+1})\beta_{t+1}(j)} \tag{1.43}$$

The estimation formulas for A, B, π are shown as follows:

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(j)} \tag{1.44}$$

$$\hat{B}_j(v_k v_l) = \frac{\sum_{t=1 s.t. x_{t-1}=v_k, x_t=v_l}^{T}\gamma_t(j)}{\sum_{t=1 s.t. x_{t-1}=v_k}^{T}\gamma_t(j)} \tag{1.45}$$

While DCMM has been used successfully in e.g., behavioral analysis (Berchtold and Sackett, 2002; Chariatte *et al.*, 2008) and social network analysis (Malmgren *et al.*, 2009), it has not been applied yet in bioinformatics.

1.3 Objectives and organization of this dissertation

The key objective of this dissertation is to develop a Markov model that incorporates the information of domain structures including inter-domain linkers and to improve the sensitivity of protein domain identification. To our knowledge, there is no prior work using inter-domain linkers for domain prediction. Incorporating this unused protein sequence information, compared to the existing methods, the proposed approaches can be a more powerful tool for domain identification.

The organization of the remainder of the dissertation is as follows:

In Chapter 2, a HMM-based approach that incorporates the length of inter-domain linkers is developed. A shuffling procedure that allows estimation of the false discovery and false positive rates is introduced and used to assess the new method compared with the other methods. Results from searching domains in six model organisms showed that our method improved non-overlapping domain predictions compared with currently available context-based approaches (*e..g,* dPUC2, DAMA). Newly predicted domains were used to enhance the functional annotation of proteins, especially for those which are still left "unannotated" (no function is assigned) in *Plasmodium falciparum*.

Chapter 3 describes the Double-chain Markov model, which removes the limitation of independence assumption among observations in Hidden Markov model. DCMM incorporates linker-length dependency to improve the sensitivity of domain prediction. This method improved domain prediction in *C. elegans, D. melanogaster and H. sapiens*

through allowing some domain overlaps in the final prediction. We showed that this method compensated the weakness of the method described in Chapter 2.

In Chapter 4, we discuss these results and describe future work.

References

Altschul,S.F. *et al.* (1990) Basic local alignment search tool. *Journal of Molecular Biology*, **215**, 403–410.

Altschul,S.F. *et al.* (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, **25**, 3389–3402.

Apic,G. *et al.* (2001) Domain combinations in archaeal, eubacterial and eukaryotic proteomes. *Journal of Molecular Biology*, **310**, 311–325.

Arai,R. *et al.* (2001) Design of the linkers which effectively separate domains of a bifunctional fusion protein. *Protein Eng.*, **14**, 529–532.

Ashburner,M. *et al.* (2000) Gene Ontology: tool for the unification of biology. *Nature Genetics*, **25**, 25–29.

Barrett,C. *et al.* (1997) Scoring hidden Markov models. *Comput Appl Biosci*, **13**, 191–199.

Berchtold,A. (2007) High-order extensions of the Double Chain Markov Model. *Stochastic Models*, **18**, 193–227.

Berchtold,A. (2009) The double chain markov model. *Communications in Statistics - Theory and Methods*, **28**, 2569–2589.

Berchtold,A. and Sackett,G. (2002) Markovian models for the developmental study of social behavior. *American Journal of Primatology*, **58**, 149–167.

Bru,C. *et al.* (2005) The ProDom database of protein domain families: more emphasis on 3D. *Nucleic Acids Research*, **33**, D212–5.

Burge,S. *et al.* (2012) Manual GO annotation of predictive protein signatures: the

InterPro approach to GO curation. *Database*, **2012**, bar068–bar068.

Camacho,C. *et al.* (2009) BLAST+: architecture and applications. *BMC Bioinformatics*, **10**, 421.

Chariatte,V. *et al.* (2008) Missed Appointments in an Outpatient Clinic for Adolescents, an Approach to Predict the Risk of Missing. *Journal of Adolescent Health*, **43**, 38–45.

Chen,X. *et al.* (2013) Fusion protein linkers: Property, design and functionality. *Advanced Drug Delivery Reviews*, **65**, 1357–1369.

Chothia,C. and Gough,J. (2009) Genomic and structural aspects of protein evolution. *Biochemical Journal*, **419**, 15–28.

Chothia,C. *et al.* (2003) Evolution of the Protein Repertoire. *Science*, **300**, 1701–1703.

Dhanyalakshmi,K.H. *et al.* (2016) An Approach to Function Annotation for Proteins of Unknown Function (PUFs) in the Transcriptome of Indian Mulberry. *PLOS ONE*, **11**, e0151323.

Durbin,R. *et al.* (1998) Biological sequence analysis: probabilistic models of proteins and nucleic acids.

Eddy,S. (2010) HMMER user's guide version 3.0 Department of Mathematics.

Eddy,S. HMMER Users Guide, October 2003.

Eddy,S.R. (2008) A Probabilistic Model of Local Sequence Alignment That Simplifies Statistical Significance Estimation. *PLoS Comput Biol*, **4**, e1000069.

Eddy,S.R. (2011) Accelerated Profile HMM Searches. *PLoS Comput Biol*, **7**, e1002195.

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

Eddy,S.R. and Birney,E. (2001) HMMER User's Guide: Biological sequence analysis

using profile Hidden Markov Models, version 2.2 Washington University School of Medicine.

Fang,X. *et al.* (2010) Sequence Comparison using Multi-Order Markov Chains. IEEE, pp. 1–5.

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

Finn,R.D. *et al.* (2013) Pfam: the protein families database. *Nucleic Acids Research*, **42**, gkt1223–D230.

Forslund,K. and Sonnhammer,E.L.L. (2008) Predicting protein function from domain content. *Bioinformatics*, **24**, 1681–1687.

Fox,N.K. *et al.* (2014) SCOPe: Structural Classification of Proteins--extended, integrating SCOP and ASTRAL data and classification of new structures. *Nucleic Acids Research*, **42**, D304–9.

George,R.A. and Heringa,J. (2002) An analysis of protein domain linkers: their classification and role in protein folding. *Protein Eng.*, **15**, 871–879.

Gribskov,M. *et al.* (1987) Profile analysis: detection of distantly related proteins. *Proceedings of the National Academy of Sciences*, **84**, 4355–4358.

Gustavsson,M. *et al.* (2001) Stable linker peptides for a cellulose-binding domain-lipase fusion protein expressed in Pichia pastoris. *Protein Eng.*, **14**, 711–715.

Harris,M.A. *et al.* (2004) The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Research*, **32**, D258–61.

Hayete,B. and Bienkowska,J.R. (2005) Gotrees: predicting go associations from protein domain composition using decision trees. *Pac Symp Biocomput*, 127–138.

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

Jurafsky,D. and Martin,J.H. (2014) Speech and language processing.

Kelley,L.A. and Sternberg,M.J. (2015) Partial protein domains: evolutionary insights and bioinformatics challenges. *Genome Biol.*, **16**, 100.

Krogh,A. (1998) An introduction to hidden Markov models for biological sequences. In, *Computational Methods in Molecular Biology*, New Comprehensive Biochemistry. Elsevier, pp. 45–63.

Krogh,A. *et al.* (1994) Hidden Markov Models in Computational Biology: Applications to Protein Modeling. *Journal of Molecular Biology*, **235**, 1501–1531.

Larsen,T.M. *et al.* (2002) Structure of Rabbit Muscle Pyruvate Kinase Complexed with Mn2+, K+, and Pyruvate. *Biochemistry*, **33**, 6301–6309.

Lees,J.G. *et al.* (2014) Gene3D: Multi-domain annotations for protein sequence and comparative genome analysis. *Nucleic Acids Research*, **42**, D240–5.

Letunic,I. *et al.* (2015) SMART: recent updates, new developments and status in 2015. *Nucleic Acids Research*, **43**, D257–60.

Levitt,M. (2009) Nature of the protein universe. *Proc. Natl. Acad. Sci. U.S.A.*, **106**, 11079–11084.

Li,W. *et al.* (2012) PSI-Search: iterative HOE-reduced profile SSEARCH searching. *Bioinformatics*, **28**, 1650–1651.

Madera,M. and Gough,J. (2002) A comparison of profile hidden Markov model procedures for remote homology detection. *Nucleic Acids Research*, **30**, 4321–4328.

Malmgren,R.D. *et al.* (2009) Characterizing individual communication patterns ACM,

New York, New York, USA.

Marsden,R.L. *et al.* (2006) Comprehensive genome analysis of 203 genomes provides structural genomics with new insights into protein family space. *Nucleic Acids Research*, **34**, 1066–1080.

Meier,M. *et al.* (2013) Proteome-wide protein interaction measurements of bacterial proteins of unknown function. *Proc. Natl. Acad. Sci. U.S.A.*, **110**, 477–482.

Mi,H. *et al.* (2013) PANTHER in 2013: modeling the evolution of gene function, and other gene attributes, in the context of phylogenetic trees. *Nucleic Acids Research*, **41**, D377–86.

Mitchell,A. *et al.* (2014) The InterPro protein families database: the classification resource after 15 years. *Nucleic Acids Research*, **43**, gku1243–D221.

Ochoa,A. (2013) Protein domain prediction using context statistics, the false discovery rate, and comparative genomics, with application to Plasmodium falciparum.

Park,J. *et al.* (1998) Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods. *Journal of Molecular Biology*, **284**, 1201–1210.

Peña-Castillo,L. and Hughes,T.R. (2007) Why Are There Still Over 1000 Uncharacterized Yeast Genes? *Genetics*, **176**, 7–14.

Punta,M. *et al.* (2011) The Pfam protein families database. *Nucleic Acids Research*, **40**, D290–D301.

Rabiner,L.R. (1989) A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, **77**, 257–286.

Ramage,D. (2007) Hidden Markov models fundamentals. *Lecture Notes http://cs229*

stanford edu/section/ ….

Robinson,C.R. and Sauer,R.T. (1998) Optimizing the stability of single-chain proteins by linker length and composition mutagenesis. *Proceedings of the National Academy of Sciences*, **95**, 5929–5934.

Rose,P.W. *et al.* (2015) The RCSB Protein Data Bank: views of structural biology for basic and applied research and education. *Nucleic Acids Research*, **43**, D345–56.

Schug,J. *et al.* (2002) Predicting gene ontology functions from ProDom and CDD protein domains. *Genome Res.*, **12**, 648–655.

Sillitoe,I. *et al.* (2015) CATH: comprehensive structural and functional annotations for genome sequences. *Nucleic Acids Research*, **43**, D376–81.

Sjölander,K. *et al.* (1996) Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology. *Comput Appl Biosci*, **12**, 327–345.

Skewes-Cox,P. *et al.* (2014) Profile Hidden Markov Models for the Detection of Viruses within Metagenomic Sequence Data. *PLOS ONE*, **9**, e105067.

Srivastava,P.K. *et al.* (2007) HMM-ModE – Improved classification using profile hidden Markov models by optimising the discrimination threshold and modifying emission probabilities with negative training sequences. *BMC Bioinformatics*, **8**, 104.

van Leeuwen,H.C. *et al.* (1997) Linker length and composition influence the flexibility of Oct-1 DNA binding. *The EMBO Journal*, **16**, 2043–2053.

Vogel,C., Bashton,M., *et al.* (2004) Structure, function and evolution of multidomain proteins. *Current Opinion in Structural Biology*, **14**, 208–216.

Vogel,C., Berzuini,C., *et al.* (2004) Supra-domains: Evolutionary Units Larger than Single Protein Domains. *Journal of Molecular Biology*, **336**, 809–823.

Wilson,D. *et al.* (2009) SUPERFAMILY--sophisticated comparative genomics, data mining, visualization and phylogeny. *Nucleic Acids Research*, **37**, D380–6.

WONG,W.-C. *et al.* (2010) More Than 1,001 Problems with Protein Domain Databases: Transmembrane Regions, Signal Peptides and the Issue of Sequence Homology. *PLoS Comput Biol*, **6**, e1000867.

WONG,W.-C. *et al.* (2011) THE JANUS-FACED E-VALUES OF HMMER2: EXTREME VALUE DISTRIBUTION OR LOGISTIC FUNCTION? *J. Bioinform. Comput. Biol.*, **09**, 179–206.

Yoon,B.-J. (2009) Hidden Markov Models and their Applications in Biological Sequence Analysis. *CG*, **10**, 402–415.

Zhang,J. *et al.* (2009) Design and optimization of a linker for fusion protein construction. *Progress in Natural Science*, **19**, 1197–1200.

Chapter 2

Application of Inter-domain Linker in Identification of Protein Domains

2.1 Introduction

Proteins often contain multiple conserved domains, where domains are considered to be the structural, functional, and evolutionary units. The combinations of domains endow the proteins with specific functions. Identification of domains is, therefore, important in annotation of protein structures and functions.

Several methods have been developed to identify domains based on structural classes or conserved sequences (Murzin *et al.*, 1995; Finn *et al.*, 2007). The majority of the methods used to identify domains in protein sequences are based on profile hidden Markov models (pHMMs) (Eddy, 1998). The Pfam database, for example, includes pHMMs for protein and domain families (Punta *et al.*, 2011). Domains can be identified from a protein sequence by performing pHMM search using such as HMMER3 against the Pfam v27 (downloaded from http://pfam.sanger.ac.uk) (Finn *et al.*, 2011). For an amino acid sequence x (A_x), the bit score S can be calculated as: $S = \log_2 \frac{P(A_x|M_D)}{P(A_x|R)}$, where $P(A_x|M_D)$ is the probability of the target sequence (A_x) given a domain family model (M_D) and $P(A_x|R)$ is the probability of the target sequence (A_x) given a random model (R). The presence of a domain belonging to a domain family in the protein can be asserted if the bit score (or the E-value calculated from it) is above the given threshold.

The thresholds used with Pfam are conservative due to the complete absence of false positives; many highly divergent domains can be missed from the proteins.

When domains are identified using a profile HMM-based method, the information from domains coexisting in the same protein is not taken into account. However, the observed number of domain combinations is much smaller than the number of all possible domain combinations (Apic *et al.*, 2001) indicating that there are some constrains in which domains co-exist in proteins. Using domain combination information, therefore, is expected to improve accuracy and sensitivity of domain identification. Several methods have been developed to take advantage of such information in domain discovery. For the Co-Occurrence Domain Discovery (CODD), a list of domain pairs, "conditionally dependent pairs" (CDP), showing statistically significantly high co-occurrence was generated (Terrapon *et al.*, 2009). A set of potential Pfam domains is identified on a protein using a lower (more permissive) threshold. Then the presence of a potential domain is considered *certified* if it, along with another non-overlapping domain in the protein, forms a pair that belongs to the list of CDP. dPUC (Domain Prediction Using Context) is a graph-theoretic framework where domains are nodes and edges connecting domains are weighted based on domain context scores (Ochoa, Llinás, and Singh, 2011a). dPUC solves the combinatorial optimization problem with integer linear programming. dPUC was shown to outperform CODD with the following advantages: 1) unobserved domain pairs are penalized; 2) log-odds scores allow a pair of weak domains to be boosted up; and 3) repetitive domains can be identified. dPUC2, an update of dPUC, uses directional domain pair context scores (Ochoa, 2013). The directional domain preference

has been observed in multi-domain architectures (Apic *et al.*, 2001; Vogel *et al.*, 2004), and dPUC2 takes such preferences into account. More recently a multi-objective optimization approach is used in DAMA (Domain Annotation by a Multi-objective Approach) (Bernardes *et al.*, 2016). It incorporates, not only domain pairs, but multi-domain co-occurrence in improving domain prediction. DAMA was reported to show a better performance compared with both CODD and dPUC.

Although CODD, dPUC, and DAMA incorporate domain co-occurrence information to select most likely domain pairs or sets along the protein sequence, they do not consider the actual order and adjacency of domains along the sequence. dPUC2 considers directional domain contexts. However, dPUC2 still does not incorporate the adjacency information of domain pairs. Coin *et al.* (2003) used a Markov chain model to incorporate not only domain co-occurrence information but also their orders in improving domain detection. Let $\mathrm{D} = \mathrm{D}_1 \ldots \mathrm{D}_n$ be a domain sequence in a protein. For simplicity, we assume that domain regions are not overlapped in the protein. Given the first-order Markov-chain model (*MC*) and the protein sequence (*A*), the probability of the domain sequence *D* is defined to be:

$$P(D|A,MC) = \frac{P(D,A,MC)}{P(A,MC)} = \frac{P(A|D,MC)P(D,MC)}{P(A|MC)P(MC)} = \frac{P(A|D)}{P(A|MC)}P(D|MC) \qquad (2.1)$$

The goal is to find a *D* that maximizes $P(D|A,MC)$. Since the model *MC* considers only domain dependency, the protein sequence *A* is conditionally independent of *MC* given *D*.

