Sequence analysis

Domain prediction with probabilistic directional context

Alejandro Ochoa1,2 and Mona Singh1,3,*

1Lewis-Sigler Institute for Integrative Genomics, 2Center for Statistics and Machine Learning and 3Department of Computer Science, Princeton University, Princeton, NJ 08544, USA

*To whom correspondence should be addressed.

Received on December 15, 2016; revised on March 21, 2017; editorial decision on April 8, 2017; accepted on April 11, 2017

Abstract

Motivation: Protein domain prediction is one of the most powerful approaches for sequence-based function prediction. Although domain instances are typically predicted independently of each other, newer approaches have demonstrated improved performance by rewarding domain pairs that frequently co-occur within sequences. However, most of these approaches have ignored the order in which domains preferentially co-occur and have also not modeled domain co-occurrence probabilistically.

Results: We introduce a probabilistic approach for domain prediction that models ‘directional’ domain context. Our method is the first to score all domain pairs within a sequence while taking their order into account, even for non-sequential domains. We show that our approach extends a previous Markov model-based approach to additionally score all pairwise terms, and that it can be interpreted within the context of Markov random fields. We formulate our underlying combinatorial optimization problem as an integer linear program, and demonstrate that it can be solved quickly in practice. Finally, we perform extensive evaluation of domain context methods and demonstrate that incorporating context increases the number of domain predictions by \( \geq 15\% \), with our approach dPUC2 (Domain Prediction Using Context) outperforming all competing approaches.

Availability and Implementation: dPUC2 is available at http://github.com/alexviiia/dpuc2.

Contact: mona@cs.princeton.edu

Supplementary information: Supplementary data are available at Bioinformatics online.

1 Introduction

Protein domains are the functional and evolutionary units of proteins; they organize protein space and their identification helps annotate new protein sequences. Domains are often predicted in protein sequences using profile Hidden Markov models (HMMs, Eddy, 2008), each of which models one domain family and is constructed from sequence alignments of known instances. In the standard approach, statistical significance is evaluated for every domain prediction independently; this is how domains are identified in Pfam, the largest HMM domain database (Finn et al., 2010), by the state-of-the-art HMMER3 program (Eddy, 2008). Several recent methods have proposed to evaluate domains in the context of other domain predictions, using higher-order scoring systems and additional information; these include a Markov model of sequential domain co-occurrence (Coin et al., 2003), a protein clustering approach based on domain architectures that can discover missing domains (Beaussart et al., 2007), a domain prediction filter based on pairwise co-occurrence (CODD, Terrapon et al., 2009), and a multi-objective optimization that uses known domain architectures (DAMA, Bernardes et al., 2016a). Our previous approach, dPUC (Domain Prediction Using Context), maximizes the log-odds score of domain predictions and incorporates pairwise co-occurrence probabilities (Ochoa et al., 2011). These earlier approaches have demonstrated that context yields improvement in domain identification, but have only been evaluated in small tests, with a focus on the compositionally biased Plasmodium falciparum proteome.
It is well-known that domains tend to have order preferences within protein sequences (Apic et al., 2001); in Pfam 25 (Finn et al., 2010), we observe that 88.59% of domain family pairs occur in only one orientation (i.e. where a domain instance of one family is always found earlier in a protein sequence than a domain instance of the other family). However, previous context-based approaches either do not consider this directionality (Bernardes et al., 2016a; Ochoa et al., 2011; Terrapon et al., 2009) or only consider it for adjacent domain pairs (Coin et al., 2003) or in limited domain architecture alignments (Beaussart et al., 2007). In this work, we extend our earlier log-odds model for context-based domain prediction (Ochoa et al., 2011) to use directional context scores. We also newly show that our approach can be equivalently viewed as a Markov random field (MRF) (Kindermann and Snell, 1980), and explain its connection to the earlier Markov model of Coin et al. (2003). We introduce a new integer linear programming (ILP) formulation to solve the underlying optimization problem, and demonstrate that in practice solutions can be obtained fast enough for routine application. Last, we demonstrate superior performance of our approach, dPUC2, over all of its publicly available competitors in testing based on the UniProt database (Consortium, 2012); this is the largest-scale testing to date of context-based domain prediction. Overall, we find that the benefits of using domain context is substantial, with ~15% more domains predicted by dPUC2 than the Standard Pfam at the same level of estimated accuracy.

