CONTRAlign: Discriminative Training for Protein Sequence Alignment

Chuong B. Do[1], Samuel S. Gross[1], and Serafim Batzoglou[1]

Stanford University, Stanford CA 94305, USA,
{chuongdo,ssgross,serafim}@cs.stanford.edu,
WWW home page: http://contra.stanford.edu/contralign/

Abstract. In this paper, we present CONTRAlign, an extensible and fully automatic framework for parameter learning and protein pairwise sequence alignment using pair conditional random fields. When learning a substitution matrix and gap penalties from as few as 20 example alignments, CONTRAlign achieves alignment accuracies competitive with available modern tools. As confirmed by rigorous cross-validated testing, CONTRAlign effectively leverages weak biological signals in sequence alignment: using CONTRAlign, we find that hydropathy-based features result in improvements of 5-6% in aligner accuracy for sequences with less than 20% identity, a signal that state-of-the-art hand-tuned aligners are unable to exploit effectively. Furthermore, when known secondary structure and solvent accessibility are available, such external information is naturally incorporated as additional features within the CONTRAlign framework, yielding additional improvements of up to 15-16% in alignment accuracy for low-identity sequences.

1 Introduction

In comparative structural biology studies, analyzing or predicting protein three-dimensional structure often begins with identifying patterns of amino acid substitution via protein sequence alignment. While the evolutionary information obtained from alignments can provide insights into protein structure, constructing accurate alignments may be difficult when proteins share significant structural similarity but little sequence similarity. Indeed, for modern alignment tools, alignment quality drops rapidly when the sequences compared have lower than 25% identity, the "twilight zone" of protein alignment [1].

In recent years, most alignment methods that have claimed improvements in alignment accuracy have done so not by proposing substantially new algorithms for alignment but rather by incorporating additional sources of information. For instance, when structures of some sequences are available, the 3DCoffee program [2] uses pairwise alignments from existing threading-based (FUGUE [3]) and structural (SAP [4] and LSQman [5]) alignment tools to guide sequence alignment construction. When homologous sequences are available and computational expense is of less concern, the $\text{PRALINE}_{\text{PSI}}$ program [6] uses PSI-BLAST–derived [7] sequence profiles to augment the amount of evolutionary

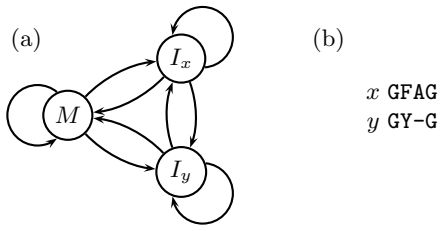

Fig. 1. Traditional sequence alignment model. (a) A simple three-state HMM for sequence alignment. (b) An example sequence alignment, a.

information available to the aligner. The SPEM program [8] takes the additional step of heuristically incorporating PSIPRED [9] predictions of protein secondary structure, a strategy also adopted in the latest version of $\text{PRALINE}_{\text{PSI}}$ [10].

As these programs demonstrate, incorporating additional information can often yield considerable benefits to alignment quality. However, choosing parameters for more complex models can be difficult. In traditional dynamic-programming–based alignment programs, log-odds–based substitution matrices are estimated from large external databases of aligned protein blocks [11], and gap parameters are typically "hand chosen" to maximize performance on benchmark tests [12]. When dealing with more expressive models, however, the high-dimensionality of the parameter space hinders such manual procedures. From the perspective of numerical optimization, the non-convexity of aligner performance as a function of parameters makes hand-tuning difficult for alignment algorithms that rely on complicated ad hoc scoring schemes.

Furthermore, optimizing benchmark performance often leads to *overfitting*, a situation in which the selected parameters are nearly optimal for training benchmark alignments but work poorly on new test data. To combat overfitting, many machine learning studies make use of *cross-validation*, a technique in which an algorithm is trained and tested on independent data sets in order to estimate the ability of the method to generalize to new situations [13].[1]

In this paper, we present CONTRAlign, an extensible and *fully automatic* framework for parameter selection and protein pairwise sequence alignment based on a probabilistic model known as a pair conditional random field (pair-CRF) [15, 16]. In the CONTRAlign methodology, the user first defines an appropriate model topology for pairwise alignment. Unlike for ad hoc algorithms in which model complexity (and hence risk of overfitting) corresponds roughly with the number of free parameters in the model, the effective complexity of a CONTRAlign pair-CRF–based model is controlled by a set of regularization parameters, allowing the user to adjust the trade-off between model expressivity

[1] Properly conducted alignment cross-validation studies are extremely rare in the literature. In the past, a typical defense for benchmark tuning was that aligners with few adjustable parameters are less susceptible to overfitting [14]; such reasoning, however, is less applicable to the complicated procedures of some modern aligners.

and the risk of overfitting. Given a set of gold standard partially labeled alignments, CONTRAlign uses gradient-based optimization and holdout cross validation to automatically determine regularization constants and a set of alignment parameters with good expected performance for future alignment problems.

We show that even under stringent cross-validation conditions, CONTRAlign can learn both substitution and gap parameters that generalize well to previously unseen sequences using as few as 20 training alignments. Augmenting the aligner with sequence-based and external features is seamless in the CONTRAlign framework, yielding large accuracy improvements over modern tools for "twilight zone" sequence sets.

2 Methods

In this section, we first review the standard three-state pair hidden Markov model (pair-HMM) formulation of the sequence alignment problem. We also describe the generalization of the standard pair-HMM to a pair conditional random field (pair-CRF), the use of regularization for trading off between the risk of overfitting and expressivity in a pair-CRF, and a standard optimization procedure for learning pair-CRF parameters from data. We then discuss a variety of model topologies and features possible within the CONTRAlign pair-CRF framework.

