Conditional Random Fields for Classification of Protein Families: An Alternative to Hidden Markov Models

Thomas J. Emerson

Abstract. Classification of a protein into a family of related proteins on the basis of its amino acid sequence is frequently done via a probabilistic model, usually a hidden Markov model (HMM). However, there are a variety of reasons - based on general modeling issues, the statistical properties of protein sequences, or biological considerations - that suggest that HMMs may not be the best type of probabilistic model to use for this classification task. In this paper we examine some issues in the use of HMMs as protein family profilers, and propose the use of another type of probabilistic model for this problem, namely the conditional random field (CRF); we also outline the the design of a CRF model to be used as a protein family classifier.

1. Protein Families and Sequence Classification

A protein family may be defined (loosely) as a group of proteins with similar biochemical function and a high degree of sequence identity when aligned. The classification of a protein's residue sequence as a member of a particular family is an important goal of bioinformatics; as Borodovsky and Ekisheva say ([BD], p.126),

> ... it is important to develop efficient computational tools able to assign a protein from a newly predicted gene to one of [the] already established families, thus characterizing the protein based on its amino acid sequence alone.

They go on to enumerate the desiderata for a protein family classification algorithm:

> The computational tools that are required to solve the classification problem should be able to: (i) make use of known structural patterns specific for a given family, (ii) detect the family patterns in the new protein sequence by alignment of the new protein to the family model, and (iii) assess the statistical significance of the detected similarity in order to help correctly identify the true family members.

Currently, membership in a protein family is usually determined in one of two ways: first, by a multiple sequence alignment, or second, by being scored with a probabilistic model. Overwhelmingly the type of probabilistic model chosen to model a protein family has been the hidden Markov model, or HMM.

However, we shall argue in this paper that, despite their widespread acceptance, there are cogent reasons why hidden Markov models are not the most suitable class of probabilistic models for the protein family classification task. Instead, we propose the use of a type of probabilistic model of relatively recent formulation, the *conditional random field (CRF)* model, which, we believe, offers several advantages over hidden Markov models; and we shall suggest how the design and implementation of a CRF model for protein sequence classification might be carried out.

2. Hidden Markov Models and Family Classification

We begin by reviewing the use of HMMs to profile protein families.

2.1. Definition of HMMs. Let us recall the formal definition of a hidden Markov model, beginning with the definition of a Markov chain.

A *Markov chain* is a countable, discrete-valued stochastic process $\{Y_1, Y_2, ...Y_n...\}$ such that for each n we have

$$P(Y_n|Y_1, Y_2, ...Y_{n-1}) \;=\; P(Y_n|Y_{n-1}) \tag{2.1}$$

that is, each Y_i is dependent only on the immediately preceding Y.

Thus, we think of a Markov chain as a stochastic process which can assume one of a number of states chosen from a finite set $S = \{s_1, s_2, ...s_n\}$ and which can change from one state to another at each $t = 1, 2, ...$ It is usually assumed that this process is stationary, i.e., that the probabilities of transition from one state to another at time t remain constant over t.

A *hidden Markov model* consists of two countable discrete-valued stochastic processes $\{Y_i\}$ and $\{X_i\}$ such that

(1) $\{Y_i\}$ is a Markov chain;
(2) $P(X_n|Y_1, Y_2, ...Y_{n-1}, X_1, ...X_{n-1}) \;=\; P(X_n|Y_n)$ i.e., X_i is dependent only on Y_i.

Informally, we think of each value of the stochastic process X as an observation, an element of an output "alphabet" $O = \{o_1, o_2, ...o_m\}$, emitted by the Markov chain Y; in most applications (including the one we are concerned with here), the underlying states assumed by Y are not observable (i.e., hidden).

To completely determine a HMM, the following sets of probabilities must be specified:

(1) for each pair s_i, s_j of states of Y, the probability a_{ij} that Y will be in state j at time $t + 1$, given that it is in state i at time t;
(2) for each state of Y s_i and output observation o_k, the probability e_{ik} that $X_t = o_k$, given that $Y_t = s_i$.

(Some definitions of HMMs also require a vector of initial state probabilities, but this can be incorporated into our definition by the specification of a "Start" state, so that the initial probabilities are just the transition probabilites from the Start state.)