2 Models

2.1 The basic probabilistic model

Given a set of candidate domains, we develop an approach that identifies the most likely subset of domains under a probabilistic domain co-occurrence model. Let \( D = \{D_1, D_2, \ldots, D_k\} \) be \( k \) candidate domain instances ordered by start coordinates, which are predicted on a protein sequence \( X \) with a permissive \( P \)-value threshold (e.g. using HMMER3). Let \( F_i \) be the domain family of the domain instance \( D_i \) that denotes that domain instances are independent predictions, while domain families correspond to Pfam HMMs. Our goal is to identify \( D' \subseteq D \) that maximizes a score consisting of two log-odds components: an individual domain score and a pairwise domain context score. We additionally require that all domain instances \( D' \) are compatible with each other. In particular, we do not allow predicted domain instances to overlap by > 40 amino acids or >50% of the smaller of the two ranges; this is the ‘permissive overlap’ definition used earlier (Ochoa et al., 2015; Yeats et al., 2010).

The individual domain scores are derived from HMMER, which scores domain instance \( D_i \) as

\[
H_i = \log_2 \frac{\text{Pr}(X_i|F_i)}{\text{Pr}(X_i|R)},
\]

where \( X_i \) is the subsequence of the predicted domain, \( \text{Pr}(X_i|F_i) \) is the probability of \( X_i \) given the HMM of \( F_i \) (the alternative model), and \( \text{Pr}(X_i|R) \) is the probability of \( X_i \) under the null model \( R \) of independent and identically distributed amino acids (Eddy, 2008).

Pfam provides expert-curated ‘domain’ thresholds \( T^d_i \) for each family \( F_i \), where in order to predict a domain instance \( D_i \) of family \( F_i \), it must have a score \( H_i \geq T^d_i \). These thresholds have been previously interpreted as incorporating a prior probability (Coin et al., 2003); in order to better fit a log-odds framework, our interpretation differs slightly by including \( \text{Pr}(R) \), the probability that a random sequence is a naturally occurring protein subsequence.

Here, if \( \text{Pr}(F_i) \) and \( \text{Pr}(R) \) are prior probabilities for domain family \( F_i \) and the background model \( R \), respectively, then the domain threshold is viewed as

\[
T^d_i = -\log_2 \frac{\text{Pr}(F_i)}{\text{Pr}(R)}.
\]

In this case,

\[
H_i - T^d_i = \log_2 \frac{\text{Pr}(X_i|F_i)}{\text{Pr}(X_i|R)} - \log_2 \frac{\text{Pr}(F_i)}{\text{Pr}(R)} = \log_2 \frac{\text{Pr}(F_i|X_i)}{\text{Pr}(R|X_i)}
\]

corresponds to a posterior log-odds score. We note that in practice Pfam sets the values of the domain thresholds \( T^d_i \) by manual cur- ation, without modeling \( \text{Pr}(F_i) \) or \( \text{Pr}(R) \) explicitly.

In addition to individual domains, we also incorporate ‘context’ scores between domain families as

\[
S_{ij} = \log_2 \frac{\text{Pr}(F_i, F_j)}{\text{Pr}(F_i)\text{Pr}(F_j)}
\]

where \( \text{Pr}(F_i, F_j) \) is the probability that a domain of family \( F_i \) appears earlier than one of family \( F_j \) in the protein sequence, and \( \text{Pr}(F_i) \) is the prior probability of observing domains of \( F_i \).

Our goal is to find \( D' \) to maximize

\[
\sum_{i D_i \in D'} \left( H_i - T^d_i + \sum_{j > i \in D'} S_{ij} \right),
\]

which is equivalent to

\[
\log_2 \prod_{i D_i \in D'} \left( \frac{\text{Pr}(F_i|X_i)}{\text{Pr}(R|X_i)} \prod_{j < i \in D'} \text{Pr}(F_j, F_i) \right).
\]

Hence, our objective is a log-odds score with two parts: the fit of each family to the sequence, and of family pairs to the co-occurrence model. Each part is a ratio of corresponding probabilities from null and alternative models. Note that no domains are predicted if the maximum value of the objective function is <0, since in this case the null model has the highest probability.

