Relational Sequence Alignments and Logos

Andreas Karwath and Kristian Kersting

University of Freiburg, Institute for Computer Science, Machine Learning Lab
Georges-Koehler-Allee, Building 079, 79110 Freiburg, Germany
{karwath,kersting}@informatik.uni-freiburg.de

Abstract. The need to measure sequence similarity arises in many applicitation domains and often coincides with sequence alignment: the more similar two sequences are, the better they can be aligned. Aligning sequences not only shows how similar sequences are, it also shows where there are differences and correspondences between the sequences.
Traditionally, the alignment has been considered for sequences of flat symbols only. Many real world sequences such as natural language sentences and protein secondary structures, however, exhibit rich internal structures. This is akin to the problem of dealing with structured examples studied in the field of inductive logic programming (ILP). In this paper, we introduce REAL, which is a powerful, yet simple approach to align sequence of structured symbols using well-established ILP distance measures within traditional alignment methods. Although straight-forward, experiments on protein data and Medline abstracts show that this approach works well in practice, that the resulting alignments can indeed provide more information than flat ones, and that they are meaningful to experts when represented graphically.

1 Introduction

Sequential data are ubiquitous and are of interest to many communities. Such data can be found in virtually all application areas of machine learning including computational biology, user modeling, speech recognition, empirical natural language processing, activity recognition, information extractions, etc. Therefore, it is not surprising that sequential data has been the subject of active research for decades. One of the many tasks investigated is that of *sequence alignment*. Informally speaking, a sequence alignment is a way of arranging sequences to emphasize their regions of similarity. Sequence alignments are employed in a variety of domains: in bioinformatics they are for instance used to identify similar DNA sequence, to produce phylogenetic trees, and to develop homology models of protein structures; in empirical language processing, they are for instance used for automatically summarizing, paraphrasing, and translating texts.

Most of the alignment approaches assume sequences of flat symbols. Many sequences occurring in real-world problems such as in computational biology, planning, and user modeling, natural language processing, however, exhibit internal structure. The elements of such sequences can be seen as atoms in a relational logic.

Example 1. Consider the following sentence adapted from [1]: '*A purple latex balloon blew himself up in a southern city Wednesday, bursting two other balloons and deforming 27*. The sentences actually provides much more complex data than shown. Applying Brill's rule-based part of speech tagger, cf. [2], which is one of the most widely used tools for assigning parts of speech to words, yields the following sequence of structured objects:

$$\texttt{dt(a)}, \texttt{jj(purple)}, \texttt{nn(latex)}, \texttt{nn(balloon)}, \texttt{vbd(blew)}, \texttt{prp(himself)}, \texttt{in(up)},$$
$$\texttt{in(in)}, \texttt{dt(a)}, \texttt{jj(southern)}, \texttt{nn(city)}, \texttt{nnp(wednesday)}, \texttt{comma}, \texttt{vbg(bursting)},$$
$$\texttt{cd(two)}, \texttt{jj(other)}, \texttt{nns(balloons)}, \texttt{cc(and)}, \texttt{vbg(deforming)}, \texttt{cd(27)}$$

The application of traditional alignment algorithms to such sequences requires one to either ignore the structure of the atoms, which results in a loss of information, or to take all possible combinations of arguments into account, which leads to a combinatorial explosion in the number of parameters. In other words, relational sequence alignment is a significant problem.

Surprisingly few works have investigated sequences of complex objects so far. Ketterlin [14] considered the clustering of sequences of complex objects but did not employ logical concepts. Likewise, Jiang *et al.* [11] and Weskamp *et al.* [27] proposed alignment algorithms for trees respectively graphs. Lee and De Raedt [15] and Jacobs [10] introduced ILP frameworks for reasoning and learning with relational sequences. Recently, Tobudic and Widmer [26] used relational instance-based learning for mining music data, where sequential, relational information is employed. To the best of our knowledge, however, none of these works investigate the alignment of relational sequences.

Indeed within bioinformatics most advances of sequence alignment for biological sequence analysis (see [6] for a good overview) have been made by incorporating additional sources of information such as sequence profiles or secondary structure predictions. As these works demonstrate, incorporating additional information can often yield considerable benefits to alignment quality. These methods, however, do not employ relational sequences, are domain-dependent and do not easily generalize across different domains. Therefore, Do *et al.* [5], McCallum *el al.* [17], Parker *et al.* [21] and Sato and Sakakibara [24] proposed more advanced probabilistic methods such as conditional random fields (CRFs) to discriminatively learn edit distances for propositional strings and trees [24]. CRFs allow to use arbitrary even relational features [8] to define the potential functions involved. This, however, leaves one with the difficult task of choosing the right representation or with the difficult task of automatically selecting the features form data, see e.g. [8]. This might explain why CRFs have so far not been used for aligning relational sequences.