The transition probabilities also determine the structure of a directed graph, whose vertices are the Markov chain states and which have an edge from vertex s_i to vertex s_j if the probability of a transition from state s_i to state s_j is strictly greater than 0.

As an example, consider the protein family profile HMM originally proposed by Krogh *et al.* ([KB]). The Markov chain had three hidden states: M, I, and D,

standing for (respectively) Match, Insert, and Delete. The Match state emitted an amino acid in a consensus position; the Insert state could emit any amino acid in a non-consensus position; and the Delete state was a non-emitting state and corresponds to skipping a state in the alignment.

2.1.1. *HMM algorithms.* There are three basic computational problems which arise in the use of HMMs:

(1) given a sequence of observations, compute the probability of that sequence with respect to a known HMM;
(2) given a sequence of observations, find the most probable sequence of states in a known HMM that could have produced the sequence;
(3) given a set of output sequences, compute the parameters of an HMM model that could have produced those sequences.

(The process of determining the values of the transition probabilities a_{ij} and the emission probabilities e_{ik} from empirical data is referred to as *parameterization* of the model, or as *training* the model.)

The widespread use of HMMs in bioinformatics (and in other areas of probabilistic modeling) is due in part to simplicity and efficiency of the algorithms that are available for these computational tasks: the forward-backward algorithm for 1), the Viterbi algorithm for 2), and the Baum-Welch algorithm for 3).

2.2. The use of profile HMMs to determine membership of a sequence in a family. In this section we shall review the application of *profile HMMs* to determine whether a protein is a member of a family. As an example, we take the case of profile HMMs applied to the Pfam database, as described in [SE] (more precisely, this description applies to Pfam-A 2.0). First, a manually verified multiple sequence alignment, known as a *seed alignment*, is computed for a representative set of sequence from the family (for this version of Pfam an average of 22 proteins per family were used for this stage). Second, a *HMM-profile* is built by training a HMM on each representative alignment. Third, to determine whether a new sequence is a member of the profiled family, its probability of occurring by chance (E-value) is computed using the HMM; if the E-value is less than a certain threshold, the protein is classified as a member of that family.

3. Critique of HMMs as probabilistic models of protein sequence families

The use of HMMs as probabilistic models for use in determining membership of a sequence, as described in the previous section, has found widespread acceptance. However, there are certain limitations of HMMs, and characteristics of the protein sequence classification problem, which suggest that HMMs may not the best type of probabilistic model for this classification problem. In this section we examine these issues, which fall into three categories: general modeling considerations, statistical properties of protein sequences, and biologically-based considerations.

3.1. Modeling issues. Birney ([B]) notes several issues in the application of HMMs to sequence analysis, among them the inability of HMMs to incorporate structural information into profile HMMs:

> Despite the almost obvious application of using structural information on a member protein family when one exists to better

the parameterization of the HMM, this has been extremely hard to achieve in practice.

Protein structural information is just one of a number of types of information other than sequence information which could aid in a correct classification if a way could be found to integrate it into the framework of the model type being used; other types of information include amino acid composition, physicochemical properties of the residues, phylogenetic properties of the sequence as a whole, and annotations which identify the biological function of residues (e.g., binding sites).

A somewhat more subjective, but still significant, issue is that of model "fit" to the abstract structure of the classification problem. The classical applications of HMMs - e.g., to speech recognition, or part-of-speech tagging in computational linguistics - are to families of sequences in which the linearity of the sequence is imposed by the temporal ordering of the set as it is being generated. In the case of protein sequences, there is no such temporal ordering - the sequential arrangement of the residues in a protein is the result, ultimately, of evolutionary change operating on the gene that encodes the protein; there is no correspondence between the age of an evolutionary change and its position (i.e., its displacement from the end of the ordered arrangement of residues) - unlike, for instance, the modeling of phonemes in speech recognition, where the temporal ordering is an inherent aspect of human language. Thus the imposition of a strict linear ordering implied by the Markov model is unnatural in the context of this problem.

3.1.1. *Classes of models: generative and discriminative.* Another modeling issue with HMMs relates to a distinction between two categories of probabilistic models. Hidden Markov models are examples of what are known as *generative* models. Such models compute the joint distribution $P(X, Y)$ for all combinations of observations X and underlying states Y; they then classify their input by using Bayes' rule to compute $P(Y|X)$, and pick the Y which maximizes this probability. Training such a model usually consists of finding parameter values which maximize the joint likelihood of the training data.