2.1 Pair-HMMs for sequence alignment

Consider the state diagram shown in Figure 1 (a). In the standard model, an alignment corresponds to a sequence of independent events describing a path through the state diagram. First, an initial state s is chosen from $\{M, I_x, I_y\}$ with probability π_s. Then, the alignment process alternates between emitting a pair of aligned residues (c, d) upon entry into some state s with probability $\delta_s^{(c,d)}$ (or a single unaligned residue c with probability $\delta_s^{(\mathtt{c},-)}$ or $\delta_s^{(-,\mathtt{c})}$) and transitioning from some state s to another state t with probability $\tau_{s\to t}$ [17].

Since each event is independent, the probability of the alignment decomposes as a product of several terms. For instance, the joint probability of generating an alignment a and sequences x and y shown in Figure 1 (b) is

$$P(a, x, y) = \pi_M \cdot \delta_M^{(\mathtt{G},\mathtt{G})} \cdot \tau_{M\to M} \cdot \delta_M^{(\mathtt{F},\mathtt{Y})} \cdot \tau_{M\to I_x} \cdot \delta_{I_x}^{(\mathtt{A},-)} \cdot \tau_{I_x\to M} \cdot \delta_M^{(\mathtt{G},\mathtt{G})}. \quad (1)$$

Alternatively, we may rewrite (1) as $P(a, x, y; \mathbf{w}) = \exp(\mathbf{w}^T \mathbf{f}(a, x, y))$ where $\mathbf{w}$ is a parameter vector and $\mathbf{f}(a, x, y)$ is a vector of "feature counts" indicating the number of times each parameter appears in the product on the right-hand side. More explicitly, if $\mathbf{w} = \left[\log \pi_M, \log \delta_M^{(\mathtt{G},\mathtt{G})}, \log \tau_{M\to M}, \cdots\right]^T$, then the corresponding feature count vector is given by

$$\mathbf{f}(a, x, y) = \begin{bmatrix} \#\text{ of times alignment starts in state } M \\ \#\text{ of times alignment generates } (\mathtt{G},\mathtt{G}) \text{ in state } M \\ \#\text{ of times alignment follows } M \to M \text{ transition} \\ \vdots \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \\ \vdots \end{bmatrix}. \quad (2)$$

Given two sequences x and y, the Viterbi algorithm computes an alignment a that maximizes $P(a \mid x, y; \mathbf{w})$ in $O(|x| \cdot |y|)$ time. For the model shown in Figure 1, the Viterbi algorithm is equivalent to the Needleman-Wunsch algorithm [18]. In this paper, we use an alternative parsing algorithm for finding alignments with the maximum expected number of correct matches; for details, see [17, 19, 20].

Given a collection of aligned training examples $\mathcal{D} = \left\{(a^{(i)}, x^{(i)}, y^{(i)})\right\}_{i=1}^{m}$, the standard parameter estimation procedure (known as *generative* training in the machine learning literature [21]) is to maximize the joint log-likelihood $\ell(\mathbf{w} : \mathcal{D}) := \sum_{i=1}^{m} \log P(a^{(i)}, x^{(i)}, y^{(i)}; \mathbf{w})$ of the data and alignments, subject to constraints ensuring that the original parameters (π_M, $\delta_M^{(\mathtt{G},\mathtt{G})}$, etc.) are nonnegative and normalize. When training with fully-specified alignments, the optimization problem not only is convex but also has a closed-form solution.

In some benchmark alignment databases, such as BAliBASE [22] and PREFAB [23], reference alignments are partially ambiguous: certain columns are marked as reliable (known as core blocks) while the alignment of other positions may be left unspecified. In these cases, the training set $\hat{\mathcal{D}} = \left\{(\hat{a}^{(i)}, x^{(i)}, y^{(i)})\right\}_{i=1}^{m}$ thus consists of partial alignments $\hat{a}^{(i)}$. Letting $\mathcal{A}^{(i)}$ denote the set of alignments consistent with the known reliable columns of $\hat{a}^{(i)}$, the joint log-likelihood becomes $\ell(\mathbf{w} : \hat{\mathcal{D}}) := \sum_{i=1}^{m} \log \sum_{a \in \mathcal{A}^{(i)}} P(a, x^{(i)}, y^{(i)}; \mathbf{w})$. Despite the nonconvexity of the new optimization problem, most numerical optimization approaches, such as EM or gradient ascent, work well in practice [17].[2]

2.2 From pair-HMMs to pair-CRFs

In the pair-HMM formalism, the constraints on the parameters $\mathbf{w}$ to represent initial, transition, or emission log probabilities allowed us to interpret a pair-HMM as defining $P(a, x, y; \mathbf{w})$, the probability of stochastically generating an alignment. Unlike pair-HMMs, pair-CRFs do not define this joint probability but instead directly model the conditional probability,

$$P(a \mid x, y; \mathbf{w}) = \frac{P(a, x, y; \mathbf{w})}{\sum_{a' \in \mathcal{A}} P(a', x, y; \mathbf{w})} = \frac{\exp(\mathbf{w}^T \mathbf{f}(a, x, y))}{\sum_{a' \in \mathcal{A}} \exp(\mathbf{w}^T \mathbf{f}(a', x, y))}, \qquad (3)$$

where $\mathcal{A}$ denotes the set of all possible alignments of x and y. As before, the parameter vector $\mathbf{w}$ completely parameterizes the pair-CRF, but this time, we impose no constraints on the entries of $\mathbf{w}$. Here, a parameter entry w_i does not corresponds to the log probability of an event (as in a pair-HMM) but rather is a real-valued feature weight that either raises or lowers the "probability mass" of a relative to other alignments in $\mathcal{A}$. Similar models have been proposed for string edit distance in natural language processing applications [24, 25].