Therefore $P(A|MC)$ is a constant, and replacing it with another constant, $P(A|R)$, which is the probability of the protein sequence *A* given a random model (*R*), does not affect searching *D* with the highest probability. Using the prior probability of domain, $P(D_i)$, (Eq. 2.1) can be expressed as:

$$P(D|A,MC) \propto (\prod_i \frac{P(A_i|D_i)}{P(A_i|R)} P(D_i)) \times \prod_i \frac{P(D_i|D_{i-1},MC)}{P(D_i)} \quad (2.2)$$

where i=1,…,n and A_i is the amino acid sequence of D_i.. Then,

$$\log_2 P(D|A,MC) \propto \sum_i (\log_2 \frac{P(A_i|D_i)}{P(A_i|R)} - \log_2 (\frac{1}{P(D_i)})) + \sum_i \log_2 \frac{P(D_i|D_{i-1},MC)}{P(D_i)} \quad (2.3)$$

Here, $H(D_i) = \log_2 \frac{P(A_i|D_i)}{P(A_i|R)}$, $H(D_i)$ is the bit score of the domain D_i, and $T(D_i) = \log_2 \frac{1}{P(D_i)}$, $T(D_i)$ is the score threshold. Then (Eq. 2.3) can be rewritten as:

$$\log_2 \propto \sum_i (H(D_i) - T(D_i)) + \sum_i \log_2 \frac{P(D_i|D_{i-1},MC)}{P(D_i)} \quad (2.4)$$

Note that when domain context is not considered, a domain can be detected on a protein simply if $H(D_i) > T(D_i)$.

Note that the implementation of program in Coin's method is not available. We therefore implemented this method, which works with HMMER3 and Pfam versions 27 or higher. We call this method "MC-DD" (Markov-chain model for domain dependency).

In addition to domains, the amino acid sequences between domains (linkers) also play important roles in, *e.g.*, protein stability, folding rates, domain-domain orientation, and functional regulation (Gokhale, 2000; George and Heringa, 2002; Wriggers *et al.*, 2005). Therefore, it is highly likely that properties of these inter-domain linkers are constrained depending on the types of neighboring domains and such linker information can be utilized to enhance domain detection performance. In this study, we developed a novel domain detection method, Hidden Markov Model using Domain dependency and Linker lengths (HMM-DL). It uses a first-order HMM to incorporate the information of domain dependency and linker lengths for sensitive domain discovery. We applied HMM-DL to proteomes from six representative organisms and demonstrated its improved sensitivity in domain detection compared with other methods. We also presented example cases where newly identified domains contributed in functional annotation of proteins.

2.2 Materials and Methods

2.2.1 Datasets

Training dataset. We downloaded the UniRef50 protein dataset from the Universal Protein Resource (UniProt) protein database (The UniProt Consortium, 2015). Each cluster of UniRef50 contains sequences that have at least 50% identity to and 80%

overlap with the longest sequence, UniRef50 included 13,597,642 protein sequences. Using hmmscan from the HMMER3 software package (hmmer 3.1b2) and the Pfam database (rel. 27), we identified 10,501,358 domains belonging to 14,828 Pfam domain families. It included 3,644,227 domain pairs (and linkers) in 11,411 types of domain family pairs. Linkers were grouped into four categories based on their lengths: short, medium, long, and extremely long, in approximately equal numbers (~910,000 linkers in each category). Length distributions of linkers for the entire dataset as well as for each representative organism are shown in Supplementary Fig. A1 and Fig. A2. In order to examine whether linker lengths are conserved or not for the specific domain pairs, we gathered the specific domain pairs which were observed 100 or more times in training dataset. For each type of domain pair, the occurrence of each length category was counted. If linker lengths have no biased distribution depending on the specific domain pairs, their frequencies should tend to be around the background probability, ~0.25 (the linkers are divided into four equal length categories). Our observations showed it is not the case (Fig. A3). It suggests that linker lengths are constrained depending on the type of domain pairs.

Proteomes. The proteomes of five representative organisms were downloaded from the Reference Proteomes database (http://www.ebi.ac.uk/reference_proteomes) as follows: *Escherichia coli* (UP000000625, taxon ID: 83333; 4,305 proteins), *Saccharomyces cerevisiae* (UP000002311, taxon ID: 559292; 6,720 proteins), *Caenorhabditis elegans* (UP000001940, taxon ID: 6239; 20,274 proteins), *Drosophila melanogaster* (UP000000803, taxon ID: 7227; 13,674 proteins), and *Homo sapiens* (UP000005640,

taxon ID: 9606; 20,882 proteins). The proteome of *Plasmodium falciparum* (5,542 proteins) was download from the *Plasmodium* Genomics Resource (PlasmoDB, release 26) (Bahl, 2003). The distributions of the number of domains per protein (standard Pfam) of *E. coli, S. cerevisiae, C. elegans, D. melanogaster, H. sapiens*, and *P. falciparum* is shown in Supplementary Fig. A4, where the most of the proteins have no domain annotation or one annotation.

2.2.2 Approach

In multi-domain proteins, domains are connected by inter-domain "linker" sequences (Fig. 2.1A). We propose HMM-DL, a domain detection method that uses a first order HMM to incorporate the information of domain dependency and linker length (Fig. 2.1B). Similar to previous methods, our approach starts with identifying all potential domain regions from a protein sequence (see Fig. 2.2 for an illustration). This is done, for example, by identifying all potential domain regions from a given protein sequence by using HMMER3 with a permissive (higher) E-value threshold against the Pfam database. Let $\mathbf{d} = \{d_1, d_2 \ldots d_m\}$ be the set of all the candidate domains with corresponding amino acid sequences $a_1, a_2 \ldots a_m$, where domains are numbered based on their ending amino acid positions in the protein sequence. Here, a candidate domain, d_j, is defined by the Pfam domain family (given as a profile HMM) and the position on the protein. . Let $H(d_j)$ be the HMMER domain score of domain d_j, and $T(d_j)$ be the score threshold (see section 1.2.3 for the score threshold specific to each domain family used in Pfam). Let

$D = D_1 \ldots D_n$ be a domain sequence without overlap, where $\{D_1, D_2 \ldots D_n\}$ is the subset of the candidate domain set **d** keeping the same numbering order as in **d**. Let K = $K_1 \ldots K_{n-1}$ be a linker-length sequence between domains in a domain sequence (D) (see Fig. 2.2).

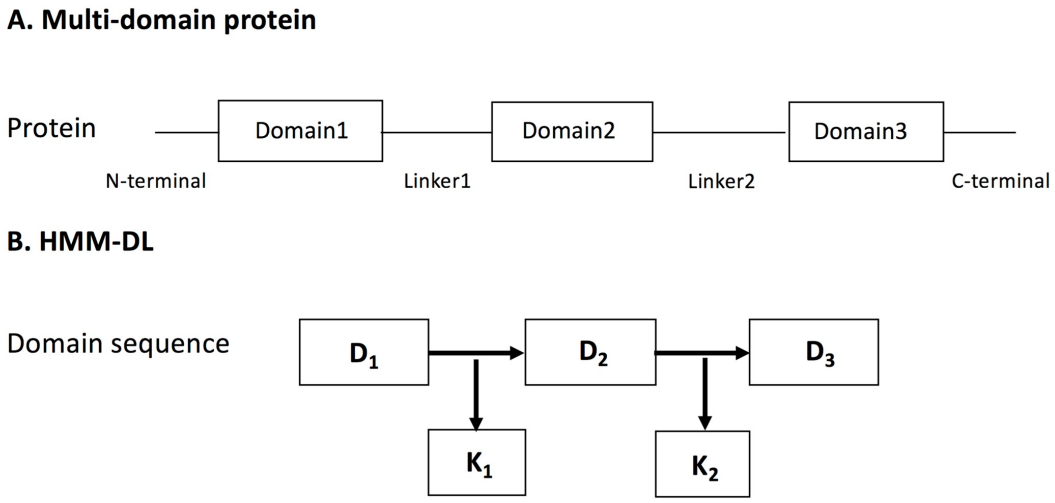

Figure 2.1: **Multi-domain protein architecture (A) and HMM-DL representations (B).** In a given protein, domains are numbered from N- to C-terminals based on the end position. D_i is the profile HMM for the i-th domain. K_{i-1} is the linker length distribution between two domains D_i and D_{i+1}, which is emitted from the domain pair.

Our goal in detecting domains is to seek the most likely domain sequence D^* with appropriate linkers among them given a protein sequence (A) and a first-order HMM model (HM):

$$D^* = \arg\max_D P(D, K \mid A, HM) \tag{2.5}$$

where

$$P(D,K,|A,HM)=\frac{P(D,K,A,HM)}{P(A,HM)}=\frac{P(A|D,K,HM)}{P(A,HM)}P(D,K,HM)$$
$$=\frac{P(A|D,K,HM)}{P(A|HM)P(HM)}P(D,K|HM)P(HM)=\frac{P(A|D,K,HM)}{P(A|HM)}P(D,K|HM) \quad (2.6)$$

Since given D, K's are known and A is independent of the HM model, $P(A|D,K,HM)=P(A|D)$. Thus (Eq. 2.6) can be written as:

$$P(D,K|A,HM)=\frac{P(A|D)}{P(A|HM)}P(D,K|HM) \quad (2.7)$$

Since $P(A|HM)$ is a constant, as before, by replacing it with another constant, $P(A|R)$, it does not affect searching *D* with the highest probability::

$$P(D,K|A,HM)\propto\frac{P(A|D)}{P(A|R)}P(D,K|HM)=(\prod_i\frac{P(A_i|D_i)}{P(A_i|R)})\times P(D,K|HM) \quad (2.8)$$

Here,

$$P(D,K|HM)=\prod_i P(D_i|D_{i-1})\cdot P(K_{i-1}|D_{i-1},D_i) \quad (2.9)$$

where i=1,…,n. We denote D_0 as the 'begin' state, and set $P(D_1|D_0)=1$ and $P(K_0|D_0,D_1)=1$. Then (2.8) can be rewritten as:

$$P(D,K|A,HM)\propto(\prod_i\frac{P(A_i|D_i)}{P(A_i|R)})\times\prod_i(P(D_i|D_{i-1})\cdot P(K_{i-1}|D_{i-1},D_i))$$
$$=(\prod_i\frac{P(A_i|D_i)}{P(A_i|R)}\cdot P(D_i)\cdot P(K_{i-1}))\times\prod_i(\frac{P(D_i|D_{i-1})}{P(D_i)}\cdot\frac{P(K_{i-1}|D_{i-1},D_i)}{P(K_{i-1})}) \quad (2.10)$$

It is equivalent to:

$$\log_2 P(D,K \mid A,HM) \propto \sum_i (\log_2 \frac{P(A_i \mid D_i)}{P(A_i \mid R)} - \log_2 \frac{1}{P(D_i)P(K_{i-1})}) + \sum_i \log_2 \frac{P(D_i \mid D_{i-t})}{P(D_i)} + \sum_i \log_2 \frac{P(K_{i-1} \mid D_{i-1}, D_i)}{P(K_{i-1})} \quad (2.11)$$

By replacing $\log_2 \frac{1}{P(D_i)P(K_{i-1})}$ with $T(D_i)$, we have:

$$\log_2 P(D,K \mid A,HM) \propto \sum_i (\log_2 \frac{P(A_i \mid D_i}{P(A_i \mid R)} - T(D_i)) + \sum_i \log_2 \frac{P(D_i \mid D_{i-1})}{P(D_i)} + \sum_i \log_2 \frac{P(K_{i-1} \mid D_{i-1}, D_i)}{P(K_{i-1})} \quad (2.12)$$

Finally, we define the score for the domain sequence D as:

$$S(D) = H(D_1) - T(D_1) + \sum_{i=2}^{n} (H(D_i) - T(D_i) + C(D_i \mid D_{i-1}) + L(D_{i-1}, D_i)) \quad (2.13)$$

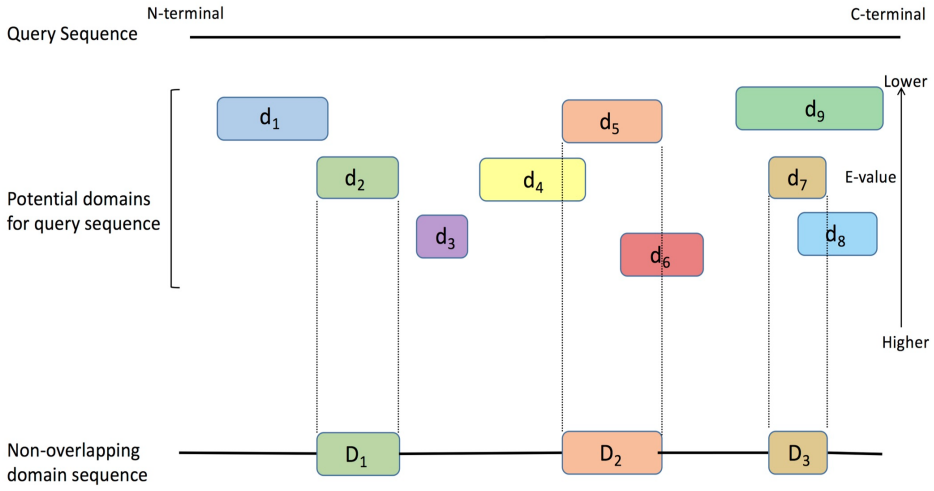

Figure 2.2: **Potential domains for a query sequence as input for domain prediction tools.** The potential domains for a query sequence are obtained through running hmmscan with permissive E-values. The potential domains are ranked from bottom to top by their decreasing E-values. HMM-DL and other tools are applied to potential domains to obtain the most likely domain sequence without overlaps.

where $H(D_i) = \log_2 \frac{P(A_i \mid D_i)}{P(A_i \mid R)}$ is the bit score for the domain D_i from HMMER and $T(D_i)$ is the score threshold. $C(D_i \mid D_{i-1})$, the domain dependency score, and $L(D_{i-1}, D_i)$, the linker emission score, are described next.

The domain dependency score, $C(D_i \mid D_{i-1})$, is defined as $\log_2 \frac{P(D_i \mid D_{i-1})}{P(D_i)}$ to match the HMMER bit scores. $P(D_i \mid D_{i-1})$ and $P(D_i)$ are the domain transition probability and the background probability (used to smooth the domain transition probability estimates), respectively. They are defined as $P(D_i \mid D_{i-1}) = \frac{N(D_{i-1}, D_i) + \alpha N(D_{i-1}) P(D_i)}{(1+\alpha) N(D_{i-1})}$ and $P(D_i) = \frac{N(D_i)}{\sum_D N(D)}$, where $\mathrm{N}(\mathrm{D_i})$ is the number of domain $\mathrm{D_i}$ counted in the training dataset, $\mathrm{N}(\mathrm{D_{i-1}}, \mathrm{D_i})$ is the number of domain $\mathrm{D_i}$ following domain $\mathrm{D_{i-1}}$ counted in the training dataset, $\sum_{\mathrm{D}} \mathrm{N}(\mathrm{D})$ is all domain occurrences in the training dataset, and $\alpha \mathrm{N}(\mathrm{D}_{i-1})$ is the size of pseudocount which is used to avoid zero probability ($\alpha = 0.1$ is used) (Coin *et al.*, 2003) .

The linker length emission score, $\mathrm{L}(\mathrm{D_{i-1}}, \mathrm{D_i})$, is defined as $\log_2 \frac{P(K_{i-1} \mid D_{i-1}, D_i)}{P(K_{i-1})}$ and calculated as follows. Linker lengths between domain pairs ($\mathrm{D_{i-1}}$ and $\mathrm{D_i}$) are classified into four categories: Extreme long, Long, Medium and Short (see **Training dataset**

section). The probability distribution of linker length categories emitted from the domain pair D_{i-1} and D_i is defined as:

$$P(K_{i-1} \mid D_{i-1}, D_i) = \frac{N(K_{i-1} \mid D_{i-1}, D_i) + \alpha' N(D_{i-1}, D_i) P(K_{i-1})}{(1+\alpha') N(D_{i-1}, D_i)} \quad (2.14)$$

$$K_{i-1} \in (\text{Extreme long}, \ \text{Long}, \text{Medium}, \text{Short})$$

where $N(K_{i-1} \mid D_{i-1}, D_i)$ is the number of linker length category K_{i-1} between the domains D_{i-1} and D_i counted in the training dataset, $N(D_{i-1}, D_i)$ is the number of domain D_i following domain D_{i-1} counted in the training dataset, and 0.01 was used for α' to make the pseudocount small enough. The background frequency $P(K_{i-1})$ is estimated by $P(K_{i-1}) = \frac{N(K_{i-1})}{\sum_K N(K)}$, where $N(K_{i-1} \mid D_{i-1}, D_i)$ is the total number of linker length category K_{i-1} in the training dataset. $\alpha' = 0.01$.

To calculate the domain score $S(D)$, we did not penalize it by using negative linker scores. The linker score is thus re-defined as:

$$L(D_{i-1}, D_i) = \max(s \cdot \log_2 \frac{P(K_{i-1} \mid D_{i-1}, D_i)}{P(K_{i-1})}, 0) \quad (2.15)$$

where s is the scaling factor (described later).