2.2 Formulation as an MRF

We can also equivalently view our problem within the context of undirected graphical models or MRFs (Kindermann and Snell, 1980). A configuration \( x = (x_1, \ldots, x_N) \) in our system consists of indicator variables \( x_i \in \{0, 1\} \) for whether domain instance \( i \) is predicted or not. Let \( X \) be the set of configurations without conflicting domains. The following probability distribution can be defined over the configurations \( x \in X \):

\[
\text{Pr}(x) = \frac{1}{Z} e^{S(x)}, \text{ where}
\]

\[
S(x) = \sum_{i=1}^{k} (x_i(H_i - T^d_i) + \sum_{j > i} x_i x_j S_{ij}),
\]

\[
Z = \sum_{x \in X} e^{S(x)}.
\]

Here, the score \( S(x) \) is treated as the negative of the energy of \( x \) and \( Z \) is the partition function that ensures that \( \sum_{x \in X} \text{Pr}(x) = 1 \).
solution \( x \in \mathcal{X} \) that maximizes \( \Pr(x) \) is equivalent to the one that maximizes the log-odds score outlined above as \( Z \) is a normalization constant shared by all \( x \).

### 2.3 Comparison to the Markov model of Coin et al.

Coin et al. (2003) derived a Markov approximation for the joint probability of domain predictions given the sequence and the model. Using our notation and ignoring transitions from the ‘begin’ and to the ‘end’ states, for a given set of domains, \( D_1 \ldots D_k \), the first-order Markov score is

\[
\sum_{i=1}^{k} \log_2 \frac{\Pr(X_i|F_i)}{\Pr(X_i|R)} \sum_{j=1}^{k} T^d_{ij} + \sum_{i=1}^{k} \log_2 \frac{\Pr(F_i|F_{i-1})}{\Pr(F_i)}
\]

\[
= \sum_{i=1}^{k} \left( (H_i - T^d_{ij} + S_{ij}) \right), \quad \text{where}
\]

\[
S_{ij} = \log_2 \frac{\Pr(F_{i-1}, F_i)}{\Pr(F_{i-1}) \Pr(F_i)}
\]

is our context score for consecutive domains (\( S_{ij} = 0 \) is set for convenience). Hence, our objective differs from the Markov model of Coin et al. (2003) by

\[
\sum_{i=1}^{k-2} \sum_{j=i+2}^{k} S_{ij}.
\]

Notably, both scores agree for \( k = 2 \) domain predictions, and for \( k = 3 \) the difference is only \( S_{ij} \). Our approach has the advantage over the Markov model of enforcing consistency between all domain pairs.

### 2.4 Modifications due to Pfam thresholds

In addition to domain thresholds \( T^d_{ij} \), the Standard Pfam approach also uses expert-curated ‘sequence’ thresholds \( T^s_{ij} \). Let \( D_F \) denote the subset of predictions within a protein that are domain instances of family \( F \). These predicted domain instances must satisfy

\[
\sum_{i \in D_F} H_i \geq T^s_F.
\]

For 97.7% of Pfam families (Punta et al., 2011), \( T^s_F = T^d_F \); in other words, the sequence thresholds can be ignored since by satisfying the individual domain thresholds, the sequence thresholds are also satisfied. However, in order to make our model consistent with Pfam for the remaining 2.3% of Pfam families, we modify our model to include the term

\[
\tilde{T}_F = T^s_F - T^d_F,
\]

despite the fact that these sequence thresholds do not have a probabilistic interpretation.

Our objective then becomes to find \( D' \) that maximize

\[
\sum_{i \in D \subset D'} (H_i - T^d_{ij}) + \sum_{j \in D \subset D'} \tilde{T}_{ij} - \sum_{j \in D} \tilde{T}_{ij},
\]

where every predicted family is included once in the sum over \( F \).

Like previous models (Coin et al., 2003; Ochoa et al., 2011), our context-based approach is designed to agree with the Standard Pfam in the absence of context (all \( S_{ij} = 0 \)), where the objective becomes

\[
\sum_{i \in D \subset D'} (H_i - T^d_{ij}) + \sum_{j \in D} \tilde{T}_{ij} \geq 0.
\]

Each \( D_i \) in the optimal solution satisfies \( H_i \geq T^d_{ij} \), since otherwise including \( D_i \) would reduce the objective. By the same reasoning, each family \( F \) in the optimal solution satisfies its sequence threshold, since

\[
\sum_{i \in D \subset D'} H_i \geq \tilde{T}_F + \sum_{i \in D \subset D'} T^d_{ij} = \tilde{T}_F + |D'_F| \cdot T^d_F = \tilde{T}_F + (|D'_F| - 1) T^d_F \geq T^s_F.
\]

The sequence threshold is set exactly for \( T^d_F = 0 \) or \( |D'_F| = 1 \), which are the most common cases. Otherwise (\(|D'_F| > 1 \) and \( 0 < T^d_F < T^d_F \)), our approach’s effective sequence threshold is more stringent than the Standard Pfam’s, since the last inequality above becomes a strict inequality.