Clearly, pair-CRFs are at least as expressive as their pair-HMM counterparts, as any suitable parameter vector $\mathbf{w}$ for an alignment pair-HMM is a valid parameter vector for its corresponding alignment pair-CRF. Furthermore, while

[2] In practice, the only step needed to ensure good convergence was to break symmetries in the model by initializing parameters to small random values.

pair-CRFs assume a particular factorization of the conditional probability distribution $P(a \mid x, y; \mathbf{w})$, they make far weaker independence assumptions regarding feature counts $\mathbf{f}(a, x, y)$. Thus, these models are amenable to using complex feature sets that may be difficult to incorporate within a generative pair-HMM.

Training a pair-CRF involves maximizing the conditional log-likelihood of the data (known as *discriminative* or *conditional* training [21]). Unlike generative training, discriminative training directly optimizes predictive ability while ignoring $P(x, y)$, the model used to generate the input sequences. When a pair-CRF places undue importance on unreliable features (i.e. the magnitude of some parameter w_j is large), overfitting may occur. To prevent this, we place a Gaussian prior, $P(\mathbf{w}) \propto \exp(-\sum_j C_j w_j^2)$, on the parameters $\mathbf{w}$. Thus, we maximize $\ell(\mathbf{w} : \mathcal{D}) := \sum_{i=1}^{m} \log P(a^{(i)} \mid x^{(i)}, y^{(i)}; \mathbf{w}) + \log P(\mathbf{w})$, or equivalently,

$$\sum_{i=1}^{m} \left(\mathbf{w}^T \mathbf{f}(a^{(i)}, x^{(i)}, y^{(i)}) - \log \sum_{a' \in \mathcal{A}} \exp(\mathbf{w}^T \mathbf{f}(a', x^{(i)}, y^{(i)})) \right) - \sum_j C_j w_j^2. \quad (4)$$

The final term in (4) encourages parameters to be "small" unless increased size yields a sufficient increase in likelihood. This technique, known as *regularization*, leads to improved generalization both in theory and in practice [26].

Parameter learning for pair-CRFs using a fixed set of regularization parameters $\mathbf{C} = \{C_j\}$ is straightforward. The objective function in (4) is convex for fully-specified alignments and hence a global maximum of the regularized likelihood can be found using any efficient gradient-based optimization algorithm (such as conjugate gradient, or L-BFGS [27]). The gradient $\nabla_{\mathbf{w}} \ell(\mathbf{w} : \mathcal{D})$ is

$$\sum_{i=1}^{m} \left(\mathbf{f}(a^{(i)}, x^{(i)}, y^{(i)}) - \mathbf{E}_{a \sim P(\mathcal{A} \mid x^{(i)}, y^{(i)})} \mathbf{f}(a, x^{(i)}, y^{(i)}) \right) - 2\mathbf{C} \circ \mathbf{w}, \quad (5)$$

where $\mathbf{C} \circ \mathbf{w}$ denotes the component-wise product of the vectors $\mathbf{C}$ and $\mathbf{w}$. Disregarding regularization, we see that the partial derivative of the log-likelihood with respect to each parameter w_j is zero precisely when the observed and expected counts for the corresponding feature f_j (taken with respect to the distribution over unobserved alignments) match. For fully-specified alignments $a^{(i)}$, the former term in the parentheses can be directly tabulated from the alignment $a^{(i)}$, and the latter term can be computed using the forward-backward algorithm. The partially-specified alignment case follows similarly [17].

2.3 Pairwise alignments with CONTRAlign

In the previous subsections, we described the standard pair-HMM model for sequence alignment and its natural extension to pair-CRFs. In this subsection, we present CONTRAlign, a feature-rich alignment framework that leverages the power of pair-CRFs to support large non-independent feature sets while controlling model complexity via regularization.

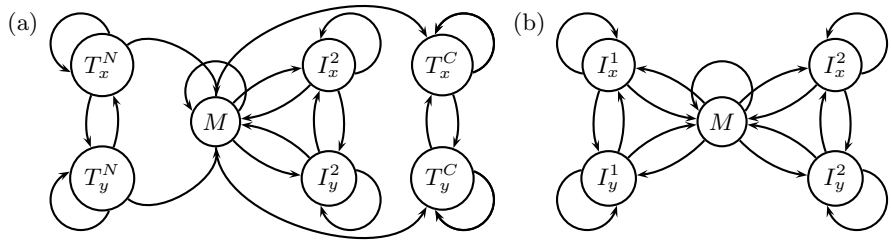

Fig. 2. Model variants. (a) CONTRAlign$_{\text{LOCAL}}$ topology with N/C-terminal flanking inserters, (b) CONTRAlign$_{\text{DOUBLE-AFFINE}}$ topology with two insert state pairs.

Choice of model topology. As a baseline, we used the standard three-state pair-HMM model (CONTRAlign$_{\text{BASIC}}$) shown in Figure 1 (a). We experimented with a variety of other model topologies as well, including:

- CONTRAlign$_{\text{LOCAL}}$: a model with flanking N-terminal and C-terminal insert states to allow for local homology detection (see Figure 2 (a)), and
- CONTRAlign$_{\text{DOUBLE-AFFINE}}$, a model with an extra pair of gap states in order to model both long and short insertions (see Figure 2 (b)).