Our goal is to find the domain sequence D^* that maximizes sequence score S(D) from the domain candidate set, $\mathbf{d} = \{d_1, d_2 \ldots d_m\}$, with corresponding amino acid sequences

$a_1, a_2 \ldots a_m$, on the given protein. We define D^j to be the highest scoring domain sequence that ends in domain d_j and use a dynamic programming technique to find the most likely domain architecture as follows:

1) Initialization

$$\begin{aligned} S(D^1) &= H(d_1) - T(d_1) \\ D^1 &= \{d_1\} \end{aligned} \tag{2.16}$$

2) Recursion:

$$S(D^j) = H(d_j) - T(d_j) + \max_{1 \le i < j, a_i \cap a_j = \phi} (S(D^i) + C(d_j \mid d_i) + L(d_i, d_j), 0) \tag{2.17}$$

where, $2 \le j \le m$

$$\begin{aligned} &\text{If} \quad S(D^i) + C(d_j \mid d_i) + L(d_i, d_j) > 0, \quad D^j = \{D^i, d_j\}; \\ &\text{otherwise,} \quad D^j = \{d_j\} \end{aligned} \tag{2.18}$$

3) Termination:

$$D^* = \arg\max_{1 \le j \le m} (S(D^j)) \tag{2.19}$$

$a_i \cap a_j = \phi$ in (2.17) ensures that no domain overlap occurs in the resulted domain sequence.

As described in Chapter 1, Pfam uses domain-specific gathering thresholds (GAs) at the domain level (domain GA) as well as at the sequence level (sequence GA). Following Pfam, we uses domain GAs in place of $T(D_i)$. The use of sequence GAs is implemented as follows. We first obtain D^* as above. Coin et al (2003) equally distributed domain dependency score of each domain pair on D method and add this score to the bit score of

each domain from corresponding pair. Similarly, we equally distributed domain dependency score, $C(D_j \mid D_i)$, and linker length score, $L(D_i, D_j)$, of each domain pair on D from HMM-DL and add this score to the bit score of each domain from corresponding pair (D_i and D_j). Finally, we sum the new scores of domains in the same family and compare this sum with the sequence GA of this family. The domains will be eliminated from D if the sum is smaller than the sequence GA.

2.2.3 Estimation of the false positive rates (FPRs) and the false discovery rates (FDRs)

To assess the performance of domain detection, we developed a method to estimate false positive rates (FPRs) and false discovery rates (FDRs) by shuffling protein sequences. Our assumption is

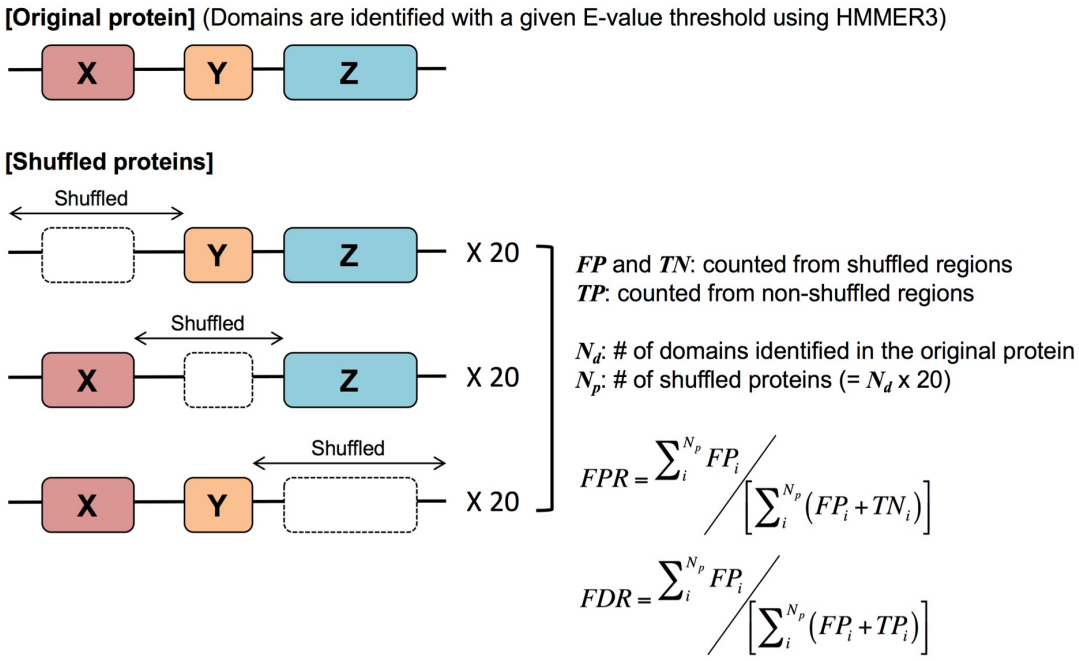

Figure 2.3: **Illustration of the FDR and the FPR estimation procedure.** For each original protein sequence, we apply HMMER and then shuffling for each domain is performed 20 times. We make predictions on the shuffled sequences and count predictions on the shuffled portions of sequences.

that any domains predicted on the shuffled portion of sequences were false positives (FPs) whereas predictions on real portion of sequences give us true positive predictions (TPs). Since HMM-DL boosts domain prediction by domain dependency and information of linker lengths between domains, shuffling procedure was done for each organism tested using the protein sequences with two or more domains identified by hmmscan (HMMER3, the option: cut_ga). As shown in Fig. 2.3, amino acid residues including each domain as well as surrounding linker (or N/C-terminal) regions were shuffled. By shuffling amino acid residues, sequences are randomized without changing amino acid composition. Each domain region was shuffled 20 times producing (the number of domains) x 20 of randomized sequences for each protein. For each of these protein

sequences, we applied domain prediction methods. Domains predicted from the shuffled portion of the sequence (p_i) were counted as FP_i, the number of shuffled region without any prediction was TN_i, and the predictions from non-shuffled regions of the sequence (p_i) were count as TP_i. The FDR and FPR are calculated as: $FPR = \frac{\sum_i^{N_p} FP_i}{\sum_i^{N_p} (FP_i + TN_i)}$ and $FDR = \frac{\sum_i^{N_p} FP_i}{\sum_i^{N_p} (FP_i + TP_i)}$.

2.2.4 Comparison of different approaches by FPR and FDR curves

We compared domain detection performance of HMM-DL against non-context method (hmmscan, v. 3.1b) (Finn *et al.*, 2011), dPUC2 (Ochoa, Llinás, and Singh, 2011b; Ochoa, 2013), DAMA (Bernardes *et al.*, 2016), and MC-DD (Coin *et al.*, 2003). Non-context method, hmmscan, was used as the baseline method, and used with options: --F1 0.1, --F2 0.1, --F3 0.0001, --domZ 1, -Z 1, and -E 0.00000001 (the E-value threshold was ranged up to 0.0000002). For domain-context methods (HMM-DL, MC-DD, dPUC2, DAMA), hmmscan was first used to obtain the candidate domain set with the following options: --F1 0.1, --F2 0.1, --F3 0.0001, --domZ 1, -Z 1, and -E 0.000001 (the permissive E-value threshold was ranged up to 0.005). For each E-value threshold, FPR and FDR were estimated using the shuffled sequences as described above and used to plot FPR and FDR curves. For the same set of candidate domain sets (including all proteins, without excluding those that have no or only a single domain predicted), each method was applied to search domains that were above the threshold and the number of domains per

protein was calculated. The dPUC2 program (version 1.03) and dpucNet.pfam27 (dPUC context scores from Pfam v27) were downloaded from http://viiia.org/dpuc2/index-2.03.yml?l=en-us. In order to avoid domain overlaps, we set the following options: --fCut 0 and --lCut 0. When a domain predicted by dPUC2 was entirely embedded within another predicted domain, the domain with a higher E-value was removed.

When running a query sequence against profile HMM (HMMER), some overlapping matches can be generated due to imprecise domain-boundary assignments. Therefore, allowing appropriate overlapping in domain detection should help increasing the performance for determining multi-domain architectures (Yeats, Redfern, and Orengo, 2010a; Bernardes *et al.*, 2016). When allowing domain overlaps, for dPUC2, the options were set to: --fCut 0.50 and --lCut 40 (the default parameters). With DAMA, while options –overlappingAA 0 and –overlappingMaxDomain 0 to avoid overlaps, the default options –overlappingAA 30 and –overlappingMaxDomain 50 were used to allow overlaps. Based on above selection of options for dPUC2 and DAMA, overlaps are allowed if the length of overlap is less than 40 amino acids and the overlap comprises at most 50% of the shortest match. In this process, $a_i \cap a_j = \phi$ in Eq. 2.17 was changed to allow overlaps for MC-DD and HMM-DL as done in dPUC2 and DAMA. If the end position of a_i minus the start position of a_j is less than 40 and the overlap comprises <50% of the shortest of a_i and a_j, the overlap was allowed in final prediction. The embedded domain predictions from dPUC2 were removed based on E-value.

2.2.5 Analysis of computational time

All runtime experiments were performed on a single-user Linux machine (Kernel Linux 3.5.0-34-generic Ubuntu 12.04 64 bit) with Intel(R) Core i5-3230M CPU 2.60GHz with 8GB RAM. The Perl module Time::HiRes was used to measure the execution time. Because all domain-context based approaches start with the set of potential domains identified by hmmscan, the execution time for hmmscan was not included in calculating search time.

2.2.6 Gene Ontology analysis

MultiPfam2GO (Forslund and Sonnhammer, 2008) was used to obtain Gene Ontology (GO) annotations from domain information obtained by HMMER3, MC-DD, and HMM-DLL. ReVigo (Supek *et al.*, 2011) was used to obtain the most specific GO terms, which removes redundant GO terms by finding representative GO terms from sibling terms and those related by inheritance.

2.3 Results

2.3.1 Determination of the scaling factor *s*

In Eq. 2.15, we used the scaling factor s to scale linker length scores. The selection of the scaling factor was done based on the observed FDR curves. 10,000 sequences were randomly selected from the UniRef50 dataset, and this random selection was repeated ten times to generate 10 groups of training datasets. Six values (2, 6, 8, 10, 16 and 22) were

tested to determine the scaling factor for calculating linker length scores. HMM-DL using each scaling factor was applied to the sequences of each group. The FDR curve was plotted using the average FDRs and the corresponding average number of domains per protein obtained from 10 datasets. As shown in Figure A5, the scaling factor of 10 produced the best performance based on the FDR curve. Therefore, we chose s=10 to calculate linker length scores for HMM-DL.

2.3.2 Comparison of domain identification performance

We compared the domain identification performance by HMM-DL against other methods including non-context method hmmscan with a range of E-values (0.00000001-0.0000002), context methods: dPUC2, MC-DD, and DAMA. For context methods, a range of E-value thresholds (0.000001 – 0.005) was used to detect candidate Pfam domains by using hmmscan from six sets of proteomes. Proteins that had two or more domains were used for the shuffling method (Fig. 2.3) to compute FPR and FDR for each domain detection method. As shown in Eq. 2.17, HMM-DL as well as MC-DD methods do not allow overlapped domains in selecting the maximum-scoring domain architecture. Therefore, the same condition was used with dPUC2 and DAMA in this comparison. For non-overlapping prediction, the similar results were obtained when using FDR (Fig. 2.4) and FPR (Fig. A6) to compare the performance of the prediction methods. As shown in Fig. 2.4 and Fig.A6, HMM-DL performed consistently better than hmmscan, dPUC2, MC-DD, and DAMA for *E. coli* and *H. sapiens* over the most range of FDRs and FPRs. HMM-DL outperforms DAMA in all tested organisms. In *S. cerevisiase*, HMM-DL

performs much better than the other prediction tools at the most of FPRs and FDRs. In *C. elegans*, HMM-DL predicts more domains than other prediction tools when FDRs and PFRs are larger, otherwise dPUC2 achieves a better performance. dPUC2 usually performed better than MC-DD except for *P. falciparum*, where the performance by HMM-DL, dPUC2, and MC-DD was similar, better than the performance of DAMA. Many domain types seem to be missing and domain coverage is lower from *P. falciparum* compared with the other organisms (Table A1). All domain-context approaches outperformed non-context method hmmscan, which suggests the introduction of domain context effectively control the False Positive rates (FPRs) and False Discovery Rates (FDRs).

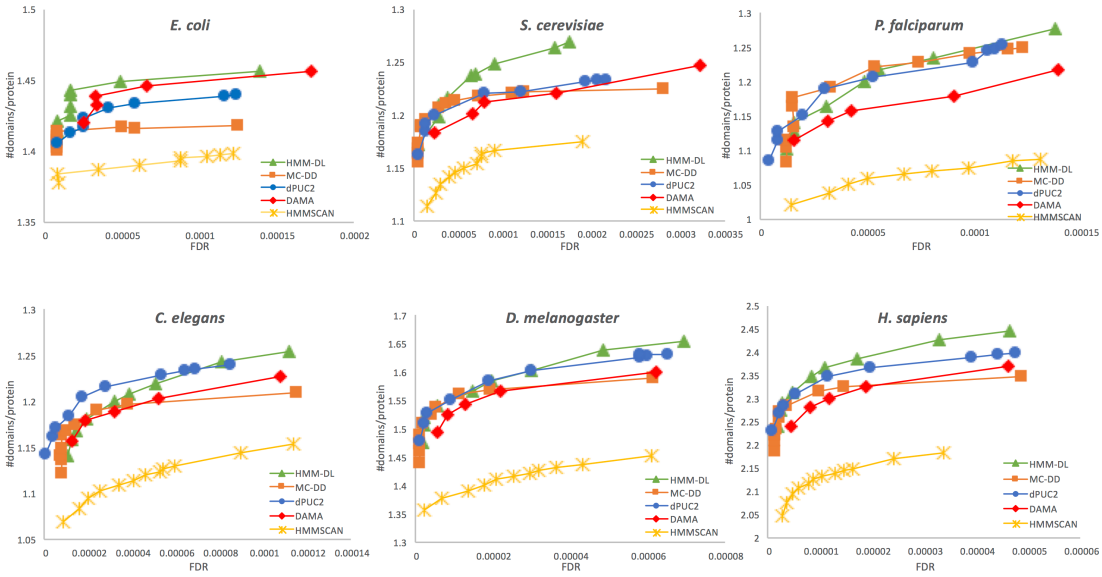

Figure 2.4: **Performance of HMMSCAN, dPUC2, DAMA, MC-DD and HMM-DL on domain identifications of *P. falciparum*, *E. coli*, *S. cerevisiae*, *D. melanogaster*, *C. elegans* and *H. sapiens* proteins.** The x-axis is the FDR and the y-axis is the number of predicted domains per protein. The better methods have higher curves.

Recent versions of dPUC2 as well as DAMA have options to allow overlapping domain matches. Therefore, we also used these options with dPUC2 and DAMA. The same conditions were used for HMM-DL and MC-DD to allow overlapping domains when searching domain architectures. The results are given in Fig. 2.5 (FDR) and in supplementary: Fig. A7 (FPR). Allowing overlaps in dPUC2 achieves a better performance than the other prediction methods in *C. elegans*, *D. melanogaster* and *H. sapiens*. In *E. coli*, DAMA with overlaps predicts more domain at the same FDRs and FPRs than HMM-DL with overlaps, but the performance of HMM-DL is better than that of dPUC2 when overlaps are allowed. In *S. cerevisiae*, HMM-DL with overlaps predicts more domains than the other prediction methods at given FDRs or FPRs.

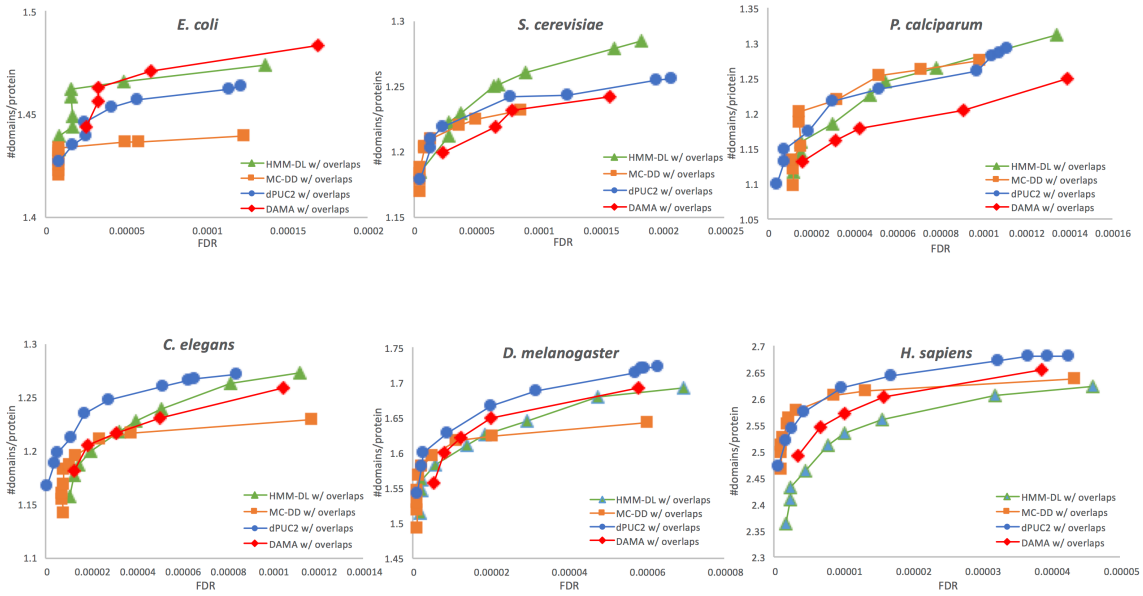

Figure 2.5: **Performance of HMMSCAN, dPUC2, MC-DD and HMM-DL with allowed overlaps on domain identifications of *P. falciparum*, *E. coli*, *S. cerevisiae*, *D. melanogaster*, *C. elegans* and *H. sapiens* proteins.** The x-axis is the FDR and the y-axis is the number of predicted domains per protein. The better methods have higher curves.