### 3 Materials and methods

#### 3.1 Context score estimation

We use a directed ‘context network’ of family pair counts \( c_{ij} \) of domain instances of family \( F_i \) that occurred before instances of family \( F_j \) in a large database. Self-pairs \( (F_i = F_j) \) model repeating domains. Pair probabilities, corresponding to the proportion of all domain pairs that are of a given family pair, are estimated using a symmetric Dirichlet prior with parameter \( \alpha \), yielding

\[
\tilde{p}_{ij} = \frac{c_{ij} + \alpha}{C + 2\alpha n},
\]

where \( n \) is the number of domain families (there are \( n^2 \) directed family pairs, including self-pairs) and \( C = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} \) sums over family pairs. All families have equal prior probabilities \( \tilde{p}_i = \frac{1}{n} \); we note that the Bayesian interpretation of Pfam thresholds outlined earlier would set different values of \( \Pr(F_i) \), but these equal priors are simple and work well in practice. These estimates satisfy \( \sum_{i=1}^{n} \sum_{j=1}^{n} \tilde{p}_{ij} = 1 \) and \( \sum_{i=1}^{n} \tilde{p}_i = 1 \). The context score for a domain instance of family \( F_i \) preceding an instance of family \( F_j \) is estimated as

\[
\log_2 \frac{\tilde{p}_{ij}}{\tilde{p}_j}.
\]

Note that unobserved family pairs \( (c_{ij} = 0) \) receive a fixed penalty of

\[
\log_2 \left( \frac{2n^2}{C + 2n^2} \right) < 0.
\]

Counts \( c_{ij} \) are obtained from domains predicted by Pfam on UniProt, as provided in the file Pfam-A.full. We remove \( c_{ij} = 1 \) cases at load time, which are assumed to be incorrect. We set \( \alpha = 10^{-3} \), which we find to perform better than the uniform Dirichlet prior where \( \alpha = 1 \).

#### 3.2 Algorithms

We find optimal domain architectures with two strategies: ‘positive elimination’ (PE), followed by ILP. PE discards domains with poor HMMER3 and context scores, returning trivial solutions (see below) or passing a smaller set to the ILP solver. The combinatorial ILP problem is solved using the lp_solve C library (Berkelaar et al., 2004); though ILP is NP-complete, in practice our optimization problem is solved quickly compared with the time taken for
predicting the initial candidate domains using a permissive threshold via HMMER3 (see section 4.2 below).

3.2.1 Positive elimination
PE consists of the following iterative procedure (Ochoa et al., 2011). At a given iteration, remaining domain instances $D_i$ are evaluated using

$$S^+_i = H_i - T^+_F + \sum_{j \neq i} S^+_j + \sum_{j \neq i} S^+_j,$$

where only positive context scores $S^+_i = \max(0, S_{ij})$ are included between $D_i$ and all remaining $D_j$, and additionally $S^+_i = 0$ for conflicting domains $D_i, D_j$. Domain $D_i$ is eliminated if $S^+_i < 0$, and all instances of family $F$ are eliminated if $\sum_{i \in F} S^+_i < T_F$. Note that eliminated $D_i$ would contribute negatively to our objective, even when every positive context score is included. Iterations end when there are no more eliminations.

After PE, the remaining domains are tested for conflicts or negative context: if both are absent, then these domains are the trivial solution and are returned directly, skipping the ILP. This follows since there are no more eliminations.

At a given iteration, remaining domain instances $D_i$ are predicted, respecting the cliques of all candidate families. We wish to find values for all $D_i, D_j$, and each $S_{ij}$.

3.2.2 Integer linear program
Let $x_i, x_{ij}$, and $x_F$ be indicator variables for whether domain instances $D_i$, pairs $D_i, D_j$, and families $F$ are predicted, respectively. The objective function is

$$\sum_{i=1}^k \{x_i \cdot (H_i - T^+_F) + \sum_{j \neq i} x_{ij} S_{ij} \} - x_F T_F,$$

where $k$ is the number of candidate domains and the sum over $F$ covers all candidate families. We wish to find values for all $x_i, x_{ij}$, and $x_F$ such that