In this context, we present REAL: a general, domain-independent approach to **r**elational s**e**quence **a**lignments and **l**ogos. The contributions of REAL are three-fold. First of all, REAL is a simple, yet powerful approach to align relational sequences. In particular, we propose to use well-established ILP distance measures within traditional alignment methods. Second, it defines the information content of relational sequence alignments. This is an important question

as it allows to evaluate alignments of and to find common motifs in relational sequences. Moreover, it can be graphically represented by so-called *relational sequence logos*, which are the third contribution of REAL. Although straightforward, experiments on real world data show that REAL works well in practice, that the resulting alignments can indeed provide more information than flat ones, and that the logos generated are meaningful to experts.

We proceed as follows. After discussing related work, we review basic alignment algorithms in Section 2. Then, we discuss relational sequences and relational distance measures in Section 3. Afterwards, in Section 3.1, we define the information content of relational sequence alignments. Based on this, we introduce relational sequences logos in Section 4. Before concluding, we empirically evaluate REAL on real-world data sets.

2 Sequence Alignment Algorithms

Alignment plays a major role in analyzing biological sequences. Consider e.g. the protein fold recognition problem, which is concerned with how proteins fold in nature, i.e., their three-dimensional structures. This is an important problem as the biological functions of proteins depend on the way they fold. Given a sequence of an unknown protein (query sequence) all approaches work in principle in a similar fashion: they scan an existing database of amino acids sequences (from more or less known proteins) and extract the most similar ones with regard to the query sequence. The result is usually a list, ordered by some score, with the best hits at the top of this list. The common approach for biologists, is to investigate these top scoring alignments or hits to conclude about the function, shape, or other features of query sequence.

One of the earliest alignment algorithm is that for global alignment by Needleman and Wunsch in 1970 [19]. The algorithm is based on dynamic programming, and finds the alignment of two sequences with the maximal overall similarity w.r.t. a given pairwise similarity model. In the biological domain, this similarity model is typically represented by pair-wise similarity or dissimilarity scores of pairs of amino acids. These scores are commonly specified by using a so-called similarity matrix, like the PAM [4] or BLOSUM [9] families of substitution matrices. The scores, or costs, associated with a match or mismatch between two amino acids, reflect to some extent the probability that this change in amino acids might have occurred over time of evolution.

More precisely, the Needleman-Wunsch algorithm proceeds as follows: initially, for two sequences of length m and n, a matrix with $m+1$ columns and $n+1$ rows is created. The matrix then is filled with the maximum score as follows:

$$M_{i,j} = \max \begin{cases} M_{i-1,j-1} + S_{i,j} & : \text{a match or mismatch} \\ M_{i,j-1} + w & : \text{a gap in the first sequence} \\ M_{i-1,j} + w & : \text{gap in the second sequence} \end{cases} \quad (1)$$

where $S_{i,j}$ is pairwise similarity of amino acids and w reflects a linear gap (insert step) penalty. The overall score of the alignment can be found in cell $M_{m,n}$.

To calculate the best *local* alignment of two sequences, one often employs the Smith-Waterman local alignment algorithm [25]. The main difference in this algorithm when compared to the Needleman-Wunsch algorithm, is that all negative scores are set to 0. When visualizing the resulting alignment matrix, strands of non negative numbers correspond to a good local alignment. For both algorithms versions using affine gaps costs exist, i.e. one employs different kind of gap costs for opening a gap or for extending one. To discourage the splitting of connected regions due the enforcement of a gap in the middle of the alignment, commonly extra gaps are allowed to be inserted at the end and at the beginning at either no additional costs or relatively low costs (padding costs).

In general, the alignments resulting from an global or local alignment, show then the more *conserved* regions between two sequences. To enhance the detection of these conserved regions, commonly multiple sequence alignments are constructed. Given a number of sequences belonging to the same class, i.e. in biological terms believed to belong to the same family, fold, or are otherwise somehow related, alignments are constructed aligning all sequences in one single alignment, a so-called profile. A common approach for the construction of a multiple alignment is a three step approach: First, all pairwise alignments are constructed. Second, using this information as starting point a phylogenetic tree is created as *guiding tree*. Third, using this tree, sequences are joined consecutively into one single alignment according to their similarity. This approach is known as the neighbour joining approach [23].

Example 2. Reconsider our natural language example from the beginning. Table 1 shows the global alignment of all five example sentences used by Barzilay and Lee [1] (adapted appropriately). As similarity measure we used the identity function, i.e., for instance $S(\texttt{balloon}, \texttt{balloon}) = 1$ but $S(\texttt{wednesday}, \texttt{sunday}) = 0$. The underlined sub-structures show the conserved regions computed by a propositional, global sequence alignment with arbitrarily chosen gap costs: gap opening cost 1.5, gap extention cost 0.5, and padding cost 0.25.