Hydropathy-based gap context features. The CLUSTALW protein multiple alignment program incorporates a large number of heuristics designed to improve performance on the BAliBASE benchmark reference [28]. One heuristic applicable to pairwise alignment is the reduction of gap penalties in runs of 5 or more hydrophilic residues. Typically, the core regions of globular proteins, where insertions and deletions are less likely, consist of hydrophobic residues. Reducing gap penalties in hydrophilic regions encourages the aligner to place gaps in regions less likely to be part of the hydrophobic core; similar heuristics are incorporated in the MUSCLE [23] alignment program as well.

In CONTRAlign, we tested a variant of this idea (CONTRAlign$_{\text{HYDROPATHY}}$) by incorporating hydropathy-based context features for insertion scoring. Specifically, for each insertion open, insertion continue, or insertion close event in sequence x, we defined the number of hydrophilic residues in a window of length 6 in sequence y to be the *hydrophilic count* context of that event (and vice versa for insertions in sequence y). We added a total of fourteen features to the model, seven indicating whether an insertion open or close occurred with a hydrophilicity context of 0, 1, ..., or 6, and similarly for insertion continues.

Incorporating external information. To test the ability of CONTRAlign to incorporate external information, we also experimented with giving CONTRAlign information about secondary structure (CONTRAlign$_{\text{DSSP}}$) and solvent accessibility (CONTRAlign$_{\text{ACCESSIBILITY}}$) of the sequences being aligned, as extracted from the PDBFinderII database [29]. In particular, DSSP annotations of sequences from PDBFinderII were converted to a three-letter code

using the grouping employed in the EVA automatic structure prediction benchmark server, {{G, H, I}, {E, B}, {T, S, C}} [30]. Similarly, annotations of positional amino acid solvent accessibilities were converted from the PDBFinderII 0-9 scale using the grouping {{0}, {1, 2, 3}, {4, 5, 6}, {7, 8, 9}}. To assess the value of using predicted external tracks of information, we also tested variants using PSIPRED single (CONTRAlign$_{\text{PSIPRED-SINGLE}}$) and multiple (CONTRAlign$_{\text{PSIPRED-MULTI}}$) sequence secondary structure predictions.

For each annotation track, we added emission features to the match and insertion states of the basic model that would allow them to simultaneously emit both sequence and annotation. A similar method based on "two-track HMMs" was previously used to improve the quality of fold recognition via predicted local structure [31]. In that work, the authors constructed an HMM that simultaneously emitted two observation signals and relied on the assumed independence of the two character emission tracks during parameter learning. To compensate for the violated independence assumption, the authors added heuristic weights to each emission; thus, the "probability" of a two-track emission was given by $P(o_1|s)^{w_1} P(o_2|s)^{w_2}$, where the weights w_1 and w_2 were selected manually. In contrast, such correction factors are not needed in the pair-CRF model presented here, as pair-CRF learning makes no assumptions regarding the independence of the emission features of each state. Thus, pair-CRFs provide a consistent framework for incorporating multiple sources of evidence without the need for artificial compensation as present in multi-track generalizations of HMMs.

3 Results

In the protein sequence alignment literature, benchmark databases of reference alignments have emerged as the standard metric for evaluating aligner performance. First, the aligner-to-be-tested performs alignments for all sequence sets in the database. Then, accuracy is measured with respect to known reliable columns of a hand-curated reference alignment.

While benchmark tests have been an invaluable asset to the development of alignment algorithms, statistics in the literature often misrepresent the significance of accuracy differences between aligners. Some reference databases, such as BAliBASE and PREFAB, contain multiple copies of a single sequence in several different alignments. Ignoring the non-independence of these test cases artificially lowers p-values when using rank tests to compare the performance of two aligners. Even more dangerous is the common practice of "tuning" parameters to improve performance on individual benchmark datasets. Due to the absence of (or improper use of) cross-validation in most studies in the literature, good benchmark results may not indicate good alignment accuracy for novel proteins.

With this in mind, we designed a series of carefully controlled cross-validation experiments to assess the contribution of the different model topologies/features toward CONTRAlign alignment accuracy, and the ability of the learned alignment model to generalize across different benchmark reference databases.

3.1 Cross-validation methodology

We extracted alignments from four standard benchmarking databases:

1. BAliBASE 3.0 [32], a collection of 218 manually refined reference multiple alignments based on 3D structural superpositions;
2. SABmark 1.65 [33], a collection of 236 very low to low identity ("Twilight Zone") and 462 low to intermediate identity ("Superfamilies") sets of all-pairs pairwise consensus structural alignments derived from the SCOP [34] classification;
3. PREFAB 4.0 (beta) [23], a collection of 1932 pairwise structural alignments supplemented by PSI-BLAST homologs from the NCBI nonredundant protein sequence database [35]; and
4. HOMSTRAD (September 1, 2005 release), a curated database of 1032 structure-based multiple alignments for homologous families [36].

We projected the BAliBASE and HOMSTRAD reference multiple alignments into all-pairs pairwise structural alignments. Then, for each multiple sequence set from BAliBASE, HOMSTRAD, and SABmark, we computed percent identity for all pairwise alignments and retained the alignment with median identity.

To construct independent training and testing sets for cross-validation, we relied on the CATH protein structure classification hierarchy [37]; a similar protocol was followed in benchmarking the PSIPRED protein secondary structure prediction program. Specifically, we considered a pair of alignments A and B independent if no two proteins $x \in A$ and $y \in B$ share the same CATH classification at the "homology" level. Using this criterion, we used a greedy procedure to select alignments for training and testing; at each step in the alignment selection process, we selected an alignment, which was independent of all alignments previously selected, from the database with the fewest representatives. The resulting selected pairwise alignments consisted of 38 alignments from BAliBASE, 123 from SABmark, 139 from PREFAB, and 187 from HOMSTRAD.