$$x_i, x_{ij}, x_F \in \{0, 1\} \quad \forall i, j, F,$$

$$0 \leq x_i, x_{ij}, x_F \leq 1 \quad \forall i, j, F,$$

$$x_{ij} \leq x_i \quad \forall i, j,$$

$$x_{ij} \leq x_{ij} \quad \forall i, j,$$

$$x_{ij} \geq x_i + x_j - 1 \quad \forall i, j,$$

$$x_F \geq x_i \quad \forall F, i : D_i \in D_F,$$

$$x_F \leq \sum_{i \in D_F} x_i \quad \forall F,$$

$$\sum_{i \in D_F} x_i \leq 1 \quad \forall C,$$

where $D_F$ is the set of candidate domains of family $F$, and each $C$ is a set of domains where all pairs conflict, and such that all conflicts among all possible domain instances are covered by all $C$.

We note that ILPs are solved in two steps: first, the fast ‘LP relaxation’ step solves the optimization for real-valued variables, and next a more expensive branch-and-bound algorithm is used if the previous solution is non-integral (Berkelaar et al., 2004). In our ILP formulation, the LP relaxation corresponds to dropping constraint (1) only, whereas in the full ILP that includes constraint (1), constraint (2) is implied by constraint (1).

In the ILP, constraints (3)–(5) correspond to the ‘AND’ operation of $x_{ij} = x_i \land x_j$, where the pairwise domain variable $x_{ij}$ is set to 1 if and only if both domains $D_i$ and $D_j$ are chosen. This is the tightest set of constraints implementing this requirement, as the four allowed settings of these variables are vertices on the resulting polytope (Fig. 1A). In contrast, Figure 1B shows that an earlier formulation of the ‘AND’ constraint (Ochoa et al., 2011) is not as effective since some of the vertices of the polytope defined by these constraints are non-integral; this increases the chance that non-integer solutions will be found for these variables at the LP relaxation stage, ultimately resulting in increased average runtimes. Constraints (6) and (7) in the ILP formulation correspond to the ‘OR’ operation of $x_F = \bigvee_{D_i \in D_F} x_i$, where the family variable $x_F$ is set to 1 only if at least one domain instance of that family is chosen. Lastly, constraint (8) in the ILP ensures that, out of a set of domains $C$ that all conflict with each other, only one is chosen ($\bigvee_{i \in D_F, D_j \in C} (x_i \land x_j)$). In practice, the cliques $C$ above are found with a greedy clique-finding heuristic, returning $C$ that are not nested but may overlap, and cover all conflicting domain pairs.

3.3 Evaluation and comparison to other methods
3.3.1 Pfam database
All tests in the main body of this article use Pfam 25 (12,273 HMMs; see the Supplementary Material for testing with the more recent Pfam 30). Pfam provides curated thresholds, clan definitions (which group related families), and a list of nesting families that may overlap without conflicting. Family pair counts $c_{ij}$ are calculated using UniProt 2010_05 (113,844 proteins), as provided in Pfam-A.full. Since sequence thresholds are trivial when $T^+_F = T^+_F$, in these cases dPUC uses only the domain thresholds (see section 2.4). For simplicity, sequence thresholds are additionally ignored for families $F$ for which $T^+_F \geq \frac{3}{2} T^+_F$ (67 families); we note that this leaves only 163 families where sequence and domain thresholds differ.

3.3.2 Hmmscan parameters
Domains are predicted with hmmscan from HMMER 3.0. We used: ‘-F1 1e-1 --F2 1e-1 --F3 1e-2’ to set ‘stage 1/2/3’ P-value thresholds

![Fig. 1. (A) Geometric view of the relaxed LP constraints of Equations (3–5). All $x_i, x_j$ and $x_{ij}$ are constrained by Equation (2) to the unit cube shown. Equation colors match their translucent surfaces, and the blue surface is behind the others. The vertices of the polytope resulting from the intersection of these three constraints and the cube arise at integral values of $x_i, x_j$ and $x_{ij}$ (green crosses). (B) Previously, we described another ILP (Ochoa et al., 2011) that consisted of two constraints relating $x_i$, $x_j$, and $x_{ij}$; that formulation is weaker than our current one as its polytope (the space between the two planes and inside the unit cube) is larger and has undesirable fractional vertices (red crosses) along with integral vertices (green crosses).]
and ‘-Z 1 --domZ 1’ to return P-values instead of E-values. For testing our approach against others and computing false discovery rates (FDRs) across a range of noise levels, we used ‘-E 1e-2 --domE 1e-2’ to set a permissive P-value threshold. For measurements of runtime, we used ‘-cpus 0’ to use a single thread (no parallel worker threads), and ‘-E 1e-4 --domE 1e-4’ to set final P-value thresholds, as this is how our tool dPUC2 would be used in practice.