This is in contrast to *discriminative* models, which compute the conditional probabilities $P(Y|X)$ directly, and then classify their input based on that probability. Training such models consists of finding parameters to maximize the likelihood of the training data with respect to the conditional probabilities, so there is no need compute the likelihood of the input observations of the training data, and there is therefore less work involved in training these models. Discriminative models are usually regarded as superior, since, in a classification problem, there is no need to compute the probability of the observations X; these are input to the classifier when it is applied. (For example, the results reported in [NJ] show a lower asymptotic error for discriminative classifiers).

3.1.2. *Training bias.* As we noted in the previous discussion of profile training, the parameterization of profile HMMs is done by applying the Baum-Welch algorithm to a set of multiply aligned sequences from the family under consideration; no examples of sequences from other families are used. It is very unusual to train a classifier only on positive examples, since otherwise it may not be effective at rejecting cases which do not belong to the class.

As Strope and Moriyama note in their discussion of a classifier not based on alignments, ([SM]),

> [A] disadvantage shared by ... multiple alignment-based methods is that their models are built only from positive samples (protein sequences of interests), and information from negative samples (unrelated protein sequences) is not directly incorporated. Since subsequently found proteins are classified based on these models, possible initial sampling bias is kept and possibly reinforced.

3.2. Statistical properties of protein sequences.

3.2.1. *Observed Distribution of Gaps in Multiple Sequence Alignments.* Consider a simplified Markov model, which has two states, Residue and Gap, and transition probabilities greater than zero for all four possible state transitions. If we make this into a hidden Markov model by allowing the Gap state to emit only the symbol "Gap", and the Residue state to emit any of the 20 amino acids found in proteins, the resulting HMM can generate any sequence that might result from a multiple sequence alignment. All profile HMMs contain a node like the Gap node in this model, which is needed to generate Gap subsequences of arbitrary length in multiple alignments.

Now, the lengths of contiguous subsequences of gaps in the sequences generated by this model will follow an exponential distribution (for a derivation of this property see, for example, [DE], p. 69). However, investigations of the statistics of multiple alignments of protein sequences - such as the multiple alignments that profile HMMs are trained on - have consistently shown that these lengths obey a power law distribution (see, for example, [C], and the references cited therein), not an exponential distribution. Thus profile HMMs model a distribution for Gap subsequences that does not reflect the distributions found empirically. A number of somewhat ad-hoc modifications to the basic HMM model have been proposed to deal with this anomaly, for example so-called duration HMMs; however, we maintain that it is preferable to use a class of model in which this problem does not arise.

3.2.2. *Correlational analysis of protein sequences and the HMM independence assumptions.* Another issue with HMMs is that the Markovian assumptions of conditional independence between non-adjacent residue positions actually do not hold ([HK]). [1]

In this section we will review some of the evidence that these assumptions are in fact unwarranted.

In [WH], Weiss and Herzel considered correlations within sequences over large sets of non-homologous proteins. The autocorrelations (i.e., the correlations the value at a residue position and another one k positions away in the sequence for $1 \leq k \leq 40$) were evaluated for two sets of proteins. The first set contained 1,733 sequences of length close to 125 residues from the Swiss-Prot database which were considered to be dissimilar based on their BLAST scores. The second set consisted of 2192 proteins, each of which was the first member of its superfamily in the PIR sequence database.

Since the computation of the autocorrelation function required that the sequence positions contain numeric values, each residue in a sequence was mapped to a number; this was done in ten different ways for each protein. The values

[1] Although it might be objected that the same criticism could be made of the use of HMMs in computational linguistics.

reflected physical chemistry properties of the amino acids: four were indicator variables for the properties of acidic, basic, neutral polar, and neutral hydrophobic; three were values on a continuous hydrophobicity scale; and two were alpha-helix and beta-strand propensity scores.

After applying finite-sample corrections to the estimation of the autocorrelation coefficients, they found a number of small but statistically significant correlations. In particular, the hydrophobicity correlations tended to oscillate with a period of three or four positions - suggestive of the 3.6-residue periodicity of alpha-helices - and correlations in the alpha-helix propensity score which decay almost monotonically from $k = 1$ to $k = 10$.

In [HK], Hemmerich and Kim examine correlations between positions in protein sequences by use of the *mutual information* of pairs of residue positions in the primary sequence.