A good overview of alignment algorithms, including construction of multiple alignments and the generation of phylogenetic trees, can be found in [6].

3 Alignment of Sequences of Relational Objects

The alignment algorithms discussed in the previous section assume a given similarity measure $S_{i,j}$. Typically, this similarity measure is flat because the considered sequences consist of flat symbols. For instance the similarity measure used in Example 2 was simply the identity function. Many sequences occurring in real-world problems such as in computational biology, planning, user modeling, and natural language processing, however, exhibit internal structure. The elements of such sequences can elegantly be represented as objects in a relational logic (see e.g. [16] for an introduction to logic).

1. **A** purple **latex balloon blew himself up in** ***a*** southern city ***Wednesday*****, bursting** two other balloons and deforming 27.
2. **A latex balloon blew himself up in** ***the*** area of Freiburg, on ***Sunday*****, bursting** itself and disfiguring seven balloons.
3. **A latex balloon blew himself up in** ***the*** coastal resort of ***Cuxhaven*****, bursting** three other balloons and deforming dozens more.
4. **A** purple **latex balloon blew himself up in** ***a*** garden cafe on ***Saturday*****, bursting** 10 balloons and deforming 54.
5. **A latex balloon blew himself up in** ***the*** centre of Berlin on ***Sunday*****, bursting** three balloons as well as itself and disfiguring 40.

Table 1. Five sentences adapted from the example given by Barzilay and Lee [1]. Underlined words show the conserved regions (exact matches across all sequences) computed by a propositional sequence alignment using gap opening cost 1.5, gap extention cost 0.5, and padding cost 0.25. The bold parts denote the conserved regions of the corresponding relational sequence alignment using the same gap costs. The italic words show *lgg* conserved regions, i.e., the lgg of all atoms at a position exists.

Example 3. Recall the extended version of the *balloon* sentence in Example 1 $\texttt{dt}(\texttt{a}), \texttt{jj}(\texttt{purple}), \texttt{nn}(\texttt{latex}), \texttt{nn}(\texttt{balloon}), \texttt{vbd}(\texttt{blew}), \ldots$ representing determiners $\texttt{dt}(\textit{Word})$, nouns $\texttt{nn}(\textit{Word})$ etc. The secondary structure of the Ribosomal protein L4 can be represented as $\texttt{st}(\texttt{null}, \texttt{short})$, $\texttt{he}(\texttt{h}(\texttt{right}, \texttt{alpha}), \texttt{long})$, $\texttt{st}(\texttt{plus}, \texttt{short}), \ldots$ representing helices and strands of certain types, orientations, and lengths, $\texttt{he}(\textit{HelixType},\textit{Length})$ respectively $\texttt{st}(\textit{Orientation},\textit{Length})$.

The symbols `dt`, `nn`, ..., `st`, `null`, `short`, `he`, `h`, ... have an associated *arity*, i.e., number of arguments such as `st/2`, `he/2`, and `h/2` having arity 2, `dt/1` and `nn/1` having arity 1, and `plus/0`, `1/0`, having arity 0. A *structured term* is a placeholder or a symbol followed by its arguments in brackets such as $\texttt{nn}(\texttt{balloon})$, `medium`, $\texttt{h}(\texttt{right}, \texttt{X})$, and $\texttt{he}(\texttt{h}(\texttt{right}, \texttt{X}), \texttt{medium})$. A *ground term* is one that does not contain any variables such as $\texttt{nn}(\texttt{balloon})$, $\texttt{st}(\texttt{null}, \texttt{short})$, $\texttt{he}(\texttt{h}(\texttt{right}, \texttt{alpha}), \texttt{long})$, ...

Relational sequence alignment simply denotes the alignment of sequences of such structured terms. More formally, the relational alignment problem can be defined as follows.

Definition 1 (Relational Sequence Alignment Problem). *Let* $\texttt{x} = \langle \texttt{x}_i \rangle_{i=1}^{n}$, $n > 0$, *and* $\texttt{y} = \langle \texttt{y}_i \rangle_{i=1}^{m}$, $m > 0$, *be two sequences of logical objects and let* $S_{i,j}$ *be a similarity measure indicating the score of aligning object* $\texttt{x}_i$ *with object* $\texttt{y}_j$. *Then, the global alignment problem seeks to find the match with highest score of both sequences in their entirety. The local alignment problem seeks to find the subsequence match with highest score.*

One attractive way to solve this problem is to use a standard alignment algorithm but to replace the flat similarity measure $S_{i,j}$ in Eq. (1) by a structured one.