3.3.3 FDR tests

We previously introduced five domain prediction FDR tests based on different definitions of true and false positives (TPs, FPs) (Ochoa et al., 2015). Briefly, each FDR test labels domain predictions as TPs or FPs (as described below). For a given domain prediction method and each protein i in the test, let TPi and FPi be the number of predicted domains that are labeled as TPs and FPs, respectively. The estimated FDR of the domain prediction method is

$$\text{FDR} = \frac{1}{m} \sum_{i=1}^{m} \frac{\text{FP}_i}{\text{TP}_i + \text{FP}_i},$$

where the average includes only proteins i with domain predictions (TPi + FPi > 0). Two FDR tests below generate FP domains from random or reversed protein sequences (ensuring that these are truly domains as belonging to one sequence (Fig. 2A). Selected domains to set final P-value thresholds, as this is how our tool dPUC2 would be used in practice.

3.3.4 Comparisons to other approaches

We first compare how well our approach identifies domains as compared with Pfam when using several different approaches for setting thresholds. The ‘Standard Pfam’ predicts domains using the Pfam domain and sequence thresholds, but removing domain instances of the same clan that overlap strictly (one or more amino acids), keeping only the most significant prediction per conflict (Punta et al., 2011). To get a range of FDRs using Pfam, we ‘extend’ the Standard Pfam thresholds by shifting domain and sequence thresholds by constant bit amounts (Ochoa et al., 2011). As an alternative approach, we also consider a version of Pfam where predictions are considered as E-value thresholds are varied. Both Extended Standard Pfam and E-values remove conflicting domains (defined as permissive overlaps, see section 2) ranking by P-value.

We also compare the performance of our method to three previous methods that use domain context. First, DAMA (Bernades et al., 2016a) was downloaded from http://www.kcib.upmc.fr/ DAMA and used with default Pfam 25 inputs. DAMA combines E-values with discrete rewards for previously observed domain architectures and other criteria (Bernades et al., 2016a). Next, we reimplemented the rule-based system CODD (Ghouila et al., 2014; Terrapón et al., 2009), as a version using Pfam 25 and UniProt is not publicly available. CODD uses a list of ‘certified domain pairs’, a filtered binary context network, to predict low-confidence domains that are supported by context with high-confidence domains (Terrapón et al., 2009). For CODD, certified domain pairs are defined (Terrapón et al., 2009) as Pfam 25 domain pairs occurring in UniProt that significantly co-occur ($P < 0.01$ using the Hypergeometric distribution). Lastly, we also test our previous dPUC1 approach, which does not take into account directional context. In this case, due to runtime limitations, it is not feasible to run dPUC1 directly in large-scale testing; instead, the dPUC1 objective is solved using the dPUC2 ILP outlined here. We did not test the methods of Coin et al. (2003) and Beausset et al. (2007), as they are not publicly available.

To compare the context methods, we first use a permissive P-value threshold of $P < 1e-2$ to get a set of candidate domains for each protein sequence. Next, we vary the P-value threshold between 1e-9 and 1e-3; in total, 25 thresholds are sampled roughly evenly in
logarithmic space \(10^{-3} \) and \( a \times 10^{-b} \) for \( a \in \{1, 2, 4, 6\} \) and \( b \in \{4, 5, 6, 7, 8, 9\} \). We run DAMA with these domain predictions and each P-value threshold. For dPUC2, dPUC1 and CODD, the input is the set domains that are predicted by HMMER at each threshold, along with all Standard Pfam domain predictions. Finally, we calculate FDRs for each method using each of the three tests, and consider as a function of FDR, the change in number of domains as compared with the Standard Pfam.

4 Results

4.1 DPUC2 outperforms competitors in FDR tests

We compare the performance of dPUC2, our tool that implements the probabilistic directional domain context approach described in this article, to five other approaches for domain identification. Our benchmarks estimate the FDR of each domain prediction method by assigning a TP or FP label to each predicted domain: in RevSeq all domains that are predicted on the reversed sequence are FP, in MarkovR the domains predicted on the random sequence are FP, and in OrthoC all domains from clans absent in orthologs are FP; in all tests, domains that are not FP are labeled as TP (see Fig. 2 and section 3.3.3 for additional details).