2.3.3 Domain identification from six proteomes

We identified domains from the six proteome sets using HMM-DL, MC-DD, dPUC2, DAMA, as well as the regular HMMER3. The default E-value threshold for dPUC2 is 0.0001. To compare the results by different methods obtained at the similar FDR and FPR levels using dPUC2 as the benchmark, based on our FDR and FPR analysis described above, we obtained the potential domains used as the input for each method using the following E-value thresholds with hmmscan: 0.001 for HMM-DL, 0.005 for MC-DD, 0.00003 for DAMA, and 0.0001 for dPUC2. To obtain the prediction by the regular HMMER3, hmmscan was used with the default parameters (the option –cut_ga) and possible domain overlaps were

resolved by taking the domain with the lowest E-value among those found in overlapping regions. Table 2.1 summarizes the domain identification results by different methods without allowing overlaps. Compared to the standard Pfam, HMM-DLL identified from 333 (for *E. coli*) to 6,256 (for human) more domains. The number of domain predictions increased by 5.6% (for *E. coli*) to 16.5% (for *P. falciparum*) compared to those obtained by the standard Pfam. Moreover, from 45 (for *E. coli*) to 118 (for *D. melanogaster*) domain types (Pfam families) were new to the organisms and had never been detected previously by using the standard Pfam analysis. While the increase in domain identification was the lowest for the *E. coli* proteome, whose genome is one of the best annotated, the highest increase was found in the *P. falciparum* with low protein coverage (Table A1). In *E. coli*, HMM-DL predicted more domains than DAMA at the lower FDR and FPR. Given the similar FDR and FPR, more domains were predicted by HMM-DL compared with dPUC2 in *D. melanogaster* and *H. sapiens* (Table 2.1).

For all organisms, HMM-DL had the similar computational time with MC-DD and both ran much faster than dPUC2 (Fig. 2.6 and Table A2) and slower than DAMA.

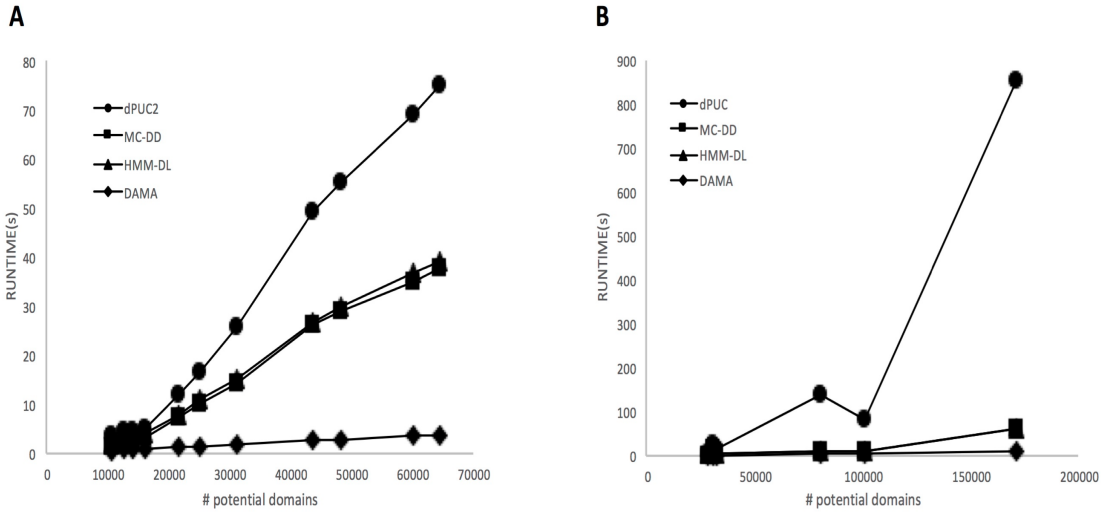

Figure 2.6: **dPUC2, MC-DD and HMM-DL time performance on the number of potential domains in *P. falciparum*(A) and on the number of potential domains in *P. falciparum*, *E. coli*, *S. cerevisiae*, *D. melanogaster*, *C. elegans* and *H. sapiens* proteins (B).**

2.3.4 Functional annotation of *P. falciparum* enhanced by newly identified domains by HMM-DL

P. falciparum is a protozoan parasite, the main causal agent of human malaria (White, 2004). Compared to the other organisms included in this study, the domain coverage (the proportion of the protein sequences where Pfam domains are assigned) is low and many domain types seem to be missed as indicated in Table A1. Using HMM-DL with E-value ≤0.001, 277, 403, 343 and 1,015 new domains were found compared with MC-DD (E-value ≤0.005), DAMA (E-value ≤0.00003), dPUC2 (E-value ≤0.0001) and standard Pfam (the option: --cut_ga), respectively. To show examples how domains newly detected by HMM-DL can contribute to protein functional annotation, we performed refinement of

GO annotations for those 226 proteins where new domains were found. Not all novel domains contribute to refining the functional annotations of protein. Table 2.2 summarizes the results of refined GO annotations based on domains identified by HMM-DL compared with the currently available annotations in PlasomDB (release 26). For example, the protein PF3D7_1304600 is annotated as "unknown function" in PlasmoDB and no Pfam domain is assigned for this protein by the standard Pfam, MC-DD, dPUC2 or DAMA. Using HMM-DL, two new repeating domains, SET (PF00856), were found and the protein's function can be now annotated as 'lysine N-methyltransferase activity'. For the PF3D7_1014800 protein, which is currently annotated as "conserved Plasmodium protein, unknown function", AKAP28 domain (PF14469) and two EF-hand_8 domains (PF13833) were detected by HMM-DL. The function of PF3D7_1014800 was annotated as "calcium ion binding" by new predicted domains. With these new domains, the function of PF3D7_1014800 was annotated as "calcium ion binding". This result was consistent with those from CD search where the aforementioned Pfam domain are in fact detected although with a weak E-value support ($4.78x10^{-3}$) (Fig. A11). As another example, for the PF3D7_0415700 annotated as "conserved, Plasmodium protein, unknown function", two Trigger_C domains (PF05698), which is associated with peptidyl-prolyl cis-trans isomerase activity, were found by only HMM-DL. These results show that Pfam domain search enhanced by HMM-DL can expand the protein annotation without using many other domain search methods. From the *P. falciparum* proteome, we could re-annotate 12 proteins, in total, that had been simply reported to have "unknown" or "conserved" functions previously.

2.4 Discussion

Enhancing domain identification is important for understanding protein functions (Yeats, Redfern, and Orengo, 2010b). The regular Pfam domain search using HMMER3 is highly conservative, and although its FPRs and FDRs are smaller, it has a difficulty in identifying divergent domains. In this study, we have shown that by combining domain co-occurrence and linker length information between domains we can improve the sensitivity of domain identification. HMM-DL has following appealing features leading to performance enhancement compared with other methods. First, HMM-DL, like MC-DD, is based on a Markov model, which takes the adjacency of domains into account, while CODD, dPUC2, and DAMA use only domain co-occurrence information. Second, HMM-DL, unlike MC-DD, takes into account the association of linker lengths with surrounding domain pairs. The introduction of additional information in the form of linker lengths lead to a reduction in false positive predictions. Finally, HMM-DL and MC-DD are fast owing to the use of a dynamic programming algorithm. dPUC2 is slower than HMM-DLL and MC-DD owing to its use of integer linear programming, which needs to search all possible combinations. DAMA is faster than HMM-DL and MC-DD. Two factors of DAMA that explain runtime enhancement: (1) DAMA is implemented in C++, while our method and dPUC2 are implemented in Perl; (2) DAMA enumerates all possible architectures based on domain co-occurrence constraints before applying the objective functions to select final predictions, which filters out some domain architectures to reduce runtime. This study also included the development of a shuffling method since our method is sensitive to the adjacency of domains. With this method, we could measure FPR as well as FDR from both HMM-DL and MC-DD.

Domains newly detected by HMM-DL are useful for expanding protein function annotations. As we showed, many proteins in *P. falciparum* that were previously annotated as unknown functions can be re-annotated. These new domains leading to the re-annotation of proteins had never been identified before in *P. falciparum*. For example, our study predicted two Tic22 (PF04278) domains on PF3D7_0415700, where Tic22 protein of *P. falciparum* (*pf*Tic22) is critical for parasite survival(Glaser *et al.*, 2012). This protein is an attractive drug target since *P. falciparum* is a protozoan parasite.

Domain overlaps are allowed in the newer version of dPUC2 (version=1.03) as well as recently released DAMA. We note that the performance of HMM-DL with overlaps is not as good as that of HMM-DL for non-overlapping domain prediction when compared with the other context prediction methods. This may be because the training dataset used for computation of domain dependency scores and linker length scores does not include overlapping domain predictions. Using training dataset containing observed domain overlap is expected to improve the performance of MC-DD and HMM-DL for allowed overlapping predictions. Further comparison and analysis between predictions by dPUC2 with and without overlaps in *E. coli* showed that most of newly identified domains by allowing overlaps (87 of 89) were those from the same clan. The remaining two overlapping domains were from WD40 domain. It implies that predictions with allowing overlaps do not contribute significantly to enhancing the annotation of protein function.

References

Apic,G. *et al.* (2001) Domain combinations in archaeal, eubacterial and eukaryotic proteomes. *Journal of Molecular Biology*, **310**, 311–325.

Bahl,A. (2003) PlasmoDB: the Plasmodium genome resource. A database integrating experimental and computational data. *Nucleic Acids Research*, **31**, 212–215.

Bernardes,J.S. *et al.* (2016) A multi-objective optimization approach accurately resolves protein domain architectures. *Bioinformatics*, **32**, 345–353.

Coin,L. *et al.* (2003) Enhanced protein domain discovery by using language modeling techniques from speech recognition. *Proceedings of the National Academy of Sciences*, **100**, 4516–4520.

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

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

Finn,R.D. *et al.* (2007) The Pfam protein families database. *Nucleic Acids Research*, **36**, D281–D288.

Forslund,K. and Sonnhammer,E.L.L. (2008) Predicting protein function from domain content. *Bioinformatics*, **24**, 1681–1687.

George,R.A. and Heringa,J. (2002) An analysis of protein domain linkers: their classification and role in protein folding. *Protein Eng.*, **15**, 871–879.

Glaser,S. *et al.* (2012) Tic22 is an essential chaperone required for protein import into the apicoplast. *J. Biol. Chem.*, **287**, 39505–39512.

Gokhale,R. (2000) Role of linkers in communication between protein modules. *Current Opinion in Chemical Biology*, **4**, 22–27.

Murzin,A.G. *et al.* (1995) SCOP: A structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, **247**, 536–540.

Ochoa,A. (2013) Protein domain prediction using context statistics, the false discovery rate, and comparative genomics, with application to Plasmodium falciparum.

Ochoa,A., Llinás,M., and Singh,M. (2011a) Using context to improve protein domain identification. *BMC Bioinformatics*, **12**, 90.

Ochoa,A., Llinás,M., and Singh,M. (2011b) Using context to improve protein domain identification. *BMC Bioinformatics*, **12**, 90.

Pizzi,E. and Frontali,C. (2001) Low-complexity regions in Plasmodium falciparum proteins. *Genome Res.*, **11**, 218–229.

Punta,M. *et al.* (2011) The Pfam protein families database. *Nucleic Acids Research*, **40**, D290–D301.

Supek,F. *et al.* (2011) REVIGO Summarizes and Visualizes Long Lists of Gene Ontology Terms. *PLOS ONE*, **6**, e21800.

Terrapon,N. *et al.* (2009) Detection of new protein domains using co-occurrence: application to Plasmodium falciparum. *Bioinformatics*, **25**, 3077–3083.

Terrapon,N. *et al.* (2012) Fitting hidden Markov models of protein domains to a target species: application to Plasmodium falciparum. *BMC Bioinformatics*, **13**, 67.

The UniProt Consortium (2015) UniProt: a hub for protein information. *Nucleic Acids Research*, **43**, D204–D212.

Vogel,C. *et al.* (2004) Supra-domains: Evolutionary Units Larger than Single Protein Domains. *Journal of Molecular Biology*, **336**, 809–823.

White,N.J. (2004) Antimalarial drug resistance. *J. Clin. Invest.*, **113**, 1084–1092.

Wriggers,W. *et al.* (2005) Control of protein functional dynamics by peptide linkers. *Biopolymers*, **80**, 736–746.

Yeats,C., Redfern,O.C., and Orengo,C. (2010a) A fast and automated solution for accurately resolving protein domain architectures. *Bioinformatics*, **26**, 745–751.

Yeats,C., Redfern,O.C., and Orengo,C. (2010b) A fast and automated solution for accurately resolving protein domain architectures. *Bioinformatics*, **26**, 745–751.

Table 2.1: **Domain identification from six proteomes using different methods.**[1]

Method / Statistics	Proteome (Number of proteins)					
	E. coli (4,305)	*S. cerevisiae* (6,720)	*P. falciparum* (5,542)	*C. elegans* (20,274)	*D. melanogaster* (13,674)	*H. sapiens* (20,882)
HMMER3						
# domains[2]	5,936 (2,607)	7,660 (2,937)	6,138 (1,834)	22,187 (3,622)	19,661 (3,987)	44,784 (5,470)
#FP (FPR/FDR)[3]	(0.02/0.011)	(0.003/0.001)	(0.005/0.002)	(0.027/0.005)	(0.014/0.002)	(0.017/0.002)
# proteins[4]	4,027	4,921	3,499	13,705	10,472	18,708
HMM-DL						
# domains[2]	6,269 (2,652)	8,528 (3,010)	7,153 (1,949)	25,420 (3,767)	22,622 (4,105)	51,040 (5,563)
#FP (FPR/FDR)[3]	(0.031/0.016)	(0.045/0.018)	(0.075/0.019)	(0.066/0.011)	(0.049/0.007)	(0.042/0.005)
# proteins[4]	4,032	4,945	3,562	13,859	10,531	18,767
MC-DD						
# domains[2]	6,106 (2,619)	8,187 (2,962)	6,876 (1,885)	24,534 (3,676)	21,722 (4,035)	49,020 (5,503)
#FP (FPR/FDR)[3]	(0.023/0.012)	(0.018/0.007)	(0.038/0.01)	(0.065/0.012)	(0.042/0.006)	(0.041/0.005)
# proteins[4]	4,032	4,948	3,581	13,885	10,545	18,787
DAMA						
# domains[2]	6,269(2,668)	8,382(3,044)	6,750(1,984)	24,884(3,791)	21,887(4,127)	49,476(5,580)
#FP (FPR/FDR)[3]	(0.032/0.017)	(0.081/0.032)	(0.052/0.014)	(0.062/0.011)	(0.042/0.006)	(0.039/0.005)
# proteins[4]	4,035	4,962	3,606	13,929	10,561	18,817
dPUC2						
# domains[2]	6,192 (2,623)	8,273 (2,959)	6,810 (1,877)	24,901 (3,671)	22,205 (4,028)	49,871 (5,492)
#FP (FPR/FDR)[3]	(0.022/0.012)	(0.049/0.019)	(0.037/0.01)	(0.031/0.005)	(0.041/0.006)	(0.034/0.004)
# proteins[4]	4,032	4,939	3,550	13,828	10,520	18,752

[1]The domain identification was performed using the following E-value thresholds: the default gathering thresholds for HMMER3 (hmmscan), 0.001 for HMM-DL, 0.005 for MC-DD, 0.00003 for DAMA, and 0.0001 for dPUC2.

[2]The number of domains identified. The number of Pfam domain families is shown in parentheses.

[3]The number of false positives estimated from the FPR given at the E-value threshold (shown in parentheses). FDRs are also shown in parenthesis.

[4]The number of proteins where domains were identified

Table 2.2: **Refined functional annotation by newly predicted domains**

Protein ID	Standard Pfam	dPUC2	DAMA	MC-DD	HMM-DL	Current annotation (PlasmoDB 26)	Reannotation
PF3D7_0930400.2	zf-CHY	zf-CHY	zf-CHY	zf-CHY	Myb_DNA-binding zf-CHY	conserved Plasmodium protein, unknown function	chromatin binding
PF3D7_1014800					AKAP28 EF-hand_8 EF-hand_8	conserved Plasmodium protein, unknown function	calcium ion binding
PF3D7_1304600					SET SET	conserved Plasmodium protein, unknown function	lysine N-methyltransferease activity
PF3D7_0823400					AXE1 AXE1	alpha/beta hydrolase, putative	cephalosporin-C deacetylase activity
PF3D7_0301800					ABC2_membrane_5 ABC2_membrane_5	Plasmodium exported protein, unknown function	ABC-type transport
PF3D7_1014800					AKAP28 EF-hand_8 EF-hand_8	conserved Plasmodium protein, unknown function	calcium ion binding
PF3D7_1035900			DUF2336 DUF2336	DUF2336 DUF2336	Trigger_C Trigger_C	probable protein, unknown function (M566)	peptidyl-prolyl cis-trans isomerase activity
PF3D7_0415700					Tic22 Tic22	conserved Plasmodium protein, unknown function	chaperone required for protein import into the apicoplast
PF3D7_1328500	Abhydrolase_5	Abhydrolase_5	Abhydrolase_5	Abhydrolase_5	Hydrolase_4 Hydrolase_4 Peptidase_S9	alpha/beta-hydrolase, putative	peptidase activity acting on L-amino acid peptides
PF3D7_0823400					AXE1 AXE1	alpha/beta hydrolase, putative	carboxylic ester hydrolase activity
PF3D7_1401500	Abhydrolase_6	Abhydrolase_6	Abhydrolase_6	Abhydrolase_6	Hydrolase_4 LCAT	lysophospholipase, putative	transferase activity, transferring acyl groups other than amino-acyl groups
PF3D7_1001600	Abhydrolase_5	Abhydrolase_5	Abhydrolase_5	Abhydrolase_5	Hydrolase_4 PGAP1	alpha/beta hydrolase, putative	hydrolase activity, acting on ester bonds
PF3D7_0528200	PCI	PCI	PCI	PCI	eIF3_N PCI	eukaryotic translation initiation factor 3 subunit E, putative (EIF3E)	translation initiation factor activity

Chapter 3

Using Inter-domain Linker Dependency to Identify Protein Domains

3.1 Introduction

We introduced a novel method (HMM-DL) in Chapter 2 for using inter-domain linkers to improve the identification of protein domain. In HMM-DL, we assume that the observed domain sequence is the result of an underlying unobserved (hidden) state (domain) sequence and inter-domain linker lengths are the emitted observations. The HMM-DL approach described in Chapter 2 is based on HMM. The limitation of HMM lies in the assumption of independence among the linker lengths. Therefore, by utilizing linker length dependency information, we expect to see increased domain identification sensitivity.