In this paper, we propose to use one of the many distance measures developed within Inductive Logic Programming [18]. As an example, consider one of the most basic measures proposed by Nienhuys-Cheng [20] [1]. It treats ground structured terms as hierarchies, where the top structure is most important and the deeper, nested sub-structures are less important. Let $\mathcal{S}$ denote the set of all symbols, then Nienhuys-Cheng distance d is inductively defined as follows:

$$\begin{array}{ll} \forall \mathtt{c}/0 \in \mathcal{S}: & d(\mathtt{c},\mathtt{c}) = 0 \\ \forall \mathtt{p}/n, \mathtt{q}/m \in \mathcal{S} : \mathtt{p}/n \neq \mathtt{q}/m : & d(\mathtt{p}(\mathtt{t}_1,\ldots,\mathtt{t}_n),\mathtt{q}(\mathtt{s}_1,\ldots,\mathtt{s}_m)) = 1 \\ \forall \mathtt{p}/n \in \mathcal{S}: & d(\mathtt{p}(\mathtt{t}_1,\ldots,\mathtt{t}_n),\mathtt{p}(\mathtt{s}_1,\ldots,\mathtt{s}_n)) = \frac{1}{2\mathtt{n}} \sum_{\mathtt{i}=1}^{\mathtt{n}} d(\mathtt{t}_\mathtt{i},\mathtt{s}_\mathtt{i}) \end{array}$$

For different symbols the distance is one; however, when the symbols are the same, the distance linearly decreases with the number of arguments that have different values, and is at most 0.5. The intuition is that longer tuples are more error-prone and that multiple errors in the same tuple are less likely.

Example 4. At this point the reader may verify that

$$\begin{aligned} d(\mathtt{nnp}(\mathtt{wednesday}),\mathtt{nnp}(\mathtt{wednesday})) &= 1/(2\cdot 0)\cdot(1) = 0.0 \\ d(\mathtt{nnp}(\mathtt{wednesday}),\mathtt{nnp}(\mathtt{sunday})) &= 1/(2\cdot 1)\cdot(0) = 0.5 \\ d(\mathtt{dt}(\mathtt{a}),\mathtt{dt}(\mathtt{the})) &= 1.0 \end{aligned}$$

so that it smooths the dichotomic identity function of the propositional case.

To solve the corresponding relational alignment problem, we simply set $S_{i,j} = 1 - d(\mathtt{x}_\mathtt{i},\mathtt{y}_\mathtt{i})$ in Equation (1).

Example 5. Continuing with our *Balloon* example but now employing the relational representation based on Brill's rule-based part of speech tagger, cf. [2], the bold parts in Table 1 show the conserved regions of the corresponding relational sequence alignment. We used the same gap costs as before but replaced the identity function by the Nienhuys-Cheng measure.. As one can see, the consensus regions of the propositional sequence alignment are proper sub-regions of the relational one.

3.1 Information Content

Now that we have introduced relational sequence alignments, we will investigate how informative they are. To this aim, we will introduce the concept of *information content* of relational sequence alignments. The information content is a significant concept as it allows to evaluate alignments of and to find common motifs in relational sequences. Moreover, it allows (see next Section) one to represented alignments graphically by so-called *relational sequence logos*.

[1] For sequences of more complex logical objects such as interpretations and queries, a different, appropriate similarity function has to be chosen. We refer to Jan Ramon's PhD Thesis [22] for a nice review of them.

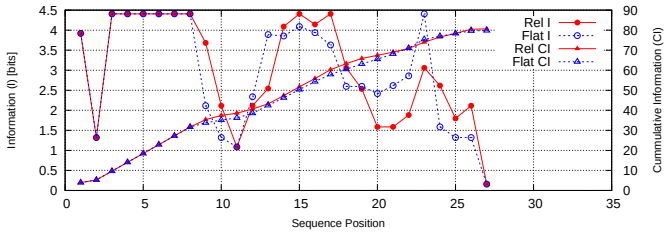


Fig. 1. Information content (IC) for the *balloon* example. The graph shows both the IC at each position (circle) and the cumulative IC (triangle) for the relational representation (solid, filled) and for the flat representation (dotted, unfilled).