If $(s^1, ..., s^n)$ is a sequence of symbols (interpreted as values assumed by n random variables), the *distance d mutual information* is defined as

$$MI(d) = \sum_{j \in \sum_A} \sum_{i \in \sum_A} P(x_i, x_j) log_2(\frac{P_d(x_i, x_j)}{P(x_i)P(x_j)}) \tag{3.1}$$

where $P_d(x_i, x_j)$ is the probability of the residues x_i and x_j occuring in the sequence exactly d positions apart, and $P(x_i)$ and $P(x_j$ are the (marginal) probabilities of those residues occurring in the sequence. (Here we are considering the sequence of a single protein).

Since the mutual information represents the reduction in entropy of a random variable given another random variable, the magnitude of the mutual information will be affected by the entropy of the two random variables; to correct for this, they use a normalization suggested by Martin et al. in [MG], namely dividing the mutual information by the entropy of the joint distribution, given by

$$H(d) = \sum_{j \in \sum_A} \sum_{i \in \sum_A} log_2(\frac{P_d(x_i, x_j)}{P(x_i)P(x_j)}) \tag{3.2}$$

In addition, to estimate P_d and P from a sample the size of a protein sequence, some correction must be made to allow for small sample effects (see [HK] for details of this correction).

After applying this correction to the estimates from actual protein sequences, they found statistically significant values of the normalized mutual information, even between widely separated (i.e., $d \geq 19$) positions, in protein sequences taken from different families in the PFAM database.

Now if non-adjacent positions within the sequence were truly statistically independent (as required by the Markov assumption), the mutual information at distance d for $d > 0$ would be zero. Hence this study fails to confirm the independence hypothesis assumed by the hidden Markov model.

3.3. Biological issues. Last, we consider issues with HMMs that arise from biological considerations.

The first has to do with the (somewhat deeper) issue of what should be considered a protein family - although this concept is often defined in terms of sequence alignments, this may be misleading. As Opiyo and Moriyama ([OM]) note,

> Some homologous proteins are highly diverged and lack enough sequence similarities even though they still share similar structures, biochemical properties, and functions. Obtaining reliable alignments among these protein sequences is difficult.

Since profile HMMs are essentially an alignment-based method, these types of homologous sequences will not be regarded by a profile HMM as belonging to the same family.

A second issue is related to the size of the training set. For already well-defined families, the choice and multiple alignment of a training set of reasonable size is straightforward. In the case of families which have not been as extensively studied, however, this may not be the case. The following example is cited by Opiyo and Moriyama ([OM]):

> ... the mildew resistance locus O (MLO) family is plant specific and currently only 15 member proteins are known. In total, only 22 GPCRs (G-protein coupled receptors) are known in the Arabidopsis thaliana genome, in a stark contrast to 1000 or more GPCRs found in human and mouse. It is possible that plants do not require this protein superfamily as much as animals. However, it is also possible that classifiers used to identify these proteins (mostly profile HMMs) are affected by insufficiently represented training datasets.

Thus, the sensitivity of HMM model parameters to the small number of examples in a training set may lead to in an incorrect biochemical conclusion (i.e, "... plants may not require this superfamily as much as animals.").

4. Conditional random fields for protein family classification

Conditional random fields (CRFs) have only recently started to be applied to bioinformatics. For instance, a recent text on methods for computational gene prediction ([M]) covers HMMs and various generalizations of HMMs at some length, but briefly mentions CRFs only once ([M], p. 383), without giving details of how they could be applied to the gene prediction problem.

4.1. Definition of conditional random fields. Conditional random fields can be thought of as generalizations of Markov random fields, which we now define.

DEFINITION 4.1. If $G = (V, E)$ is an undirected graph with vertices V and edges E, and Y is sets of random variables with $Y = \{Y_v\}_{v \in V}$, then Y is a *Markov random field* with respect to G if

$$P(Y_v|Y_w, w \neq v) = P(Y_v|Y_w, w \sim v) \tag{4.1}$$

That is, each Y_v satisfies a Markov property with respect to G - its dependences on the other random variables within the set Y are limited to those members of Y corresponding to vertices adjacent to v in G.

DEFINITION 4.2. ([LMP]) If Y and X are sets of random variables with Y corresponding as before to the vertices V of the graph G, then (X, Y) is a *conditional random field* if the set of conditional random variables $Y|X$ form a Markov random field with respect to G.