Double Chain Markov model (DCMM) proposed by Berchtold (Berchtold, 2009; 2007) allows more general dependency structure than HMM. As illustrated in Fig. 1.6, DCMM allows first-order dependency between successive observations given the hidden states. While DCMM has been used successfully in e.g., behavioral analysis (Berchtold and Sackett, 2002; Chariatte *et al.*, 2008) and social network analysis (Malmgren *et al.*, 2009), it has not been applied yet in bioinformatics. For multi-domain protein families, we can also incorporate linker length dependency information with DCMM (see Fig. 3.1C). Our second approach (DCMM-DLL) is to use the DCMM to model the domain dependency, the distribution of linker length, and the dependency between the linker lengths in domain detection (Fig. 3.1C).

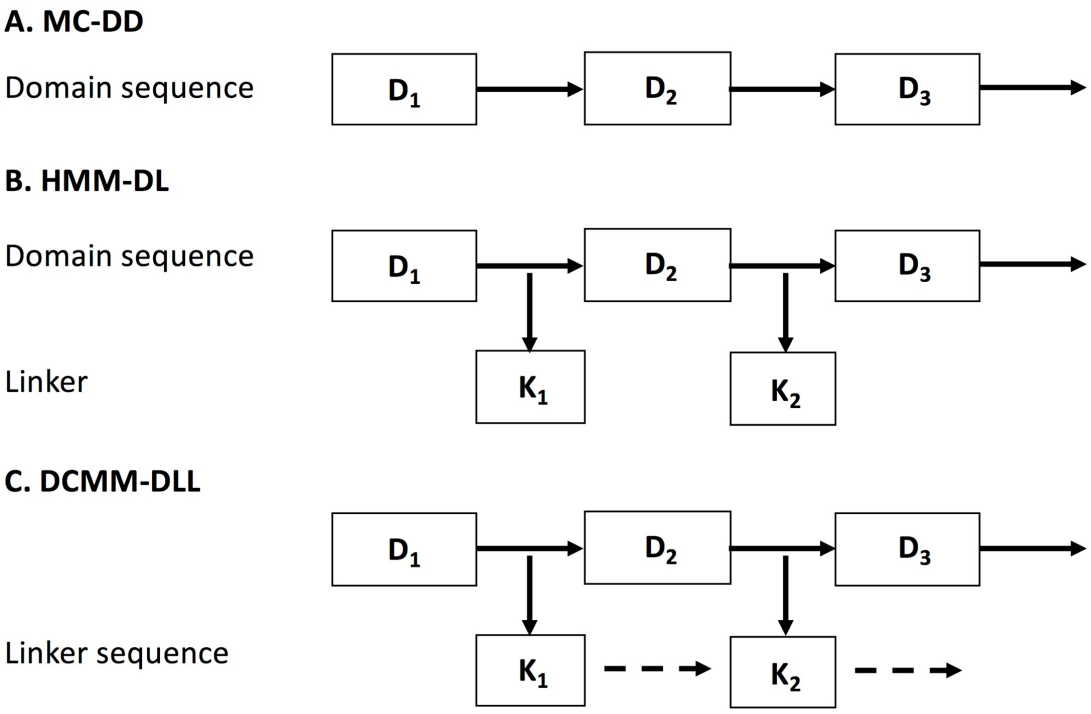

Figure 3.1: **Multi-domain protein representations for the three methods.** MC-DD (A) is a sequential domain dependency method based on the first-order Markov chain developed by Coin *et al.* (2003). HMM-DL (B) and DCMM-DLL (C) are our newly proposed methods that incorporate linker dependency information. D_i is a domain in a protein sequence. I_i is a linker length distribution between two domains D_i and D_{i+1}, which is emitted from the domain pair. Linker length dependency, indicated by dashed arrows, exists only in DCMM-DLL (C).

3.2 Materials and Methods

3.2.1 Datasets

Training dataset. We used the same dataset (UniRef50) used in Chapter 2 as training dataset. Using hmmscan from the HMMER3 software package (hmmer 3.1b2) and the Pfam v27, we identified 10,501,358 domains belonging to 14,828 Pfam domain families.

It included 3,644,227 domain pairs (and linkers) in 11,411 types of domain family pairs. Linkers were grouped into four categories based on their lengths: short, medium, long, and extra-long, in approximately equal numbers (~910,000 linkers in each category). 1,763,060 domain triplets were identified, which belonged to 10,4671 types of domain triplets. The linker-pairs were determined by the domain triplets and the types of linker-pairs were defined by the types of domain triplets as shown in Fig. 3.2.

Proteomes. The proteomes of six tested organisms (*E. coli, S. cerevisiae, P. falciparum, C. elegans, D. melanogaster, and H. sapiens*) used in Chapter 2 were used here to assess the performance of DCMM-DLL.

In order to examine whether linker length dependency is conserved for the specific domain triplets, we gathered the specific domain triplets that were observed 100 or more times in training dataset. For each type of domain triplet, the fraction of the total number of this triplet for each type of linker length category combination was calculated. If linker length dependencies are not conserved for the specific domain triplets, the most of values should tend to around the background probability. Fig. A8 shows that it was not the case, suggesting that linker-length dependencies are conserved for the specific domain triplets.

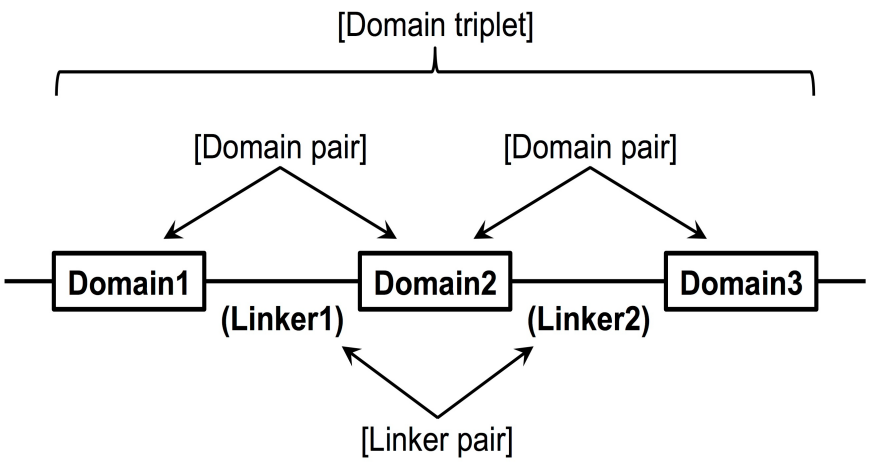

Figure 3.2: **Multidomain protein architecture.** Domains 1, 2, and 3 are all co-occurring domains in this protein. In this proposal, "domain pair", "linker pair", and "domain triplet" indicate pairs and triplets of domains and linkers directly adjacent to each other on a protein sequence.

3.2.2 Approach

Similar to Chapter 2, let $\mathbf{d} = \{d_1, d_2 \ldots d_m\}$ be the set of all the candidate domains with corresponding amino acid sequences $a_1, a_2 \ldots a_m$, where domains are ordered based on their ending amino acid positions in the protein sequence. Here, a candidate domain, d_j, is defined by the Pfam domain family (given as a profile HMM) and the position on the protein. Let $H(d_j)$ be the HMMER domain score of domain d_j, and $T(d_j)$ be the score threshold (see Supplementary file for the score threshold specific to each domain family used in Pfam). Let $D = D_1 \ldots D_n$ be a domain sequence without overlap, where $\{D_1, D_2 \ldots D_n\}$ is the subset of the candidate domain set **d** keeping the same numbering order as in **d**. Let $K = K_1 \ldots K_{n-1}$ be a linker-length sequence between domains in a domain sequence (D) and $Q = Q_1 \ldots Q_{n-2}$ be linker-length dependency among domain triplets (see Fig. 3.2). Our goal is to find the most likely sequence of domains D^* with

appropriate linkers among them given protein sequence (A) and first-order DCMM model (DCM):

$$D^* = \arg\max_D P(D,K,Q \mid A,DCM) \tag{3.1}$$

$$\begin{aligned} P(D,K,Q \mid A,DCM) &= \frac{P(D,K,Q,A,DCM)}{P(A,DCM)} = \frac{P(A \mid D,K,Q,DCM)}{P(A,DCM)} P(D,K,Q,DCM) \\ &= \frac{P(A \mid D,K,Q,DCM)}{P(A \mid DCM)P(DCM)} P(D,K,Q \mid DCM)P(DCM) \\ &= \frac{P(A \mid D,K,Q,DCM)}{P(A \mid DCM)} P(D,K,Q \mid DCM) \end{aligned} \tag{3.2}$$

Since domain sequence D contains information of domain families and their positions, Ks and Qs are known given D. Assuming that A is independent of the DCM model given D. So, $P(A \mid D,K,Q,DCM) = P(A \mid D)$ $P(A|D,K,Q,DCM) = P(A|D)$ and Eq. (3.2) can be written as

$$P(D,K,Q \mid A,DCM) = \frac{P(A \mid D)}{P(A \mid DCM)} P(D,K,Q \mid DCM) \tag{3.3}$$

$P(A \mid DCM)$ is a constant, and replacing it with another constant, $P(A \mid R)$, which is the probability of the protein sequence A given a random model (R), does not affect searching D with the highest probability. (Eq. 3.3) can be expressed as:

$$\begin{aligned} &P(D,K,Q \mid A,DCM) \\ &\propto \frac{P(A \mid D)}{P(A \mid R)} P(D,K,Q \mid DCM) = \left(\prod_i \frac{P(A_i \mid D_i)}{P(A_i \mid R)}\right) \times P(D,K,Q \mid DCM) \end{aligned} \tag{3.4}$$

Then,

$$P(D,K,Q \mid DCM) = \prod_i P(D_i \mid D_{i-1}) \cdot P(K_i \mid K_{i-1}, D_{i-1}, D_i, D_{i+1}) \tag{3.5}$$

We denote D_0 as the begin state and D_{n+1} as end state. Set $P(D_1|D_0)=1$, $P(K_n|D_n,D_{n+1})=1$, $P(K_1|K_0,D_0,D_1,D_2)=1$ and $P(K_n|K_{n-1},D_{n-1},D_n,D_{n+1})=1$. Then (3.5) can be rewritten as

$$\begin{aligned}
&P(D,K,Q|A,DCM)\\
&\propto(\prod_i \frac{P(A_i|D_i)}{P(A_i|R)})\times\prod_i(P(D_i|D_{i-1})\cdot P(K_i|K_{i-1},D_{i-1},D_i,D_{i+1}))\\
&=(\prod_i \frac{P(A_i|D_i)}{P(A_i|R)}\cdot P(D_i)\cdot P(K_i))\\
&\times(\prod_i \frac{P(D_i|D_{i-1})}{P(D_i)}\cdot\frac{P(K_i|D_i,D_{i+1})}{P(K_i)}\cdot\frac{P(K_i|K_{i-1},D_{i-1},D_i,D_{i+1})}{P(K_i|D_i,D_{i+1})})
\end{aligned} \tag{3.6}$$

This is equivalent to:

$$\begin{aligned}
&\log_2 P(D,K,Q|A,DCM)\\
&=\sum_i(\log_2\frac{P(A_i|D_i)}{P(A_i|R)}-\log_2\frac{1}{P(D_i)P(K_i)})+\sum_i\log_2\frac{P(D_i|D_{i-1})}{P(D_i)}\\
&+\sum_i\log_2\frac{P(K_i|D_i,D_{i+1})}{P(K_i)}+\sum_i\log_2\frac{P(K_i|K_{i-1},D_{i-2},D_{i-1},D_i)}{P(K_i|D_i,D_{i+1})}
\end{aligned} \tag{3.7}$$

Since $P(D_i)P(K_i)$ is constant, we use $T(D_i)=\log_2\frac{1}{P(D_i)P(K_i)}$, Eq (3.7) can be changed to

$$\begin{aligned}
&\log_2 P(D,K,Q|A,DCM)\\
&\propto\sum_i(\log_2\frac{P(A_i|D_i)}{P(A_i|R)}-T(D_i))+\sum_i\log_2\frac{P(D_i|D_{i-1})}{P(D_i)}\\
&+\sum_i\log_2\frac{P(K_i|D_i,D_{i+1})}{P(K_i)}+\sum_i\log_2\frac{P(K_i|K_{i-1},D_{i-2},D_{i-1},D_i)}{P(K_i|D_i,D_{i+1})}
\end{aligned} \tag{3.8}$$

Here $H(D_i)=\log_2\frac{P(A_i|D_i)}{P(A_i|R)}$ is the bit score for the domain D_i from HMMER and $T(D_i)$ is the score threshold.

For this method, we incorporate the information on the linker length dependency in the previously defined domain sequence score (Eq. 2.13). To do so, we define the linker length dependency score as

$$Q(D_{i-1},D_i,D_{i+1})=\log_2\frac{P(K_i\,|\,K_{i-1},D_{i-1},D_i,D_{i+1})}{P(K_i\,|\,D_i,D_{i+1})} \tag{3.9}$$

where K_{i-1} and K_i are the observed linker length categories between D_{i-1}, D_i and D_i, D_{i+1}, respectively,

$$\begin{aligned}&P(K_i\,|\,K_{i-1},D_{i-1},D_i,D_{i+1})\\&=\frac{N(K_i\,|\,K_{i-1},D_{i-1},D_i,D_{i+1})+\alpha''N(D_i,D_{i+1})P(K_i\,|\,D_i,D_{i+1})}{N(K_{i-1},D_{i-1},D_i,D_{i+1})+\alpha''N(D_i,D_{i+1})}\end{aligned} \tag{3.10}$$

$$P(K_i\,|\,D_i,D_{i+1})=\frac{N(K_i\,|\,D_i,D_{i+1})+\alpha'N(D_i,D_{i+1})P(K_i)}{(1+\alpha')N(D_i,D_{i+1})} \tag{3.11}$$

here, $N(K_i\,|\,K_{i-1},D_{i-1},D_i,D_{i+1})$ and $N(K_{i-1},D_{i-1},D_i,D_{i+1})$ are the count of the linker length pair (K_{i-1},K_i) and the count of K_{i-1} from the domain triplet (D_{i-1},D_i,D_{i+1}) in the training dataset, respectively. $N(K_i\,|\,D_i,D_{i+1})$ is the number of linker length category K_i between the domains D_i and D_{i+1} counted in the training dataset, $N(D_i,D_{i+1})$ is the number of domain D_{i+1} following domain D_i counted in the training dataset, and $\alpha''=0.00001$ to make the pseudocount small enough.

Similar to calculation of linker length scores, we did not penalized unobserved domain triplets and negative linker length dependency scores when calculating the domain

sequence score S(D). The linker length dependency score is thus re-defined as:

$$Q(D_{i-1},D_i,D_{i+1}) = \max(s \cdot \log_2 \frac{P(K_i \mid K_{i-1},D_{i-1},D_i,D_{i+1})}{P(K_i \mid D_i,D_{i+1})},0) \tag{3.12}$$

where s is the scaling factor. We used the same value we used in Chapter 2.

With the linker length dependency score (3.12), we define the domain sequence score of D as follows

$$\begin{aligned} &S(D) = H(D_1) - T(D_1) \\ &+ \sum_{i=2}^{n} (H(D_i) - T(D_i) + C(D_i \mid D_{i-1}) + L(D_{i-1},D_i) + Q(D_{i-1},D_i,D_{i+1}) \end{aligned} \tag{3.13}$$

Our goal is to find the domain sequence $D = D_1 \ldots D_n$ that maximizes sequence score $S(D)$. If $Q(D_{i-1},D_i,D_{i+1}) > 0$, it enhances S(D) which helps weak domains to be identified.