Following Gorodkin *et al.* [7], the information content I_i of position i of a relational sequence alignment is

$$I_i = \sum\nolimits_{k \in G} I_{ik} = \sum\nolimits_{k \in G} q_{ik} \log_2 \left(\frac{q_{ik}}{p_k} \right),$$

where G is the Herbrand base over the language of the aligned sequences including gaps (denoted as '$-$') and q_{ik} is the fraction of ground atoms k at position i. When k is not a *gap*, we interpret p_k as the *a priori* distribution of the ground atom. Following Gorodkin *et al.*, we set $p_- = 1.0$, since then $q_{i-} \log_2(q_{i-}/p_-)$ is zero for q_{i-} equal to zero or one. For the work reported here, we set $p_k = 1/(|G| - 1)$ when $k \neq -$. The intuition is as follows:

> *if I_{ik} is negative, we observe fewer copies of ground atom k at position i than expected, and vice versa if I_{ik} is positive, we observe more of it.*

Example 6. Figure 1 shows the (cumulative) information content for our running *balloon* example. As prior we use the empirical frequencies over all five sentences. As one can see, both the relational and the flat representation agree on the information content for '*A [...] latex balloon blew himself up in [...]*'. They, however, disagree on the rest. Actually, the relational representation puts more information into the positions 14–18 whereas the flat representation put more information into the positions 19–23.

The total information content becomes $I = \sum_i I_i$ and can be used to evaluate relational sequence alignments.

Example 7. In the *balloon* example, the relational representation provides more information than the flat one, 80.7 vs. 79.8.

So far, we have defined the information content at the most informative level, namely the level of ground atoms. Relational sequences exhibit a rich internal structure and, due to that, multiple abstraction levels can be explored: variables allow to make abstraction of specific symbols. To compute the information content at a higher abstraction levels, i.e., of an atom a replacing all covered ground atoms k at position i, we view q_{ia} (resp. p_a) as the sum of q_{ik} (resp. p_k) of the ground atoms k covered by a.

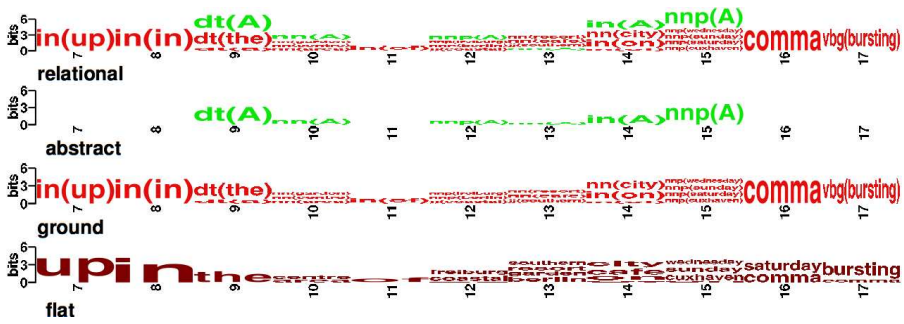


Fig. 2. Sequence logos (positions 7 – 17) for the *balloon* example (from bottom to top: flat , ground, abstract, and relational).

4 Relational Sequence Logos

Reconsider the alignment in Table 1. It consists of several lines of textual information. This makes it difficult – if not impossible – to read off information such as the general consensus of the sequences, the order of predominance of the symbols at every position, their relative frequencies, the amount of information present at every position, and significant locations within the sequences. In contrast, the corresponding sequence logo as shown in Figure 2 concentrates all of this into a single graphical representation. In other words, '*a logo says more than a thousand lines alignment*'.

Each position i in a *relational sequence logo* is represented by a stack consisting of the atoms at position i in the corresponding alignment. The height of the stack at position i indicates the information content I_i available. The height h_{ik} of each atom k at position i is proportional to its frequency relative to the expected frequency, i.e.,

$$h_{ik} = \alpha_i \cdot \left(\frac{q_{ik}}{p_k} \right) \cdot I_i \ ,$$

where α_i is a normalization constant. The atoms are sorted according to their heights. If I_{ik} is negative, the atom is shown upside-down.

Sequence logos at lower abstraction levels can become quite complex. Relational abstraction can be used to straighten them up. Reconsider Fig. 2. It also shows the logo at the highest abstraction level, where we considered as symbols the *least general generalization* of all ground atoms over the same predicate at each position in the alignment only. Because the prior probabilities change dramatically, the *abstract logo* looks totally different from the ground one. It actually highlights the determiner at position 9 and the propositional phrase at positions 14 and 15. Both views provide relevant information. *Relational logos* now combine both by putting at each position the individual stack items together and sort them in ascending order of heights.

To summarize, relational sequence logos illustrate that while relational alignments can be quite complex, they exhibit rich internal structures which, if exploited, can lead to new insights not present in flat alignments.

5 Experiments

Our intention here is to investigate to which extent relational sequence alignment is useful to analyze real-world data. More precisely, we investigated the following questions:

(Q1) *Can* Real*'s alignments be more informative than propositional ones?*
(Q2) *If so, can there be a gain in applications over propositional alignments?*
(Q3) *Can* Real *easily be applied across different domains?*
(Q4) *Is* Real *competitive with advanced ILP approaches?*

To this aim, we implemented Real in Python and Prolog and conducted a number of experiments on real-world data sets. In the following we will present their results.