Although conditional random fields have been introduced relatively recently, Markov random fields have been studied for some time (mainly in the context of statistical physics - see, for example, [KS]).

We can also think of CRFs as generalizations of HMMs - the conditioning random variables X are the observations, and the variables Y are the states.

4.1.1. *Probabilities in CRFs.* Although we have said very little about what conditions the joint distribution P in the definition of CRFs must satisfy, the conditions 4.1 in the definition in fact impose fairly stringent constraints on the form that P can take. According to the Hammersley-Clifford Theorem ([CL]), P must be a product of "potential functions" which are constant on the cliques (i.e., complete subgraphs) of the graph G. We will not discuss this further, except to note that this theorem accounts for the form of probabilities defined on the graphical model.

The form of conditional probability is then

$$P(Y|X) = \frac{1}{Z_X} exp(\sum_k \lambda_k f_k(X, Y)) \tag{4.2}$$

where the f_k are "feature functions" defined on the observation sequence X and state sequence Y, the λ_k are weights learned from the training data, and Z_X is a normalization factor needed to make the probabilities sum to 1.0. It is the flexibility of the feature functions, and the fact that they operate on the entire observation sequence, that allow CRFs to utilize information which is not restricted to properties of a single observation or vertex. For instance, it would be easy to accommodate gap penalties that are computed from the length (or perhaps position) of a gap subsequence, rather than incur a fixed penalty for each occurrence of a gap in an alignment.

The form of the graph G can in principle be arbitrary; the form of graph considered by Lafferty *et al.*, and commonly used in modeling sequential data, is known as a *linear chain*: if there are n nodes in the set V, there is a edge between v_i and v_{i+1} for $i = 1, 2, ..., n-1$, and no other edges (see Figure 2). Since the cliques in such a graph consist of i) the individual vertices and ii) the pairs of vertices that are joined by edges, the expression for the conditional probability P becomes

$$P(Y|X) = \frac{1}{Z_X} exp(\sum_{e \in E, k} \lambda_k f_k(X, Y_e, e) + \sum_{v \in V, j} \mu_j g_j(X, Y_v)) \tag{4.3}$$

(where Y_e denotes the vertices associated with the edge e.) In a model whose graph takes this form, the f_k and g_j are sometimes called "transition features" and "state features", respectively.

4.2. Methods of model parameterization for CRFs. The actual implementation of a CRF model requires that numerical values be estimated for the model parameters, which is typically done using a maximum likelihood approach - that is, by maximizing the conditional log likelihood of the training examples over the parameter space. One reason for the widespread use of HMMs in biological sequence analysis is the availability and simplicity of an efficient algorithm - the Baum-Welch algorithm - for this task. So far, at least, no comparable algorithm has emerged for CRFs, although a number of candidates have been put forward. In this section we briefly mention various algorithms that have been proposed.

4.2.1. *Iterative scaling.* In their original paper introducing CRFs ([LM]), Lafferty *et al.* considered two variants of iterative scaling. They report, however, extremely slow convergence, which renders this approach to training impractical for problems of realistic size.

4.2.2. *Gradient-based optimization.* Wallach ([W]) considers several algorithms that utilize the gradient of the objective function to maximize the conditional log-likelihood of the training set, among them conjugate gradient and quasi-Newton methods. However, her results are not conclusive.

4.2.3. *Gradient tree boosting.* Dietterich *et al.* considered an adaptation of Friedman's gradient tree boosting ([F]). This algorithm seems to have reasonable convergence properties, although it is not clear if it is superior to the following algorithm.

4.2.4. *Gradient-based empirical risk minimization.* Gross *et al.* considered minimizing the empirical risk (that is, the loss function on the training set); this is another leading candidate for use in this task.

4.3. Other applications of CRFs in bioinformatics. In this section we list a few applications of CRFs in bioinformatics. This list is not intended to be complete (or even representative), but merely to give some idea of the diversity of problems to which this modeling approach may be applied. We also note some of the model features, to point out some of the various types of information that can be incorporated into the model in a way which would be difficult or impossible with HMMs or their variants.

4.3.1. *Gene finding.* In [DV], DeCaprio *et al.* report on a gene prediction system based on semi-Markov CRFs. They used a linear chain graph structure and trained the model in two ways, the first using a gradient-based function optimizer to maximize the conditional maximum likelihood, and the other a gradient-based optimization of maximum expected accuracy. The additional features include several functions of gap and alignment positions.