We begin by observing that across all three tests, context methods improve substantially over the Standard Pfam (Fig. 3). For example, dPUC2 predicts roughly 15% more domains than the Standard Pfam at the same empirical FDR. We note that all context methods improve upon the uncurated version of Pfam, where E-value thresholds (or P-value thresholds, equivalently) are simply varied, across the entire range of estimated FDRs (E-value in Fig. 3). In practice, however, Pfam uses curated domain and sequence bitscore thresholds (Pfam in Fig. 3); if these bitscores are varied instead (see section 3.3.4; PfamExt in Fig. 3), the context method DAMA is outperformed by Pfam at low FDRs in two of our three tests.

We find that dPUC2 outperforms the previous context-based methods in each of the three tests (Fig. 3). Notably, in all three tests, dPUC2 predicts more domains than CODD across the entire range of estimated FDRs. CODD does not penalize domain combinations that have not been observed before (i.e. it has no negative context) and this contributes to its higher FDRs. CODD also predicts fewer domains, in part because no domains can be predicted by CODD in sequences that lack Standard Pfam predictions (Terrapon et al., 2009). Our approach dPUC2 also consistently predicts more domains across the range of FDRs than the context method DAMA in the RevSeq and OrthoC tests. In the MarkovR test, when very low P-value thresholds are used to identify candidate domains (P-value < 6e-7), dPUC2 identifies more domains than DAMA but also results in higher estimated FDRs; this is due in part to dPUC2 keeping all Standard Pfam predictions in its input (see below). Thus, while overall dPUC2 performs better than DAMA, when using highly confident domain predictions, dPUC2 and DAMA trade off coverage and noise in the MarkovR test. Finally, in every FDR test, dPUC2 also improves upon dPUC1, illustrating the benefit of incorporating directional context. Directional context reduces the FDR of predictions produced for the same inputs; for example, when using candidate domains with P < 1e-4, dPUC2 results in a lower FDR than dPUC1 (blue and red crosses in Figure 3). Although the results described here are based on Pfam 25, dPUC2 also outperforms all competing approaches when using the newer Pfam 30 (see Supplementary Material and Supplementary Fig. S1).

We note that the dPUC2, dPUC1 and CODD methods always keep the Standard Pfam predictions in the input, so the candidate P-value threshold is applied to the non-Standard Pfam predictions only. For this reason, as more stringent thresholds are used to identify candidate domains, these methods perform similarly to the Standard Pfam predictions (dark gray cross in Fig. 3), differing only by how they remove predictions using negative context. On the other hand, for stringent P-value thresholds, DAMA removes those Standard Pfam predictions that are not predicted at that level of confidence. Therefore, DAMA performs similarly to the E-value curve when the P-value threshold is stringent, and differences between DAMA and the other context methods at low P-value thresholds are affected by the differences in the way the approaches handle Standard Pfam predictions. However, DAMA also incurs high FDRs at larger P-value thresholds which would include Standard Pfam predictions; e.g. when using P < 1e-4 to identify candidate domains, DAMA’s FDR is higher than those for the other context methods in

---

**Fig. 3.** Our new approach, dPUC2, predicts more domains than its competitors across a wide range of FDRs, as estimated by the RevSeq (left), MarkovR (middle), and OrthoC (right) tests. The dark gray cross gives the FDR of the Standard Pfam, and the changes in the number of domain predictions for all methods at different FDRs are given with respect to the number of Standard Pfam predictions. For reference, we highlight with crosses the performances of dPUC2, dPUC1, CODD and DAMA when run on candidate domains identified with HMMER at P < 1e-4.
all three tests (purple cross in Fig. 3). Thus, the superior overall performance of dPUC2 as compared with DAMA is due to a variety of methodological factors.

### 4.2 Accelerated performance

DPU2C is optimized to be much faster than dPUC1. We compare the dPUC1 and dPUC2 runtime using candidate Pfam domain predictions with \( P < 1 \times 10^{-4} \) on the *Homo sapiens* proteome (UniProt UP000005640, dated 2015-06-07, 69,693 proteins). Using a 3.2 GHz processor with 16 GB RAM, we find that the HMMER3 runtime averages 5.5e-3 s/protein, while dPUC1 averages 0.291 s/protein (Fig. 4). Therefore, dPUC2 is 53 times faster than dPUC1 on average, and amounts to 1.21% of HMMER3 runtime. Since the runtime distributions are highly skewed, we also compare median runtimes: the dPUC2 median is 1.11e-4 s/protein, while the dPUC1 median is 0.0282 s/protein; i.e. the dPUC2 median run is 255 faster than dPUC1’s. Additionally, dPUC2 is faster than dPUC1 in 99.2% of individual proteins. Hence, dPUC2 is orders of magnitude faster than dPUC1 and its runtime is negligible compared with that of initially identifying candidate domains via HMMER3.