For parameter learning in CONTRAlign, we considered all matched positions (in core blocks where applicable) to be labeled and treated gapped or unannotated regions as missing data. To select regularization constants in a manner strictly independent of the testing set, we used a staged holdout cross validation procedure on the training data only. Specifically, for a given training collection $\mathcal{D}$, we randomly chose 20% of the alignments for a holdout set and performed training only on the remaining 80%. We manually divided model features into a small number of regularization groups (usually two or three) and constrained the regularization constants for features in each group to be the same. Starting from a model with only transition features, we introduced new features, one group at a time. In each iteration, we used a golden section search and standard L-BFGS optimization to optimize holdout set conditional log-likelihood over possible settings of the regularization parameter for the newly introduced group. Once all features were introduced, we retrained the model on all of the training data using the chosen regularization constants.

We measured alignment accuracy using the Q score [23], the proportion of true alignment character matches correctly predicted. For pairwise alignments, the Q score is equivalent to both the sum-of-pairs (SP) and total column (TC) score commonly used for measuring multiple alignment accuracy [22].

3.2 Comparison of model topologies and feature sets

In our first set of cross-validation experiments, we selected each of the reference databases in turn as the testing set, and used alignments pooled from the other three databases as the training set.[3] Table 1 compares the various models described in Section 2.3 as evaluated on each of the four databases. As shown in the table, changes in model topology (also possible in pair-HMM aligners) give small improvements in overall accuracy. As expected, the major improvements come with the incorporation of features based on external information, such as DSSP secondary structure or solvent accessibility annotations.

Interestingly, accounting for some sequence features present in the input sequence alone (in particular, hydropathy) gives a larger increase in performance than any change in model topology. We return to this observation in Section 3.3. Also, in contrast to the massive performance gains when using real DSSP secondary structure annotations, our numbers suggest that predicted PSIPRED single sequence secondary structures are not informative for alignment. PSIPRED multiple sequence predictions, however, are substantially more accurate and give strong improvements in aligner performance.

Based on these observations, we constructed the CONTRAlign$_{\text{COMBINED}}$ model, which incorporated the four most informative components: double-affine insertion scoring, hydropathy, DSSP secondary structure, and solvent accessibility. To do this, we built an alignment model incorporating the latter two types of features as separate "tracks" of information. A variety of other encodings are possible that allow for more explicit dependencies between secondary structure and solvent accessibility, but we did not explore this further. For the model described, resulting alignments are on average 10% more accurate than those using the basic model alone.

3.3 Comparison to modern sequence alignment tools

Next, we compared the CONTRAlign$_{\text{HYDROPATHY}}$ model to a variety of modern sequence alignment methods, including MAFFT 5.732 (both L-INS-i and G-INS-i) [38, 39], CLUSTALW 1.83 [28], MUSCLE 3.6 [23], T-Coffee 2.66 [40],

[3] For most reference databases, with the notable exception of SABmark 1.65, alignment accuracies are roughly consistent. This difference is likely explained by the substantially higher proportion of low-identity alignments in SABmark, though we did not conduct a careful investigation of this phenomenon.

CONTRAlign variant	BAliBASE (38)	SABmark (123)	PREFAB (139)	HOMSTRAD (187)	Overall (487)	p-value
BASIC	78.93	42.04	74.40	82.61	69.73	n/a
LOCAL	79.10	42.06	74.46	83.34	70.05	7.8×10^{-2}
DOUBLE-AFFINE	78.85	44.50	75.40	84.02	71.17	0.00040
HYDROPATHY	82.07	45.61	76.75	84.78	72.38	1.5×10^{-9}
ACCESSIBILITY	80.80	52.09	79.47	86.84	75.49	3.1×10^{-27}
PSIPRED-SINGLE	77.97	44.94	74.97	82.40	70.47	2.9×10^{-1}
PSIPRED-MULTI	83.13	51.91	79.25	85.35	74.99	2.3×10^{-21}
DSSP	83.01	57.50	81.89	86.88	77.73	1.2×10^{-33}
COMBINED	88.46	61.85	83.66	88.68	80.45	1.2×10^{-44}

Table 1. Comparison of CONTRAlign variants. We counted the number of times each variant outperformed or was outperformed by the basic model, and assigned p-values using a simple yet robust statistical sign test to check for deviations from a symmetric distribution in which either aligner is equally likely to do better. Accuracy improvements relative to the basic model are significant in every case with the exceptions of the local and PSIPRED single sequence prediction models.