1) Initialization 1:

$$\begin{aligned} &S(D^1) = H(d_1) - T(d_1) \\ &D^1 = \{d_1\} \end{aligned} \tag{3.14}$$

2) Initialization 2:

$$\begin{aligned} &S(D^2) = H(d_2) - T(d_2) + \max_{a_1 \cap a_2 = \phi} (S(D^1) + C(d_2 \mid d_1) + L(d_1,d_2),0) \\ &\text{If} \quad S(D^1) + C(d_2 \mid d_1) + L(d_1,d_2) > 0, D^2 = \{D^1,d_2\} \\ &\text{otherwise} \quad D^2 = \{d_2\} \end{aligned} \tag{3.15}$$

3) Recursion:

$$
\begin{aligned}
&S(D^k)\\
&=H(d_k)-T(d_k)\\
&+\max_{1\le i<j<k,(a_i\cap a_j)\cup(a_j\cap a_k)=\phi}(S(D^i)+C(d_j\mid d_i)+L(d_i,d_j)+C(d_k\mid d_j)+L(d_j,d_k)\\
&+Q(d_i,d_j,d_k),S(D^i)+C(d_k\mid d_i)+L(d_i,d_k),0)
\end{aligned}
\tag{3.16}
$$

where $3\le k\le m$

$$
\begin{aligned}
&\text{If}\\
&S(D^i)+C(d_j\mid d_i)+L(d_i,d_j)+C(d_k\mid d_j)+L(d_j,d_k)+Q(d_i,d_j,d_k)\\
&>S(D^i)+C(d_k\mid d_i)+L(d_i,d_k)\\
&\textit{and}\\
&S(D^i)+C(d_j\mid d_i)+L(d_i,d_j)+C(d_k\mid d_j)+L(d_j,d_k)+Q(d_i,d_j,d_k)>0,\\
&D^k=\{D^i,d_j,d_k\};\\
&\text{Elseif}\\
&S(D^i)+C(d_k\mid d_i)+L(d_i,d_k)\\
&>S(D^i)+C(d_j\mid d_i)+L(d_i,d_j)+C(d_k\mid d_j)+L(d_j,d_k)+Q(d_i,d_j,d_k)\\
&\text{and}\\
&S(D^i)+C(d_k\mid d_i)+L(d_i,d_k)>0,\\
&D^k=\{D^i,d_k\};\\
&\text{otherwise}\\
&D^k=\{d_k\}
\end{aligned}
\tag{3.17}
$$

4) Termination:

$$D^*=\arg\max_{1\le k\le m}(S(D^k)) \tag{3.18}$$

$(a_i\cap a_j)\cup(a_j\cap a_k)=\phi$ in (Eq. 3.16) ensures that no domain overlaps occur in the resulted domain sequence.

As described in Chapters 1 and 2, Pfam, MC-DD and HMM-DL use domain-specific gathering thresholds (GAs) at the domain level (domain GA) as well as at the sequence level (sequence GA). In a similar way, we equally distributed domain dependency score

($C(d_j|d_i)$, $C(d_k|d_j)$) and linker length score ($L(d_i,d_j)$, $L(d_j,d_k)$) to the bit score of each domain from corresponding pair; and linker length dependency score ($Q(d_i,d_j,d_k)$) of each domain triplet on D from DCMM-DLL and add this score to the bit score of each domain from corresponding triplet. Finally, we sum the new scores of domains in the same family and compare this sum with sequence GA of this family. The domains will be eliminated from D if the sum is smaller than the sequence GA.

3.2.3 Estimation of the false positive rates (FPRs) and the false discovery rates (FDRs)

A shuffling method has been developed in Chapter 2 to assess the performance of different domain prediction methods (HMM-DL, MC-DD, dPUC2 and DAMA). Since DCMM-DLL boosts domain prediction by not only domain dependency and inter-domain linker length but also linker-pairs among domain triplets, shuffling procedure was done for each organism tested using the protein sequences with three or more domains by hmmscan (HMMER3, the option: cut_ga). To obtain enough simulated sequences for estimation of FPRs and FDRs in each organism, each domain region was shuffled 40 times for each sequence in tested organisms.

In Chapter 2, we constructed FDR and FPR curves to compare the performances of different prediction methods. Similarly, FDR and FPR curves were used to compare DCMM-DLL with the other methods. In order to allow appropriate overlapping domain identification, $(a_i \cap a_j) \cup (a_j \cap a_k) = \phi$ in Eq. 3.16 was changed to allow overlaps in

DCMM-DLL. If both the end position of a_i minus the start position of a_j and the end position of a_j minus the start position of a_k are less than 40, respectively and the overlap comprises <50% of the shortest match of each overlapped pair, the overlaps were allowed in final prediction.

3.2.4 Analysis of computational time

All runtime experiments were performed on a single-user Linux machine (Kernel Linux 3.5.0-34-generic Ubuntu 12.04 64 bit) with Intel(R) Core i5-3230M CPU 2.60GHz with 8GB RAM. The Perl module Time::HiRes was used to measure the execution time. Because all domain-context based approaches start with the set of potential domains identified by hmmscan, the execution time for hmmscan was not included in calculating search time.

3.3 Results

3.3.1 Comparison of domain identification performance

We compared the domain identification performance of DCMM-DLL with the other approaches, including non-context method hmmscan with a range of E-values (0.00000001-0.0000002) for non-overlapping domain prediction, context methods: HMM-DL, MC-DD, dPUC2 and DAMA for both non-overlapping and overlapping domain prediction. For context methods, the range of E-values (0.000001 – 0.005) was used to detect candidate Pfam domains by using hmmscan from six sets of proteomes. Proteins that had three or more domains were used for the shuffling method (Fig. 2.3) to

compute FPR and FDR for each domain detection method. For non-overlapping prediction, the similar results were obtained when using FDR (Fig. 3.3) and FPR (Fig. A9) to compare the performance of the prediction methods. As shown in Fig. 3.3 and Fig. A9, the performance by HMM-DL and DCMM-DLL was similar and better than the performance of the other methods (HMMSCAN, MC-DD, dPUC2 and DAMA).

In Chapter 2, we note that dPUC2 with allowed overlaps achieves a better performance than the other prediction methods including HMM-DL in *D. melanogaster* and *H. sapiens*. Allowing overlaps in DCMM-DLL achieves a better performance than dPUC2 with allowed overlaps in *D. melanogaster* and *H. sapiens* (Fig. 3.4 and Fig. A10). In *E. coli*, DAMA with overlaps predicts more domain at the same FDRs and FPRs than HMM-DL with overlaps and DCMM-DLL with overlaps, but the performance of HMM-DL and DCMM-DLL are better than that of dPUC 2 when overlaps are allowed.

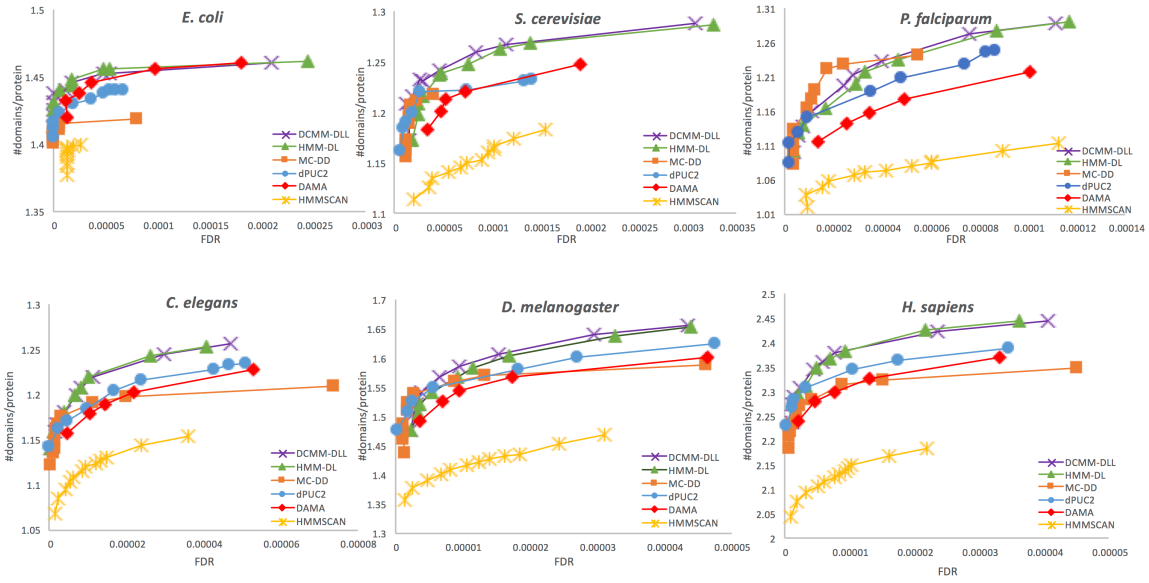

Figure 3.3: **Performance of HMMSCAN, dPUC2, MC-DD, HMM-DL and DCMM-DLL on domain identifications of *P. falciparum*, *E. coli*, *S. cerevisiae*, *D. melanogaster*, *C. elegans* and *H. sapiens* proteins.** The x-axis is the FDR and the y-axis is the number of predicted domains per protein. The better methods have higher curves.

3.4 Discussion

In this study, we have shown that the similar performance was achieved by DCMM-DLL and HMM-DL. Both of them were better than the other context methods for non-overlapping domain prediction. This may be because no much more information was obtained from domain triplets including linker dependency compared with the information from domain pairs. Interestingly, allowing overlaps in DCMM-DLL improved the identification of domains in *D. melanogaster* and *H. sapiens* compared with the other context methods with overlaps. The possible reason for this improvement is that the relatively higher percentage of proteins from *D. melanogaster* and *H. sapiens* contains three or more domains (Fig. A4) compared with the other organisms, and incorporating information of domain triplets provided more power for resolving the

incorrect boundary positions of domains.

DCMM-DLL is faster than dPUC2, but slower than MC-DD and HMM-DL. This is because DCMM-DLL used more complicated recursion steps including not only domain pairs in MC-DD and HMM-DL but also domain triplets.

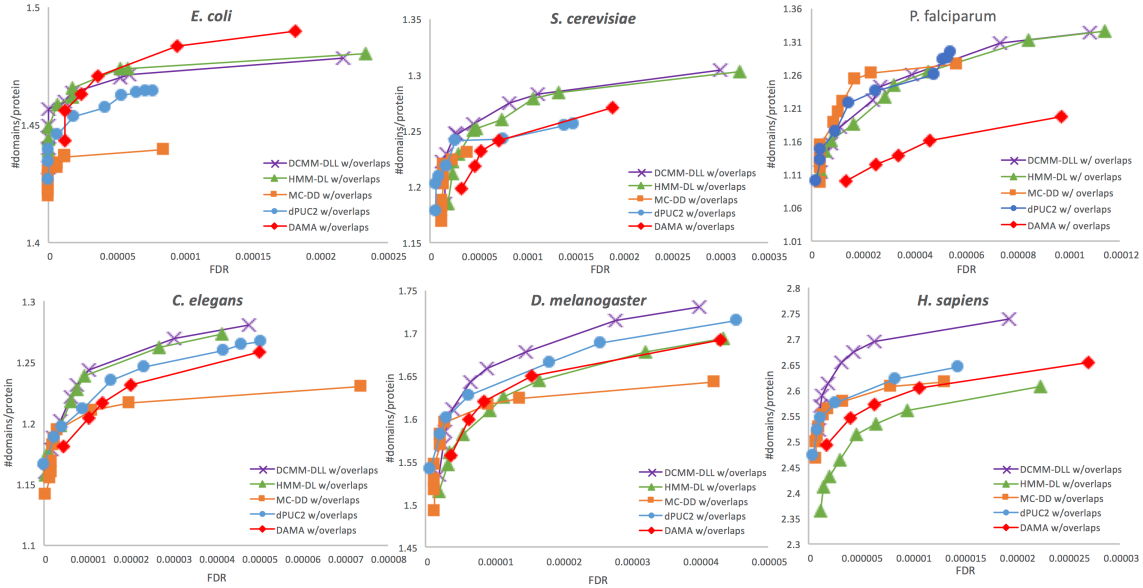

Figure 3.4: **Performance of HMMSCAN, dPUC2, MC-DD, HMM-DL and DCMM-DLL with allowed overlaps on domain identifications of *P. falciparum*, *E. coli*, *S. cerevisiae*, *D. melanogaster*, *C. elegans* and *H. sapiens* proteins.** The x-axis is the FDR and the y-axis is the number of predicted domains per protein. The better methods have higher curves.

References

Berchtold,A. (2007) High-order extensions of the Double Chain Markov Model. *Stochastic Models*, **18**, 193–227.

Berchtold,A. (2009) The double chain markov model. *Communications in Statistics - Theory and Methods*, **28**, 2569–2589.

Berchtold,A. and Sackett,G. (2002) Markovian models for the developmental study of social behavior. *American Journal of Primatology*, **58**, 149–167.

Bioinformatics Bioinformatics Oxford University Press.

Chariatte,V. *et al.* (2008) Missed Appointments in an Outpatient Clinic for Adolescents, an Approach to Predict the Risk of Missing. *Journal of Adolescent Health*, **43**, 38–45.

Malmgren,R.D. *et al.* (2009) Characterizing individual communication patterns ACM, New York, New York, USA.

Chapter 4

Conclusion and Future Research

4.1 Summary

In this dissertation, two statistical methodologies to improve the sensitivity of protein domain prediction have been introduced. A new simulation method has also been developed to assess the domain prediction approaches. The first domain prediction method, HMM-DL (hidden Markov model using domain dependency and linker length) captured the information from both domain dependency and inter-domain linker length in proteins to identify domains without markedly increasing the false discovery rates (**Chapter 2**). The main feature of HMM-DL is that it takes inter-domain linkers and adjacency of domains into consideration so that more information is used to control the false positives. Our benchmarks showed that HMM-DL improved non-overlapping domain predictions for the known model organisms compared with the current context-based approaches (MC-DD, dPUC2, DAMA)., HMM-DL was also faster than dPUC2, while it was a little slower than DAMA. As an example, newly predicted domains were used to refine the functional annotation of proteins, especially for proteins which are left "unanotated" (no function is assigned). The imprecise predicted domain boundaries can lead to conflict overlapping domains on proteins (Yeats *et al.*, 2010; Bernardes *et al.*, 2016). Thus, domain overlaps were allowed in HMM-DL to increase the number of correct domain predictions.

The second domain prediction method we developed, DCMM-DLL, was based on the

idea of dependency between successive inter-domain linkers, which is carried by the Double Chain Markov Model (DCMM). DCMM is a generalization of HMM with dependency between observations. Application of DCMM-DLL (DCMM using Domain dependency, Linker length and Linker length dependency, **Chapter 3**) on protein sets from the model organisms showed that DCMM-DLL for non-overlapping domain prediction performed similarly to HMM-DL, both better than the other methods. Moreover, DCMM-DLL showed improved domain predictions when domain-overlaps were allowed in *C. elegans, D. melanogaster and H. sapien* datasets. DCMM-DLL compensated the weakness of HMM-DL in prediction of overlapping domains.

4.2 Future Research

Extension of the HMM-DL and DCMM-DLL to include higher, variable-order dependency. Many multi-domain proteins contain more than three domains. Therefore, using only the first-order in Markov models restricts the power of domain indentification (Coin *et al.*, 2003). HMM and DCMM can be generalized to have higher, multi-order dependencies among the domains and the linker lengths. Fang *et al*. (2009) solved this problem and showed that multi-order DCMM can improve modeling performance over the first-order DCMM. Removing the restriction of usig fixed-order HMM and DCMM can also improve the ability of domain prediction for proteins. Therefore, we will develop variable-order HMM and DCMM with flexible dependency structures that can model domain sequences with complex structures.

Using various training sets. Domain architecture and linker information can be gathered from a wide range of protein family/domain databases. Different organism groups are likely to have different domain structures and different linker length distributions. Therefore, taxonomy-specific training may improve prediction performance (Terrapon *et al.*, 2012). We will generate most inclusive as well as taxonomy-specifc traning datasets and evaluate the performance. Since the boundaries of predicted domains are imprecise, allowing appropriate overlapping in domain detection should help increasing the performance for determining multi-domain architectures (Yeats *et al.*, 2010; Bernardes *et al.*, 2016). In our current implementation, non-overlapping domain predictions by hmmscan (the option: cut_ga) were used to calcualte domain dependency scores, linker length scores and linker length dependency scores for MC-DD, HMM-DL and DCMM-DLL. The improvement in domain identificaiton by our methods is expected if the training dataset contains proteins with overlapped domains.

Linker sequence properties. Secondary structures, disorderness, and other properties in linker regions can be associated with neighboring domains (Dong *et al.*, 2006; Ekman *et al.*, 2005) (Shatnawi and Zaki, 2015). Therfore, in addition to their lengths, we will incorporate amino acid profiles of linkers in our models as the covariates and examine how it affects the domain identificiation performance.

Inclusion of N/C-terminals. In our preliminary study, we used only inter-domain linkers. In reality, N- and C-terminal regions are important parts of proteins.

Incorporating properties from these regions could further enhance the performance of domain discovery. Since DCMM-DLL requires two linker regions, inclusion of N/C-terminals also extends applicability of DCMM-DLL to single domain proteins.