5.1 (Q1) Alignment of Protein Sequences

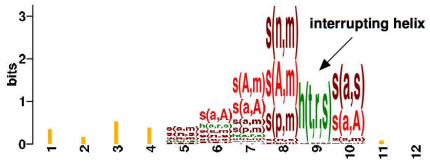
To answer **(Q1)**, we considered as real-world data set the five most populated folds in the SCOP class *Alpha and beta proteins (a/b)*, i.e., folds c.1, *TIM beta/alpha-barrel*, c.2, *NAD(P)-binding Rossmann-fold domains*, c.23, *Flavodoxin-like*, c.37, *P-loop containing nucleotide triphosphate hydrolases*, and c.55. *Ribonuclease H-like motif*. The examples are sequences of secondary structure elements of proteins which are similar in their three dimensional shape, but in general do not share a common ancestor (i.e. are not homologous). In total there are 2086 sequence distributed over the folds as follows: (c.1: 721), (c.2: 360), (c.23, 274), (c.37, 441), (c.55,290). The data set was generated using the ASTRAL database for the SCOP version 1.63[2].

We actually considered the subset of proteins which do not share more than 40 per cent amino acid sequence identity (*cut 40*). Overall, there are 522 example sequences (c.1: 182, c.2: 100, c.23: 66, c.37: 121, c.55: 53). We aligned sequences from one fold into a multiple alignment. Here we used the global alignment algorithm Needleman-Wunsch with affine gap penalties. The question of finding the appropriate gap costs in computational Biology is commonly answered by a trial and error approach. Here, we have solely concentrated on global alignments with affine gap costs using low padding costs. We have arbitrarily chosen the following gap costs: opening 1.5, extentsion 0.5, and padding 0.25.

Overall, Real yield a larger information content than the propositional approach (treating each ground atom as a different symbol). More precisely, the information contents for all folds were (relational/flat): c.1 (6.14/5.01), c.2 (7.66/7.54), c.23 (6.65/5.34), c.37 ($-0.12/-0.62$), c.55 ($1.05/-0.24$). Making gaps less expensive even increased the difference in information content. This affirmatively answers question **Q1**.

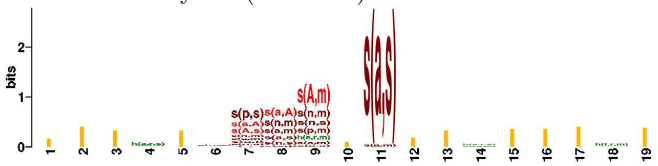
[2] http://astral.berkeley.edu/scopseq-1.63.html

Fold: SH3 (1 002 032)



SCOP: barrel, partly opened; np 1/4 4; Sp 1/4 8; meander; the last **strand** is **interrupted by** a turn of 310 **helix**

Fold: Barrel-sandwich hybrid (1 002 079)



SCOP: sandwich of half-barrel-shaped b-**sheets**

Fig. 3. Comparison of REAL's logos to SCOP descriptions for several folds. The logos are compared to the expert-like descriptions of those folds taken from the SCOP database (caption). **Bold** words denote matches.

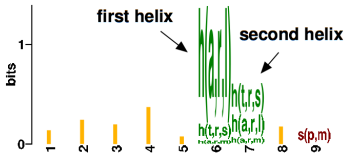
5.2 (Q2, Q3) Information Extraction

5.3 (Q4) Protein Fold Classification and Description

In general, however, more informative alignments can also come at an expense: even apparent unrelated sequences get higher similarity scores. For instance, in our protein sequence data set, we found sequences from different folds, where the relational alignment score is 4.75 times higher than the flat one. This can be a drawback in discriminative machine learning tasks. To validate this, we performed a 10-fold cross-validated nearest neighbour classification (k=7) on the cut 40 protein data set. This yielded 74.33% for the flat and 68.01% for the relational representation. On the full protein data sets, the predicative accuracies increase to 93.86% respectively 90.17%. The reason for the increase are obviously in the missing of close homologues in the cut 40 subset. Although, the experimental results favour the flat representation, the performances themselves are very good. They are comparable to more sophisticated statistical relational learning results on similar data: LoHHMs 74.0% [12], Fisher kernels 84% [13], CRFs 92.96% [8]. This tends to affirmatively answer **Q4**.