Method	BAliBASE (38)	SABmark (123)	PREFAB (139)	HOMSTRAD (187)	Overall (487)	p-value
MAFFT (G-INS-i)	74.56	41.25	71.37	80.53	67.53	9.8×10^{-22}
MAFFT (L-INS-i)	78.08	39.58	71.95	82.01	68.12	7.1×10^{-17}
T-Coffee	74.73	42.84	72.99	82.40	69.12	1.2×10^{-11}
CLUSTALW	79.43	41.36	73.29	81.62	68.90	1.5×10^{-5}
CLUSTALW (-nohgap)	79.65	40.92	73.51	81.35	68.77	6.2×10^{-7}
MUSCLE	77.42	41.72	72.67	82.63	69.05	2.1×10^{-13}
MUSCLE (-hydrofactor 0.0)	74.78	37.78	69.19	77.83	65.01	7.1×10^{-32}
CONTRAlign (Bali, no reg)	92.57	39.33	68.77	80.45	67.68	5.7×10^{-14}
CONTRAlign (Bali, reg)	84.75	39.08	73.45	82.21	69.01	1.2×10^{-7}
CONTRAlign (All, reg)	82.42	47.39	76.74	85.22	73.03	0.00021
ProbCons (Bali)	78.62	42.53	73.75	83.64	70.04	4.8×10^{-8}
ProbCons (cv)	78.48	43.31	71.78	81.36	68.79	9.7×10^{-11}
$\text{CONTRAlign}_{\text{HYDROPATHY}}$	82.07	45.61	76.75	84.78	72.38	n/a

Table 2. Comparison of modern alignment methods. p-values indicate significance of performance difference between each method and $\text{CONTRAlign}_{\text{HYDROPATHY}}$ based on a sign test, as in Table 1.

and ProbCons 1.10 [20].[4] In these experiments, we used the existing multiple alignment tools to compute pairwise alignments from the cross-validation setup.

Obtaining a proper cross-validated estimate of an aligner's performance requires tuning the program to multiple training collections, unbiased by testing set performance. For most modern alignment programs, avoiding testing set bias is difficult since parameters are typically tuned by hand. Methods with automatic training procedures, like ProbCons, permit cross-validation to some extent, with the caveat that the program by default uses BLOSUM62-based amino acid frequencies estimated from data overlapping all testing sets.

In Table 2, the overall accuracies of most modern hand-tuned methods fall within a one percent range (68-69%). The ProbCons (Bali) method, which uses an automatic unsupervised learning algorithm to infer parameters from all 141 BAliBASE 2 alignments, outperforms most other methods on the BAliBASE

[4] The Align-m program, which was developed by the creator of the SABmark reference set, could not be tested on pairwise alignments since the current version (2.3) requires at least three input sequences for an alignment.

dataset except CLUSTALW, which is based on a much more complex model with many internal parameters adjusted to maximize performance on BAliBASE [41]. As previously suggested [41, 42], CLUSTALW's lower relative performance on other databases suggest that it may indeed be overfit to its training set.

To demonstrate the dangers of such overfitting, we trained CONTRAlign on the small set of 38 BAliBASE sequences, with and without regularization. In this situation, omitting regularization leads to tremendous overfitting to BAliBASE, with regularization giving a significant improvement in accuracy. Regularization, however, is not a substitute for proper cross-validation; when overfitting to all four databases, CONTRAlign yields clearly over-optimistic numbers compared to the properly cross-validated test. Similarly, cross-validated ProbCons (despite using BLOSUM62 amino acid frequencies and thus having an easier learning task than CONTRAlign) performs worse than the non-cross-validated model as expected, confirming that absence of cross-validation can give significantly unrealistic estimates of aligner performance.

As shown, cross-validated CONTRAlign (i.e., $\text{CONTRAlign}_{\text{HYDROPATHY}}$) beats current state-of-the-art methods by 3-4% despite (1) estimating all model parameters, including the emission matrix, and (2) following a rigorous cross-validated training procedure. Based on the comparison of the hydropathy and basic models in Table 1, it is clear that these accuracy gains result directly from the use of hydropathy-based gap scoring. Perhaps most striking, however, is that a variety of existing methods, including CLUSTALW and MUSCLE, already incorporate hydropathy-based modifications in their alignment scoring, yet do not manage to achieve above 70% accuracy on our benchmarks. Disabling these modifications in the respective programs gives no substantial change in performance for CLUSTALW and greatly reduces MUSCLE accuracy.[5] Our result confirms that hydropathy is indeed an important signal for protein sequence alignment and that properly accounting for this can yield significantly higher alignment accuracy than the current state-of-the-art.

3.4 Regularization and generalization performance

To understand the effects of regularization at low training set sizes, we reserved a set of 200 randomly chosen pairwise alignments pooled from all four reference databases to use as a testing set. We then experimented with learning parameters for the $\text{CONTRAlign}_{\text{HYDROPATHY}}$ topology using varying training set sizes. For staged regularization, we considered a variant of the basic model in which we introduced amino emission features corresponding to the six-character reduced amino alphabet, {{A, G, P, S, T}, {C}, {D, E, N, Q}, {F, W, Y}, {H, K, R}, {I, L, M, V}}, in addition to the regular twenty-letter amino acid emissions [43]. In the first regularization stage, the program learns a coarse-grained substitution matrix, followed by finer-grained refinements in the second stage.

[5] Performing a sign test to compare performance when hydropathy scoring is either enabled or disabled yields p-values of 0.56 and 6.28×10^{-31} for CLUSTALW and MUSCLE, respectively.

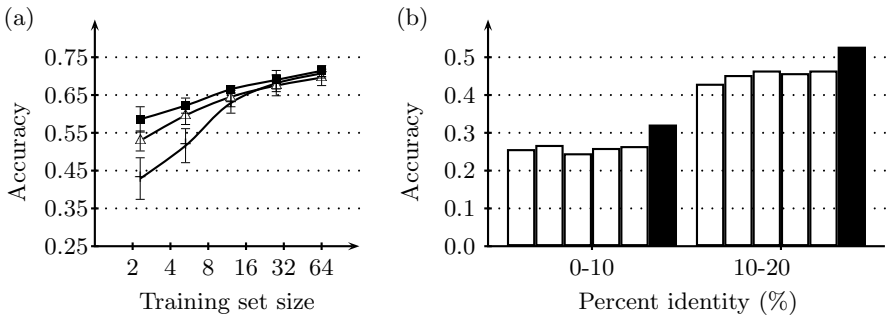

Fig. 3. Alignment accuracy curves. (a) Accuracy as a function of training set size. The three curves give performance when using no (+), simple (△), and staged (•) regularization. All data points are averages over 10 random training/test splits. (b) Accuracy in the "twilight zone." For each conservation range, the uncolored bars (□) give accuracies for MAFFT (L-INS-i), T-Coffee, CLUSTALW, MUSCLE, and ProbCons (Bali) in that order, and the colored bar (■) indicates the accuracy for CONTRAlign.