4.3.2. *Prediction of protein interactions sites.* In [LL], Li *et al.* describe a CRF model to predict the sites of protein-protein interaction. Their graph model was a linear chain, with one vertex per residue and two possible states, "I" and "N" (for interface site and non-interface site). The features used included accessible surface of a residue, a residue profile score computed from a position-specific scoring matrix, and a transition feature that scored if the residue's label was the same as that of the preceding residue. The model was implemented using the FlexCRF toolkit ([PN]). The model was compared with three other algorithms (neural networks, maximum entropy model, and support vector machine) on various subsets of a set of 1276 chains from the Protein Data Base; the CRF model was found to be superior to or competitive with the other algorithms.

4.3.3. *Protein sequence pairwise alignment.* In [DG], Do *et al.* describe CONTRAlign, a CRF-based modeling system for pairwise alignment of protein sequences, designed to be used in place of pair HMMs. They experimented with several model topologies, including the simple three-state (i.e., Match, Insert-1, and Insert-2) topology frequently used in pair HMM alignment algorithms. Additional features included counts of hydropathic and hydrophilic residues, and secondary structure and solvent accessibility information. They report results competitive with available state-of-the art pair alignment methods using only sequence

information, and superior performance when additional secondary structure and physicochemical properties are incorporated.

4.4. Design of a CRF model for protein sequence family classification. In this section we consider some design choices for a CRF model to be used for classifying proteins into families. It is important to note here that what we are planning is the solution not to a single machine learning problem, but to a class of such problems. That is, we want to specify a general method for implementing, and training a group of classification algorithms, with each algorithm able to recognize the members of a single protein family (or superfamily), as opposed to constructing an algorithm which, given the sequence of a protein, would assign the protein to the correct choice from a large number of families. This approach to classifier construction is sometimes referred to as a "one against all" algorithm and is commonly applied to the solution of multiple-valued classification problems with algorithms, such as support vector machines, which are more suited to binary classification.

4.4.1. *Topology of underlying graph.* The first element to be specified in the design of a CRF is the topology of the underlying dependency graph. Although the use of linear chains is a common choice for this structure, it is not clear whether an effective sequence classifier could be built on such a graph which could only classify sequences of bounded length. The most likely structure for this graph is probably some modification of an existing HMM transition graph structure. It is an open research question whether it might be possible to automate learning the CRF graph structure from the training set, rather than having it specified *a priori*.

4.4.2. *Definition of feature functions.* As noted above, one advantage of the CRF modeling approach is the ease with which it can accomodate information in addition to the sequence. Thus a CRF to recognize protein families could utilize features such as amino acid composition, three-dimensional structure information (if available), annotations of biological function, and phylogenetic information. We also note that the statistical anomaly mentioned in section 3.2.1 could be avoided by a feature function which assigned a probability to contiguous gap subsequence that was not exponential in the length of the subsequence.

4.4.3. *Selection of training set.* The use in practice of a CRF model for recognizing protein families requires a training set for each protein family to be recognized. However, for research into the efficacy of this algorithm, it would be desirable to choose for a first trial a family (or superfamily) on which the performance of HMM methods has been problematic. As we noted above, the class of G-coupled protein receptors was cited in [OM] as an example of a family for which current alignment-based classification methods (such as HMMs) do not perform well; this family would probably be a good choice for the experimental training set for a new CRF model.

4.4.4. *Parameterization algorithm.* As we noted above, no algorithm for training CRFs has so far emerged as the dominant choice for this task, in the way that the Baum-Welch algorithm has emerged to dominate the methods for the analogous problem with HMMs. A preliminary evaluation of available approaches suggests that the gradient tree boosting algorithm of Dietterich *et al.* ([DA]) would be a good initial choice for this application of CRFs; this is, however, only a tentative choice, and may be revised after some computational experience has been gained with the CRF model.

References

[B] Ewan Birney, "Hidden Markov Models in Biological Sequence Analysis", *IBM Journal of Research and Development*,**45** No. 3/4 (2001), 449-454.

[C] Reed A. Cartwright, "Logarithmic gap costs decrease alignment accuracy", *BMC Bioinformatics*,**2006**, 7:257.