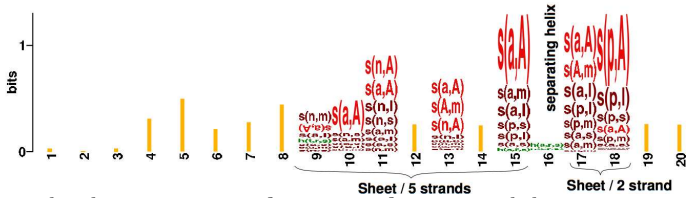
To further investigate **(Q4)** empirically, we investigated to which extend REAL's logos can be used to describe structural principles underlying SCOP folds. Understanding how proteins fold in nature, i.e., their three-dimensional shapre and structure is an important research question because the biological functions of proteins depend on the way they fold. We considered the SCOP protein data

Fold: Long a-hairpin (1 001 002)



SCOP:**two helices**; antiparallel hairpin, left-handed twist

Fold: Immunoglobulin (1 002 001)



SCOP: sandwich; **seven strands** in **two sheets**; greek-key; some members of the fold have additional strands

Fig. 4. Comparison of REAL's logos to SCOP descriptions for several folds. The logos are compared to the expert-like descriptions of those folds taken from the SCOP database (caption). **Bold** words denote matches.

set used by Cotes *et al.* [3]. We computed the logos for those protein folds for which Cotes *et al.* [3] provide the ILP rules computed using PROGOL. The logos together with a comparison to SCOP's expert-like descriptions of the folds are shown in Figures 3–6.

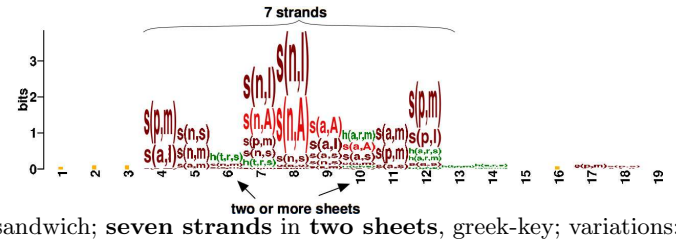
The relational logos match surprisingly well the fold descriptions[3]: only the parts of the SCOP descriptions, which can not be expressed using our simple protein representation, are missing and the relevant positions are highlighted due to relational abstraction. According to Cotes *et al.* [3], the logos can be considered to be meaningful to protein experts and, hence, a success in terms of the application domain. This clearly affirmatively answers **(Q4)**. In contrast to Cotes *et al.*'s ILP rules found using PROGOL, our discovered descriptions are less detailed and discriminative. This, however, is not surprising given the small amount of domain knowledge we used (particularly compared to Cotes *et al.*'s PROGOL approach).

6 Conclusions

We presented REAL, the first – to the best of our knowledge – alignment approach for relational sequences, i.e., sequences of logical objects. The experimental re-

[3] Using the flat representation, we were not able to discover the SCOP descriptions.

Fold: Prealbumin-like (1 002 003)



SCOP: sandwich; **seven strands** in **two sheets**, greek-key; variations: some members have additional one or two strands to common fold

Fig. 5. Comparison of REAL's logo to SCOP description for the Prealbumin-like fold. The logo is compared to the expert-like descriptions of this folds taken from the SCOP database (caption). **Bold** words denote matches.

sults clearly show that relational sequences alignments reveal useful information in practice across different domains and that they can indeed be more informative. REAL's alignments and logos are objective and reveal information not present in flat alignments such as the structural principles underlying protein folds in a way meaningful to experts.

REAL suggests a very interesting line of future research, namely to address the alignment of more complex logical objects such as interpretations, i.e., graphs. This has interesting applications e.g. in activity recognition, music mining, and plan recognition. Furthermore, extending CRF-based alignment methods [21, 8] to the relational case could be explored. Here, REAL should serve as a baseline.

Acknowledgments

The authors thank Luc De Raedt for his support, Ross King for helpful discussions, and Adrian Cootes, Stephen Muggleton, Michael Sternberg for providing their protein data. The research was partly supported by the EU IST programme: FP6-508861, *Application of Probabilistic ILP II*; FP6-516169, *Inductive Queries for Mining Patterns and Models*.