The results in Figure 3 (a) demonstrate that with intelligent use of regularization, good accuracy can be achieved with only 20 example alignments, far fewer than the number of blocks used to estimate traditional alignment substitution matrices such as BLOSUM [11]; nevertheless, the simpler regularization scheme was still quite effective compared to having no regularization at all.

For specific classes of alignments, such as sequences with long insertions or compositional biases, a robust training procedure allows one to tailor the alignment algorithm to the data; when, in addition, training data is sparse, regularization deters overfitting and enables further customization of alignment parameters. Furthermore, as the amount of available training data grows, accuracy will continue to increase as well.

3.5 Alignment accuracy in the "twilight zone"

To understand the situations in which CONTRAlign$_{\text{HYDROPATHY}}$ was most effective, we stratified the 487 sequences of our dataset into several percent identity ranges and measured the accuracy of all methods for each range. For alignments with at least 20% identity, all methods obtained similar accuracies, ranging from 87.2% to 88.7%. In the 0-10% and 10-20% identity ranges, however, CONTRAlign accuracy was substantially higher than that of other methods; here, CONTRAlign achieved cross-validated accuracies of 32.2% and 52.8% compared to non-cross-validated accuracy ranges of 25.7-26.8% and 43.0-46.5% for all other methods (see Figure 3 (b)). Incorporating external sequence features such as in the combined model of Section 3.2 yields accuracies of 48.0% and 68.5% (not shown in figure), indicating that external sequence information can significantly increase the reliability of alignments when available.

4 Discussion

Construction of a modern high-performance sequence alignment program involves understanding the variety of biological features available when performing alignment, building a model of interactions demonstrating how those features may be combined in an aligner, and careful cross-validation experiments to ensure good generalization performance of the aligner on future data. In this paper, we presented CONTRAlign, a pair conditional random field for learning alignment parameters effectively even when small amounts of training data are available. Using regularization and holdout cross-validation, our algorithm automatically learns parameters with good generalization performance. Public domain source code for CONTRAlign, datasets used in experiments from this paper, and a web server for submitting sequences are available online at `http://contra.stanford.edu/contralign`.

Since CONTRAlign specifies a conditional probability distribution over pairwise alignments, the ProbCons methodology provides one straightforward extension of CONTRAlign to multiple alignment. The main limitation of the CONTRAlign framework, however, is training time: L-BFGS gradient-based optimization is expensive, especially in the context of the holdout cross validation procedure used. Typical training runs for the experiments in this paper (including holdout cross-validation to find regularization constants) took approximately an hour on a 40-node Pentium IV cluster. Perceptron learning [44], a recent technique for discriminatively training structured probabilistic models, may provide a scalable alternative to gradient-based optimization.

The primary advantage of CONTRAlign is its ability to free aligner developers to focus on the *biology* of sequence alignment—modelling and feature selection—while transparently taking care of details such as parameter learning and generalization performance. The models described in this paper were only the first steps toward a better understanding of the sequence alignment problem. Combining new CONTRAlign topologies and features with known successful variants should result in even higher performance. A systematic exploration of such possibilities remains to be done.

5 Acknowledgments

We thank G. Asimenos for his tireless technical assistance in dealing with cluster issues during the development of the program. We also thank R. C. Edgar and A. Sidow for useful discussions regarding aligner features and testing methodology. CBD and SSG were supported by NDSEG fellowships. This work was supported in part by NSF grant 0312459. SB acknowledges support from the NSF CAREER Award and the Alfred P. Sloan Fellowship.