### 5 Discussion

In this work, we introduce dPUC2, the first domain context prediction method that models directional domain context between all domain pairs, and perform the largest-scale evaluation to date of domain context methods. Using large and unbiased protein sets and several different criteria to estimate the FDR, we demonstrate that the HMMER3 step that predicts candidate domains averages 0.455 second per protein (s/protein) in wallclock runtime when run in a single thread; this is comparable to a previous reported average of \( \sim 1 \) s/protein (Finn et al., 2010). The dPUC2 step (excluding the HMMER3 runtime) averages 5.5e-3 s/protein, while dPUC1 averages 0.291 s/protein (Fig. 4). Therefore, dPUC2 is 53 times faster than dPUC1 on average, and amounts to 1.21% of HMMER3 runtime. Since the runtime distributions are highly skewed, we also compare median runtimes: the dPUC2 median is 1.11e-4 s/protein, while the dPUC1 median is 0.0282 s/protein; i.e. the dPUC2 median run is 255 faster than dPUC1’s. Additionally, dPUC2 is faster than dPUC1 in 99.2% of individual proteins. Hence, dPUC2 is orders of magnitude faster than dPUC1 and its runtime is negligible compared with that of initially identifying candidate domains via HMMER3.

### Acknowledgements

We thank Manuel Linán for helpful discussions about this work.

### Funding

This work has been supported by the National Science Foundation [Graduate Research Fellowship DGE 0646086 to A.O., ABI 1062371 to M.S.]; and the National Institutes of Health [R21-01085415 and R01-GM076275 to M.S.].

**Conflict of Interest**: none declared.

### References

Apic, G. et al. (2001) Domain combinations in archaeal, eubacterial and eu-karyotic proteomes. *J. Mol. Biol.*, 310, 311–325.

Beaussart, F. et al. (2007) Automated Improvement of Domain ANnotations using context analysis of domain arrangements (AIDAN). *Bioinformatics*, 23, 1834–1836.

Berkelaar, M. et al. (2004). lp_solve: Open source (Mixed-Integer) Linear Programming system. http://lpsolve.sourceforge.net.

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

Bernardes, J., et al. (2016b) Improvement in protein domain identification is reached by breaking consensus, with the agreement of many profiles and domain co-occurrence. *PLoS Comput. Biol.*, 12, e1005038.

---

**Fig. 4.** Distribution of wallclock runtimes on human protein sequences using dPUC1 or dPUC2 on a 3.2 GHz processor. A candidate domain threshold of \( P < 1 \times 10^{-4} \) was used in both cases. Density is for the log of the runtime.
Chen, F. et al. (2006) OrthoMCL-DB: querying a comprehensive multi-species collection of ortholog groups. *Nucleic Acids Res.*, 34(Database issue), D363–D368. PMC1347485.

Coin, L. et al. (2003) Enhanced protein domain discovery by using language modeling techniques from speech recognition. *Proc. Natl. Acad. Sci. USA*, 100, 4516–4520. PMC404693.

Consortium, T.U. (2012) Reorganizing the protein space at the Universal Protein Resource (UniProt). *Nucleic Acids Res.*, 40, D71–D75.

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

Finn, R.D. et al. (2010) The Pfam protein families database. *Nucleic Acids Res.*, 38(suppl. 1), D211–D222.

Ghouila, A. et al. (2014) Identification of divergent protein domains by combining HMM-HMM comparisons and co-occurrence detection. *PLoS One*, 9, e95275.

Kindermann, R., and Snell, L. (1980). *Markov Random Fields and Their Applications*. Amer Mathematical Society, Providence, Rhode Island, USA.

Ochoa, A. et al. (2011) Using context to improve protein domain identification. *BMC Bioinformatics*, 12, 90.

Ochoa, A. et al. (2015) Beyond the E-value: stratified statistics for protein domain prediction. *PLoS Comput. Biol.*, 11, e1004509.

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

Storey, J.D. (2003) The positive false discovery rate: a Bayesian interpretation and the q-value. *Ann. Stat.*, 31, 2013–2035.

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.

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