Evaluation of domain discovery performance using simulated data. Our preliminary analysis was based on simple simulation data. More realistic and extensive benchmark datasets will be generated using sequence simulators with varying evolutionary parameters. For example, Indel-Seq-Gen (iSG) (Strope *et al.*, 2009; 2007) and REvolver (Koestler *et al.*, 2012) can simulate protein evolution with multi-domain architectures. Using simulated multi-domain protein sequences with various divergence levels, we will evaluate performance of each domain discovery method using FPR, FDR, F-meausre (a weighted harmonic mean of Precision and Recall), as well as AUC (the area under the receiver operating characteristic, or ROC, curve).

Acceleration heuristic for HMM-DL and DCMM-DLL. Our results showed that HMM-DL and DCMM-DLL made contributions to the domain prediciton and sequence similarity search. However, they were slower than DAMA. The increase in the sequence databases may hinder efficient utility of our methods. Using some filters, for example, we can remove highly divergent protein sequences from the set of potential domains and the rest of the domains can be used as input for HMM-DL and DCMM-DLL to reduce runtime.

Application of HMM-DL and DCMM-DLL. We will apply our domain discovery approaches against various genomes across kingdoms. Comparative analysis of domain architectures among organisms and kingdoms will be carried out as part of evaluation of newly identified domains. For example, domains that are identified as part of conserved domain architectures can be considered more reliable.

References

Bernardes,J.S. *et al.* (2016) A multi-objective optimization approach accurately resolves protein domain architectures. *Bioinformatics*, **32**, 345–353.

Coin,L. *et al.* (2003) Enhanced protein domain discovery by using language modeling techniques from speech recognition. *Proceedings of the National Academy of Sciences*, **100**, 4516–4520.

Dong,Q. *et al.* (2006) Domain boundary prediction based on profile domain linker propensity index. *Computational Biology and Chemistry*, **30**, 127–133.

Ekman,D. *et al.* (2005) Multi-domain Proteins in the Three Kingdoms of Life: Orphan Domains and Other Unassigned Regions. *Journal of Molecular Biology*, **348**, 231–243.

Fang,X. (2009) Sequence comparison and stochastic model based on multi-order Markov models.

Koestler,T. *et al.* (2012) REvolver: modeling sequence evolution under domain constraints. *Mol Biol Evol*, **29**, 2133–2145.

Shatnawi,M. and Zaki,N. (2015) Inter-domain linker prediction using amino acid compositional index. *Computational Biology and Chemistry*, **55**, 23–30.

Strope,C.L. *et al.* (2009) Biological sequence simulation for testing complex evolutionary hypotheses: indel-Seq-Gen version 2.0. *Mol Biol Evol*, **26**, 2581–2593.

Strope,C.L. *et al.* (2007) indel-Seq-Gen: a new protein family simulator incorporating domains, motifs, and indels. *Mol Biol Evol*, **24**, 640–649.

Terrapon,N. *et al.* (2012) Fitting hidden Markov models of protein domains to a target species: application to Plasmodium falciparum. *BMC Bioinformatics*, **13**, 67.

Yeats,C. *et al.* (2010) A fast and automated solution for accurately resolving protein domain architectures. *Bioinformatics*, **26**, 745–751.

Appendix

A.1 Tables

Table A1: **Number of distinct Pfam domains and protein coverage in test organisms**

Organisms	Proteome size	Avg. protein length	Pfam domains	Domain types	aa coverage
E . coli	4305	315	5936	2607	70%
S. cerevisiae	6720	450	7660	2937	41%
P. falciparum	5542	756	6138	1833	21%
C. elegans	20274	412	22187	3622	40%
D. melanogaster	13674	539	19661	3987	36%
H. sapiens	20882	547	44784	5470	43%

Table A2: **Runtime performance comparison for the context methods**

Methods	***P. falciparum***	***E. coli***	***S. cerevisiae***	***C. elegans***	***D. melanogaster***	***H. sapiens***
dPUC2	26.08±0.12 (25.01, 26.18)	6.85 ± 0.1 (5.87, 6.91)	14.21±0.08 (13.88, 14.83)	82.97±0.33 (82.67, 86.17)	139.3±3.12 (138.56, 170.12)	854.31±6.04 (852.26, 912.18)
MC-DD	14.48±0.12 (14.38, 14.89)	2.42 ± 0.03 (2.38, 2.60)	2.73±0.03 (2.68, 2.86)	10.36±0.11 (10.15, 10.98)	9.45±0.07 (9.31, 9.61)	63.48±0.5 (62.89, 65.95)
HMM-DL	15.01±0.14 (14.89, 15.68)	3.02 ± 0.02 (2.98, 3.11)	3.32±0.03 (3.27, 3.42)	12.49±0.09 (12.28, 12.75)	11.64±0.09 (11.47, 11.94)	63.59±0.64 (62.91, 66.17)
DCMM-DLL	92.11±0.78 (91.19, 93.69)	8.59 ± 0.17 (8.4, 8.94)	8.04±0.12 (7.89, 8.22)	65.71±1.02 (64.09, 67.89)	65.06±1.1 (63.82, 67.36)	662.19±4.79 (657.57, 673.03)
DAMA	1.85±0.02 (1.81,1.90)	1.67 ± 0.02 (1.63, 1.77)	1.91±0.02 (1.88, 2.09)	5.90±0.2 (5.78, 7.69)	4.67±0.05 (4.59, 4.86)	10.04±0.2 (9.73, 10.52)
No. proteins	5542	4305	6720	20274	13674	20882
No. potential domains	31243	28688	32967	101074	80838	171382

A.2 Figures

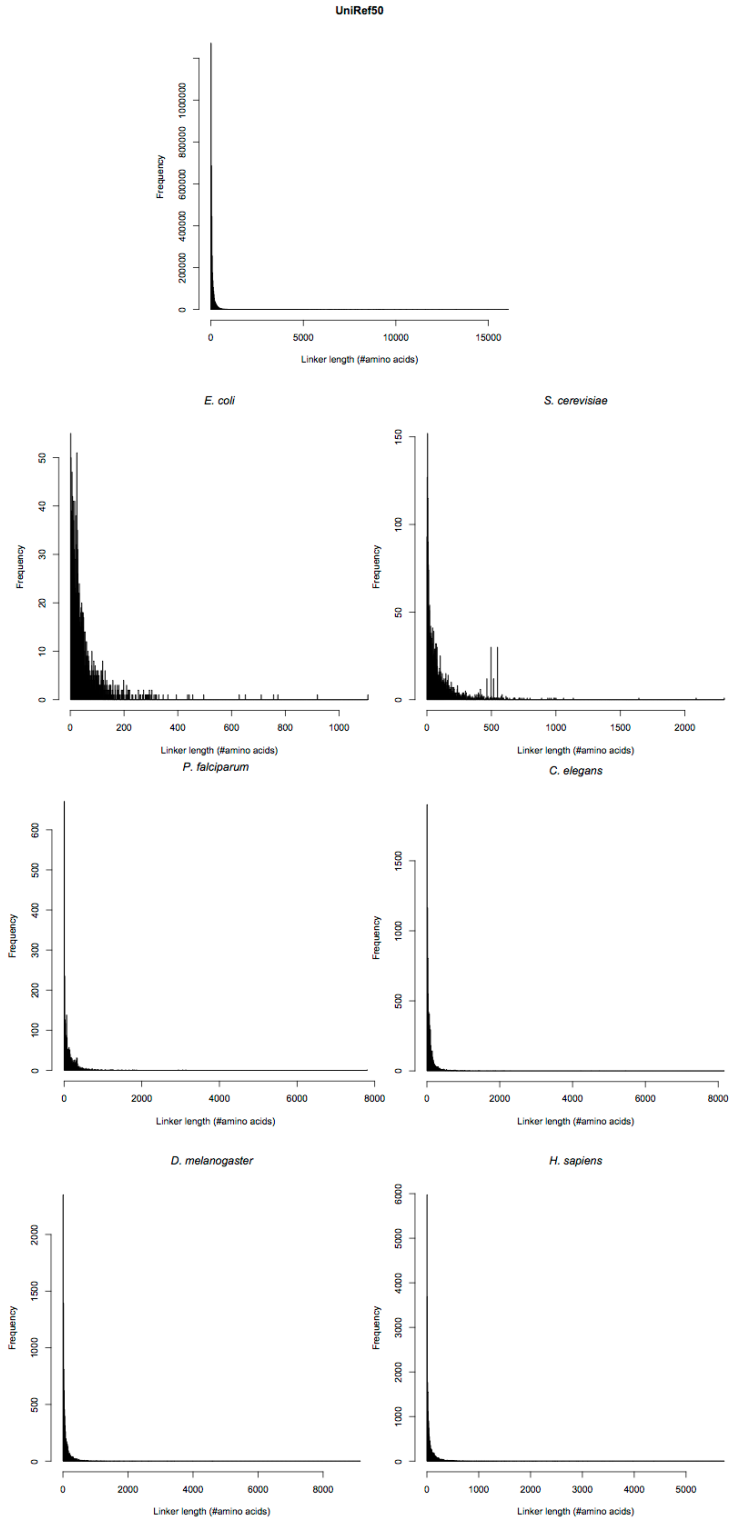

Figure A1: **Histogram of Linker lengths from UniRef50 (top) and from different model organisms.**

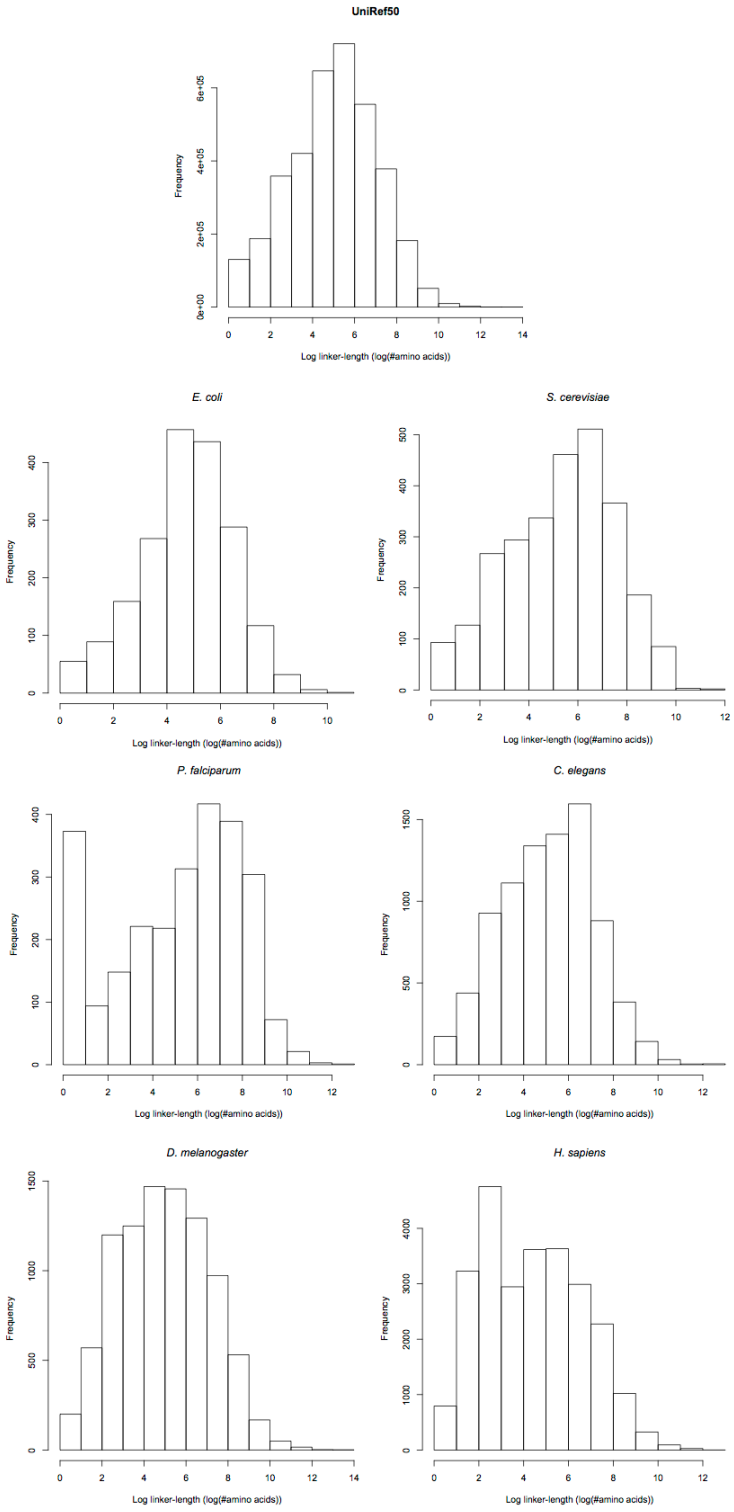

Figure A2: **Histogram of log Linker lengths from UniRef50 (Top) and from different model organism.**

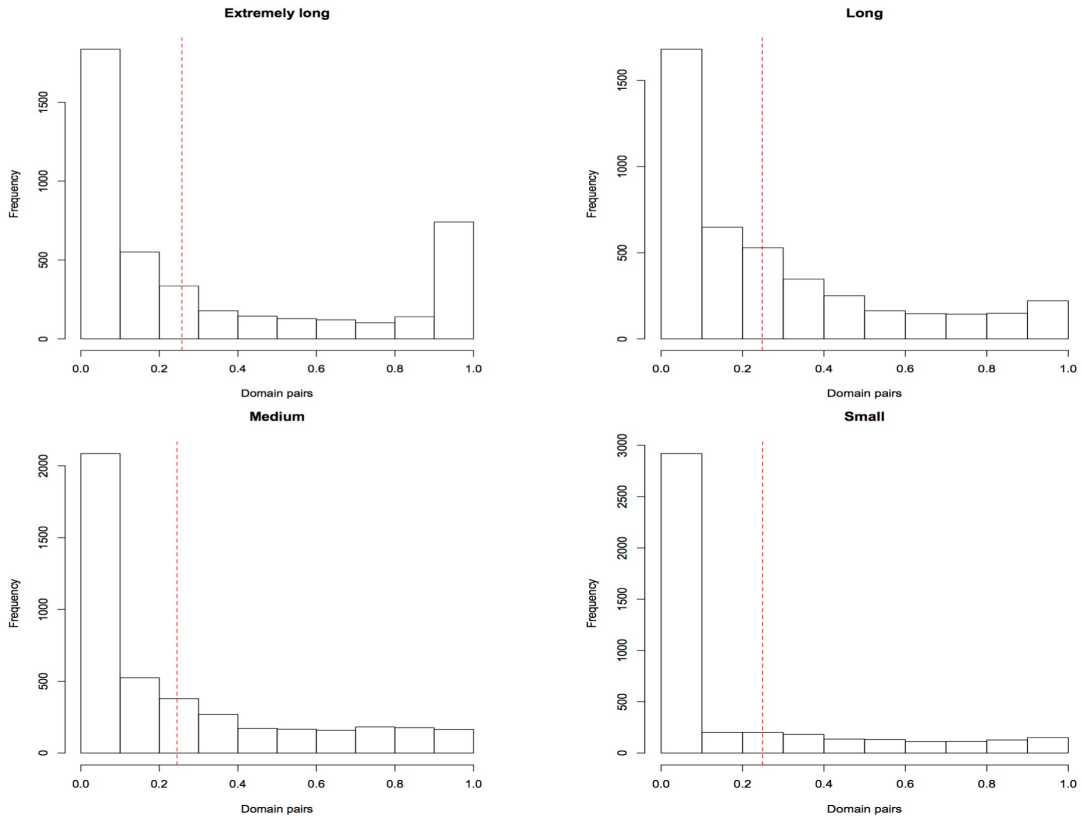

Figure A3: **Testing the conservation of linker lengths for the specific domain pairs.**