References

1. Rost, B.: Twilight zone of protein sequence alignments. Protein Eng **12**(2) (1999) 85–94

2. O'Sullivan, O., Suhre, K., Abergel, C., Higgins, D.G., Notredame, C.: 3DCoffee: combining protein sequences and structures within multiple sequence alignments. J Mol Biol **340** (2004) 385–395
3. Shi, J., Blundell, T.L., Mizuguchi, K.: FUGUE: sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties. J Mol Biol **310** (2001) 243–257
4. Taylor, W.R., Orengo, C.A.: Protein structure alignment. J Mol Biol **208** (1989) 1–22
5. Kabsch, W.: A discussion of the solution for the best rotation to relate two sets of vectors. Acta Crystallog Sect A **34** (1978) 827–828
6. Simossis, V.A., Kleinjung, J., Heringa, J.: Homology-extended sequence alignment. Nucleic Acids Res **33**(3) (2005) 816–824
7. Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J.: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res **25**(17) (1997) 3389–3402
8. Zhou, H., Zhou, Y.: SPEM: improving multiple sequence alignment with sequence profiles and predicted secondary structures. Bioinformatics **21**(18) (2005) 3615–3621
9. Jones, D.T.: Protein secondary structure prediction based on position-specific scoring matrices. J Mol Biol **292**(2) (1999) 195–202
10. Simossis, V.A., Heringa, J.: PRALINE: A multiple alignment toolbox that integrates homology-extended and secondary structure information. Nucleic Acids Res **33 (Web Server issue)** (2005) W289–W294
11. Henikoff, S., Henikoff, J.G.: Amino acid substitution matrices from protein blocks. Proc Nat Acad Sci USA **89** (1992) 10915–10919
12. Vingron, M., Waterman, M.S.: Sequence alignment and penalty choice. Review of concepts, case studies and implications. J Mol Biol **235**(1) (1994) 1–12
13. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: IJCAI. (1995) 1137–1145
14. Raghava, G.P.S., Searle, S.M.J., Audley, P.C., Barber, J.D., Barton, G.J.: OXBench: A benchmark for evaluation of protein multiple sequence alignment accuracy. BMC Bioinformatics **4**(47) (2003)
15. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proc. 18th ICML. (2001) 282–289
16. Sha, F., Pereira, F.: Shallow parsing with conditional random fields (2003)
17. Durbin, R., Eddy, S.R., Krogh, A., Mitchison, G.: Biological Sequence Analysis : Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press (1999)
18. Altschul, S.F.: Amino acid substitution matrices from an information theoretic perspective. J Mol Biol **219** (1991) 555–565
19. Holmes, I., Durbin, R.: Dynamic programming alignment accuracy. J Comp Biol **5**(3) (1998) 493–504
20. Do, C.B., Mahabhashyam, M.S., Brudno, M., Batzoglou, S.: PROBCONS: probabilistic consistency-based multiple sequence alignment. Genome Res **15**(2) (2005) 330–340
21. Ng, A., Jordan, M.: On discriminative vs. generative classifiers: a comparison of logistic regression and naive Bayes. In: NIPS 14. (2002)
22. Thompson, J.D., Plewniak, F., Poch, O.: A comprehensive comparison of multiple sequence alignment programs. Nucleic Acids Res **27**(13) (1999) 2682–2690

23. Edgar, R.C.: MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res **32**(5) (2004) 1792–1797
24. McCallum, A., Bellare, K., Pereira, F.: A conditional random field for discriminatively-trained finite-state string edit distance. In: Proc. UAI. (2005)
25. Bilenko, M., Mooney, R.J.: Alignments and string similarity in information integration: A random field approach. In: Proc. Dagstuhl Seminar on Machine Learning for the Semantic Web. (2005)
26. Vapnik, V.N.: Statistical Learning Theory. Wiley-Interscience (1998)
27. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer (1999)
28. Thompson, J.D., Higgins, D.G., Gibson, T.J.: CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties, and weight matrix choice. Nucleic Acids Res **22**(22) (1994) 4673–4680
29. Krieger, E., Hooft, R.W.W., Nabuurs, S., Vriend, G.: PDBFinderII—a database for protein structure analysis and prediction. Submitted (2004)
30. Eyrich, V.A., Mart'i-Renom, M.A., Przybylski, D., Madhusudhan, M.S., Fiser, A., Pazos, F., Valencia, A., Sali, A., Rost, B.: EVA: continuous automatic evaluation of protein structure prediction servers. Bioinformatics **17**(12) (2001) 1242–1243
31. Karchin, R., Cline, M., Mandel-Guttfreund, Y., Karplus, K.: Hidden markov models that use predicted local structure for fold recognition: alphabets of backbone geometry. Proteins: Structure, Function, and Genetics **51**(4) (2003) 504–514
32. Thompson, J.D., Koehl, P., Ripp, R., Poch, O.: BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark. Proteins **61** (2005) 127–136
33. Walle, I.V., Lasters, I., Wyns, L.: SABmark—a benchmark for sequence alignment that covers the entire known fold space. Bioinformatics **21**(7) (2005) 1267–1268
34. Murzin, A.G., Brenner, S.E., T., H., C., C.: SCOP: a structural classification of proteins database for the investigation of sequences and structures. J Mol Biol **247** (1995) 536–540
35. Pruitt, K.D., Tatusova, T., Maglott, D.R.: NCBI Reference Sequence project: update and current status. Nucleic Acids Res **31**(1) (2003) 34–37
36. Mizuguchi, K., Deane, C.M., Blundell, T.L., Overington, J.P.: HOMSTRAD: a database of protein structure alignments for homologous familes. Protein Sci **7** (1998) 2469–2471
37. Orengo, C.A., Michie, A.D., Jones, S., Jones, D.T., Swindells, M.B., Thornton, J.M.: CATH—a hierarchic classification of protein domain structures. Structure **5**(8) (1997) 1093–1108
38. Katoh, K., Misawa, K., Kuma, K., Miyata, T.: MAFFT: a novel method for rapid multiple sequence alignment based on fast fourier transform. Nucleic Acids Res **30** (2002) 3059–3066
39. Katoh, K., Kuma, K., Toh, H., Miyata, T.: MAFFT version 5: improvement in accuracy of multiple sequence alignment. Nucleic Acids Res **33** (2005) 511–518
40. Notredame, C., Higgins, D., Heringa, J.: T-Coffee: a novel method for multiple sequence alignments. J Mol Biol **302** (2000) 205–217
41. Heringa, J.: Local weighting schemes for protein multiple sequence alignment. Computers and Chemistry **26** (2002) 459–477
42. Edgar, R.C.: MUSCLE: low-complexity multiple sequence alignment with T-Coffee accuracy. In: ISMB/ECCB. (2004)
43. Edgar, R.C.: Local homology recognition and distance measures in linear time using compressed amino acid alphabets. Nucleic Acids Res **32**(1) (2004) 380–385
44. Collins, M.: Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In: EMNLP. (2002)