[CL] Peter Clifford, "Markov random fields in statistics", in Geoffrey Grimmett and Dominic Welsh, eds., *Disorder in Physical Systems: A Volume in Honour of John M. Hammersley*, pp. 19.32. Oxford University Press, 1990.

[BD] Borodovsky, Mark, Svetlana Ekisheva, *Problems and Solutions in Biological Sequence Analysis*. Cambridge University Press, 2006.

[DA] Dietterich, Thomas, Adam Ashenfelter, and Yaroslav Bulatov, "Training Conditional Random Fields via Gradient Tree Boosting", *Proceedings, International Conference on Machine Learning*, 2004.

[DE] Durbin, Richard, Sean Eddy, Anders Krogh, Graeme Mitchison, *Biological Sequence Analysis*. Cambridge University Press, 1998.

[DG] Do, Chuong B., Samuel S. Gross, and Serafim Batzoglou, "CONTRAlign: Discriminative Training for Protein Sequence Alignment". *Proceedings, Tenth Annual Conference On Research In Computational Molecular Biology*,2006.

[DV] DeCaprio, David, Jade P Vinson, Matthew D. Pearson, Philip Montgomery, Matthew Doherty, and James E. Galagan, "CONRAD: Gene prediction using conditional random fields", *Genome Research*, doi:10.1101/gr.6558107, August 9, 2007.

[F] Friedman, J. H., "Greedy function approximation: A gradient boosting machine", *Annals of Statistics*, 29. 2001

[GR] Gross, S. S., O. Russakovsky, C. B. Do, and S. Batzoglou, "Training conditional random fields for maximum parse accuracy", *Advances in Neural Information Processing Systems19*, 2006.

[HK] Hemmerich, Chris, and Sun Kim, "A Study of Residue Correlation within Protein Sequences and Its Application to Sequence Classification", EURASIP Journal on Bioinformatics and Systems Biology,**2007** article ID 87356, 1-9.

[KB] Krogh, A., M. Brown, K. Sjolander, and D. Hausssler, "Hidden Markov Models in Computational Biology: Applications to Protein Modeling", Journal of Molecular Biology,**235** 1501-1531, 1994.

[KS] Kindermann, Ross, and J. Laurie Snell, *Markov Random Fields and Their Applications*. Providence, RI: American Mathematical Society, 1980.

[LL] Li, Ming-Hue, Lei Lin, Xiao-Long Wang, and Tao Li, "Protein-protein interactio site prediction based on conditional random fields", *Bioinformatics*,**12**,5 (2007) pp. 597-604.

[LM] Lafferty, John, Andrew McCullum, and Fernando Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data", *Proceedings, 18th International Conference on Machine Learning*, 2001.

[M] Majoros, William H., *Methods for Computational Gene Prediction*. Cambridge University Press, 2007.

[MC] McCullum, Andrew, "Efficiently Inducing Features of Conditional Random Fields", *Proceedings, 19th Conference on Uncertainty in Artificial Intelligence*, 2003.

[NJ] Ng, A. N., and M. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes," in *Advances in Neural Information Processing Systems 14*. Cambridge, MA: MIT Press, 2002.

[PN] Phan, X.-H. and L.-M. Nguyen, "FlexCRFs: Flexible conditional random fields toolkit", http://www.jaist.ac.jp/ hieuxuan/softwares.html.

[SM] Strope,Pooja K. and Etsuko N. Moriyama, "Simple alignment-free methods for protein classification: A case study from G-protein-coupled receptors", *Genomics*, **89** (2007) 602612.

[OM] Opiyo, Stephen O., and Etsuko N. Moriyama, "Protein Family Classification with Partial Least Squares", *Journal of Proteome Research* 2007, **6**, 846-853.

[W] Wallach, Hanna, "Efficient training of conditional random fields," Master's thesis, University of Edinburgh, 2002.

[WH] Weiss, Olaf, Hanspeter Herzel, "Information Content of Protein Sequences", *Journal of Theoretical Biology*,**190** 341-353, 1998.

[WJ] Weiss, Olaf, Miguel A. Jiminez-Montano, Hanspeter Herzel, "Correlations in Protein Sequences and Property Codes", *Journal of Theoretical Biology*,**206** 379-386, 2000.

Yahoo!, Inc., Sunnyvale, CA 94089
E-mail address: tj$_e$*merson@yahoo.com*