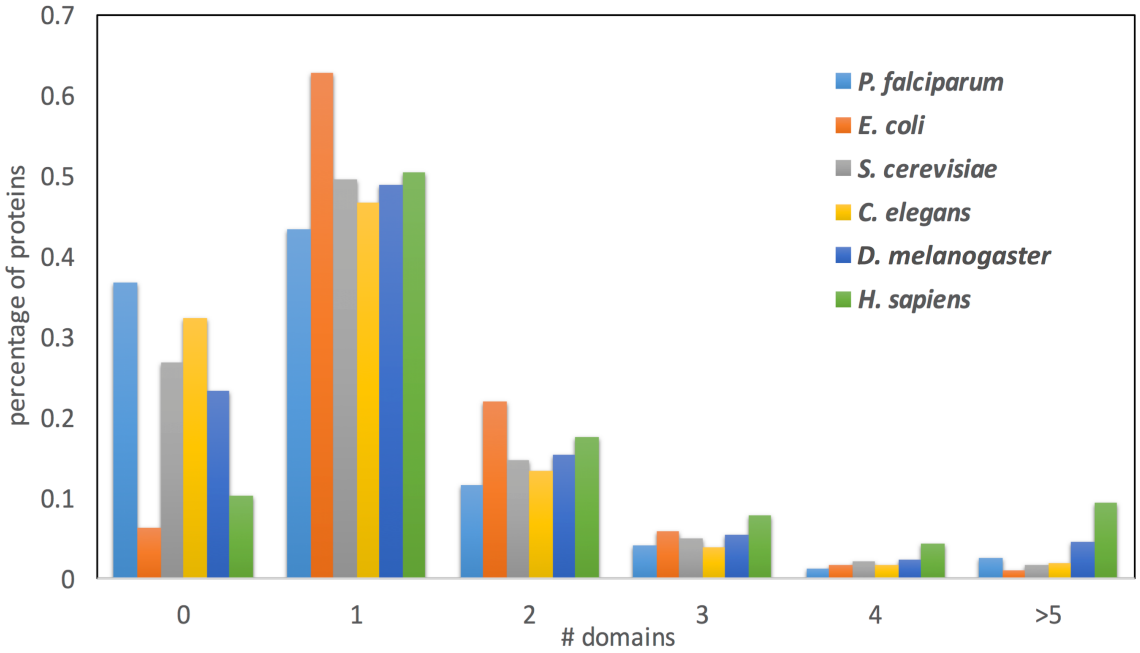

Figure A4: **Distribution of the number of domains present in tested organisms**

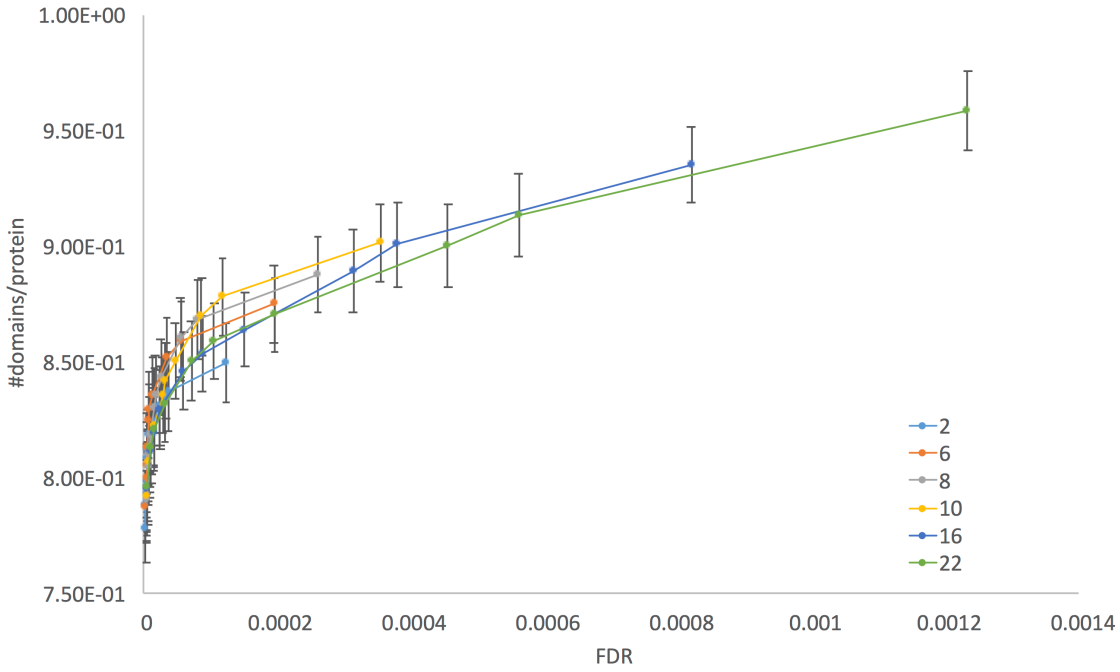

Figure A5: **Test of selecting scale factor for linker length scores.** FDRs in x-axis are the averaged values and the number of proteins per protein in y-axis is averaged values from ten groups. The error bars are standard deviations of the number of proteins per protein.

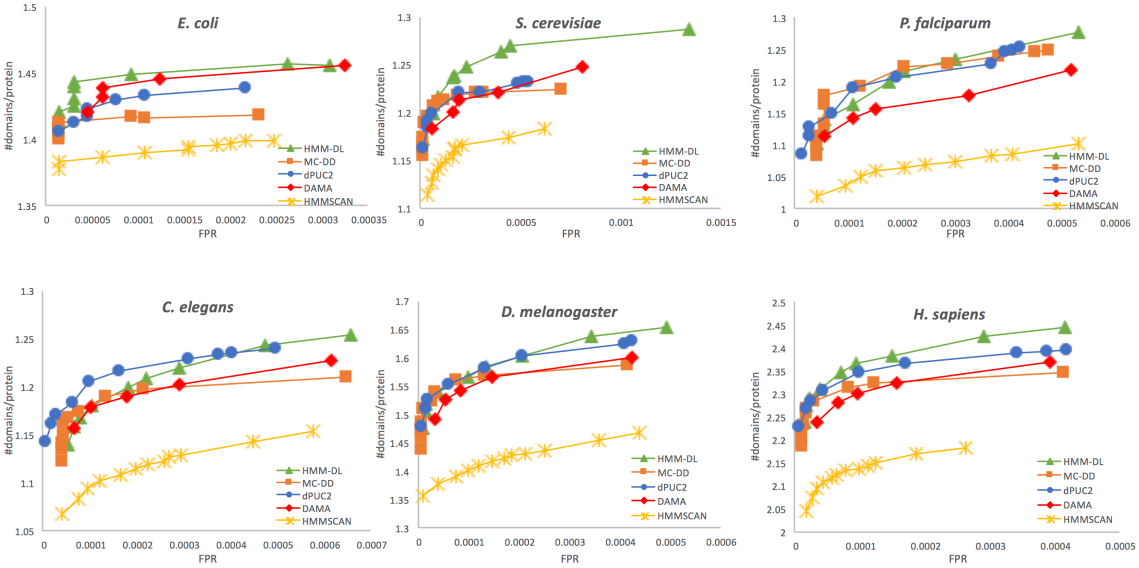

Figure A6: **Performance of HMMSCAN, dPUC2, DAMA, MC-DD and HMM-DL on domain identifications of *P. falciparum*, *E. coli*, *S. cerevisiae*, *D. melanogaster*, *C. elegans* and *H. sapiens* proteins.** The x-axis is the FPR and the y-axis is the number of predicted domains per protein. The better methods have higher curves.

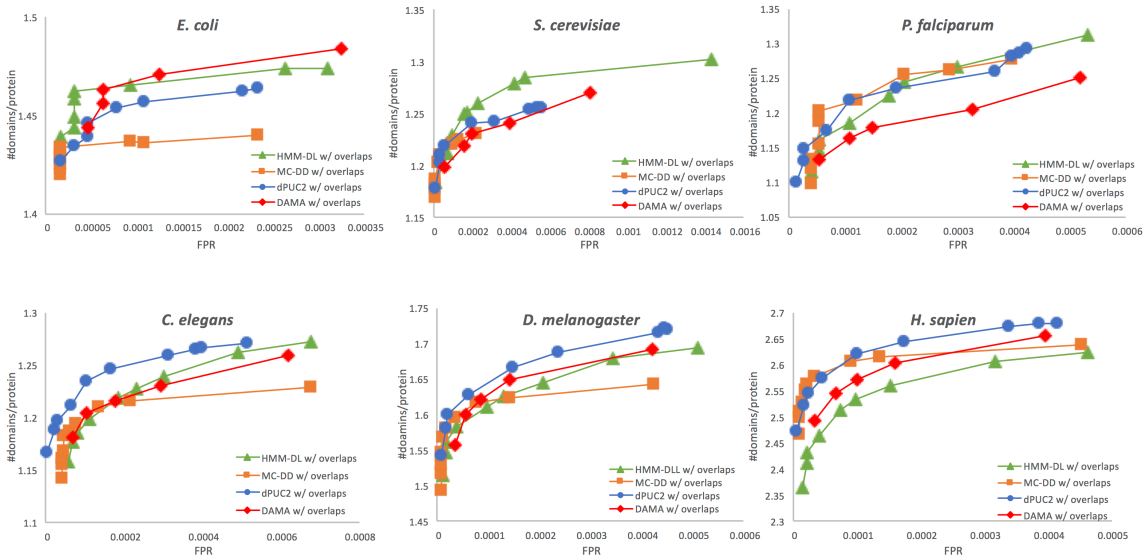

Figure A7: **Performance of HMMSCAN, dPUC2, MC-DD and HMM-DL with allowing overlaps on domain identifications of *P. falciparum*, *E. coli*, *S. cerevisiae*, *D. melanogaster*, *C. elegans* and *H. sapiens* proteins.** The x-axis is the FPR and the y-axis is the number of predicted domains per protein. The better methods have higher curves.

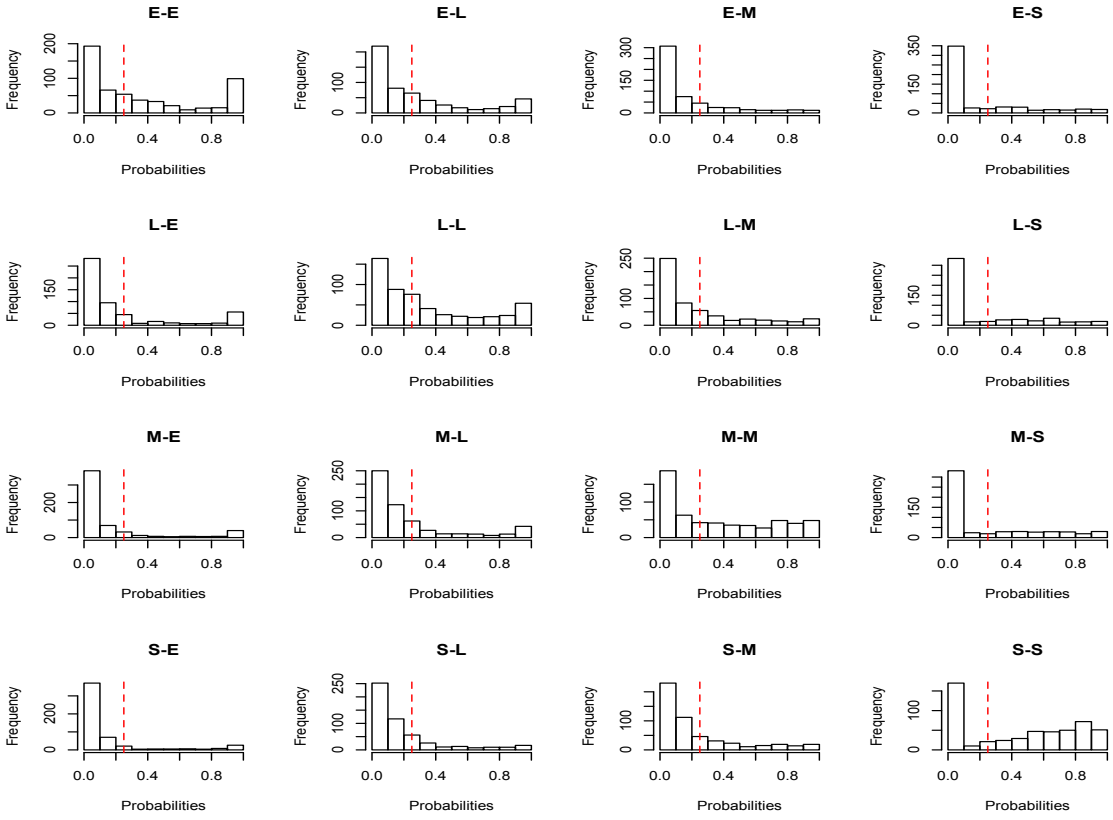

Figure A8: **Testing the conservation of linker-length dependencies for the specific domain triplets.**

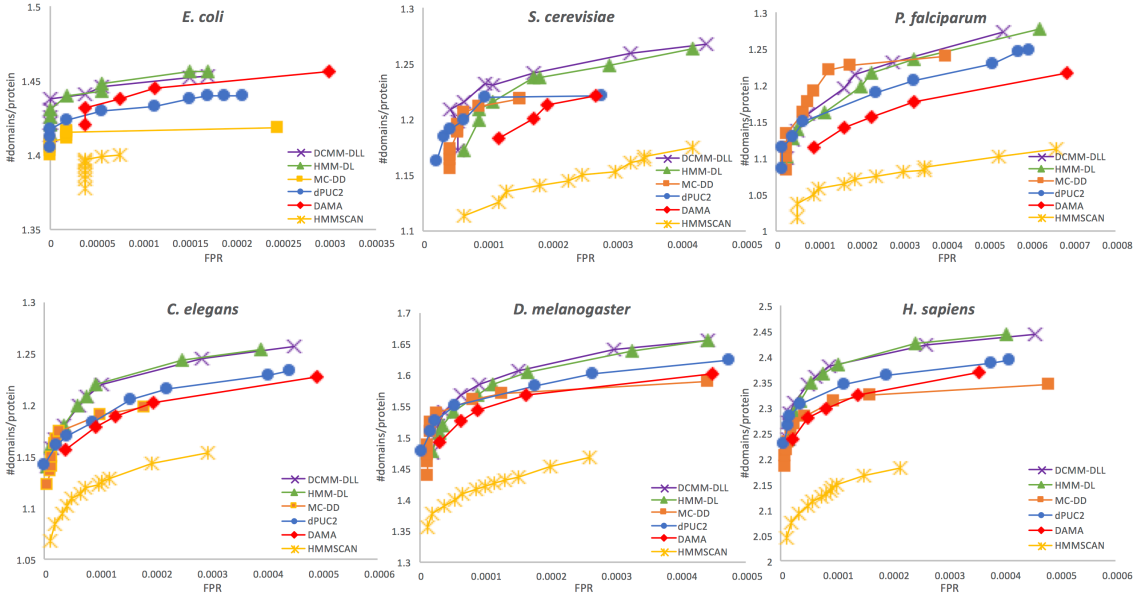

Figure A9: **Performance of HMMSCAN, dPUC2, MC-DD, HMM-DL and DCMM-DLL on domain identifications of *P. falciparum*, *E. coli*, *S. cerevisiae*, *D. melanogaster*, *C. elegans* and *H. sapiens* proteins.** The x-axis is the FPR and the y-axis is the number of predicted domains per protein. The better methods have higher curves.

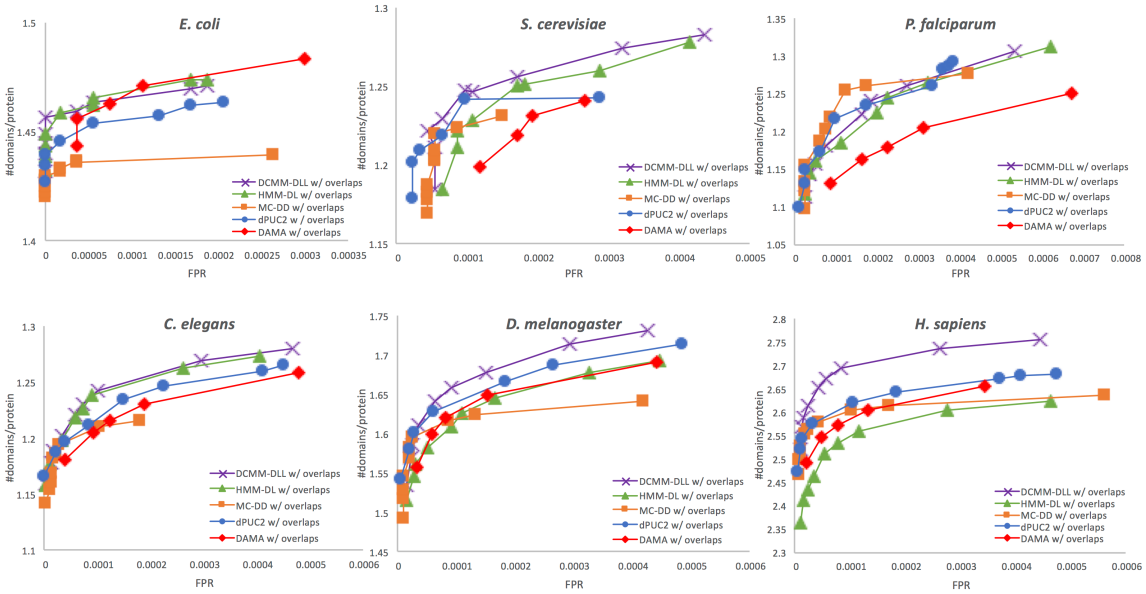

Figure A10: **Performance of HMMSCAN, dPUC2, MC-DD, HMM-DL and DCMM-DLL with allowed overlaps on domain identifications of *P. falciparum*, *E. coli*, *S. cerevisiae*, *D. melanogaster*, *C. elegans* and *H. sapiens* proteins.** The x-axis is the FPR and the y-axis is the number of predicted domains per protein. The better methods have higher curves.

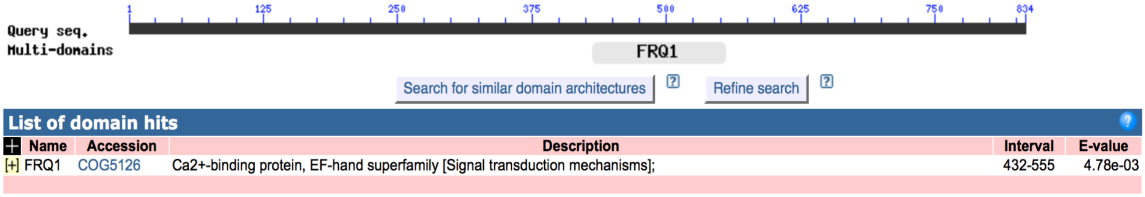

Figure A11: **Comparison of new functional annotation of PF3D7_1014800 from *P. falciparum* with annotation from pBlast.**