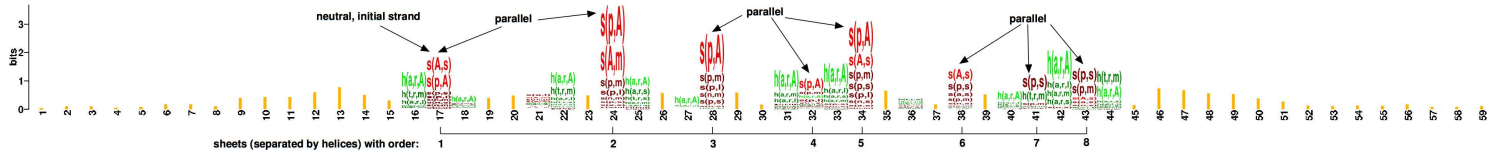
References

1. R. Barzilay and L. Lee. Learning to Paraphrase: An Unsupervised Approach Using Multiple-Sequence Alignment. In *Proc. of HLT-NAACL-03*, pages 16–23, 2003.
2. E. Brill. Some advances in rule-based part of speech tagging. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, 1994.
3. A. Cootes, S. H. Muggleton, and M. J. E. Sternberg. The automatic discovery of structural principles describing protein fold space. *Journal of Molecular Biology*, 330(4):839–850, 2003.
4. M. O. Dayhoff, R. M. Schwartz, and B. C. Orcutt. A model of evolutionary change in proteins. In M. O. Dayhoff, editor, *Atlas of Protein Sequence and Structure*, volume 5, chapter 22, pages 345–352. Nat. Biomedical Research Foundation, 1978.

5. C. B. Do, S. S. Gross, and S. Batzoglou. CONTRAlign: Discriminative Training for Protein Sequence Alignment. In A. Apostolico, C. Guerra, S. Istrail, P. Pevzner, and M. Waterman, editors, *In Proceedings of the Tenth Annual International Conference on Computational Molecular Biology (RECOMB 06)*, volume 3909 of *LNCS*, pages 60–74, Venice, Italy, April 2–5 2006. Springer.
6. R. Durbin, S. Eddy, A. Krogh, and G. Mitchinson. *Biological Sequence Analysis*. Cambridge University Press, 1998.
7. J. Gorodkin, L. J. Heyer, S. Brunak, and G. D. Stormo. Displaying the information contents of structural RNA alignments: the structure logos. *CABIOS*, 13(6):583–586, 1997.
8. B. Gutmann and K. Kersting. TildeCRF: Conditional Random Fields for Logical Sequence. In *Proceedings of the 15th European Conference on Machine Learning (ECML-06)*, 2006. (To appear).
9. S. Henikoff and J. G. Henikoff. Amino acid substitution matrices from protein blocks. *Proc. Natl Acad. Sci.*, 89:10915–10919, 1992.
10. N. Jacobs. *Relational Sequence Learning and User Modelling*. PhD thesis, Computer Science Department, Katholieke Universiteit Leuven, Belgium, 2004.
11. T. Jiang, L. Wang, and K. Zhang. Alignment of trees: an alternative to tree edit. *Theoretical Computer Science*, 143(1), 1995.
12. K. Kersting, L. De Raedt, and T. Raiko. Logial Hidden Markov Models. *Journal of Artificial Intelligence Research (JAIR)*, 25:425–456, 2006.
13. K. Kersting and T. Gärtner. Fisher Kernels for Logical Sequences. In *Proc. of 15th European Conference on Machine Learning (ECML-04)*, pages 205 – 216, 2004.
14. A. Ketterlin. Clustering Sequences of Complex Objects. In *Proc. of the 3rd Int. Conf. on Knowledge Discovery and Data Mining (KDD-97)*, pages 215–218, 1997.
15. S. D. Lee and L. De Raedt. Constraint Based Mining of First Order Sequences in SeqLog. In R. Meo, P. L. Lanzi, and M. Klemettine, editors, *Database Support for Data Mining Application*, pages 155–176. Springer, July 2004.
16. J. W. Lloyd. *Foundations of Logic Programming*. Springer, Berlin, 2. edition, 1989.
17. A. McCallum, K. Bellare, and F. Pereira. A Conditional Random Field for Discriminatively-trained Finite-state String Edit Distance. In F. Bacchus and T. Jaakkola, editors, *Proceedings of the Twenty-Firstst Conference on Uncertainty in Artificial Intelligence (UAI-05)*, Edinburgh, Scotland, July 26-29 2005.
18. S. H Muggleton and L. De Raedt. Inductive Logic Programming: Theory and Methods. *Journal of Logic Programming*, 19(20):629–679, 1994.
19. S. Needleman and C. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, 48(3):443–453, 1970.
20. S.-H. Nienhuys-Cheng. Distance between Herbrand interpretations: A measure for approximations to a target concept. In *Proc. of the 8. International Conference on Inductive Logic Programming (ILP-97)*, pages 250–260, 1997.
21. C. Parker, A. Fern, and P. Tadepalli. Gradient Boosting for Sequence Alignment. In Y. Gil and R. J. Mooney, editors, *Proceedings of National Conference on Artificial Intelligence (AAAI-06)*, Boston, Massachusetts, USA, July 16-20, 2006. AAAI, AAAI Press.
22. J. Ramon. *Clustering and instance based learning in first order logic*. PhD thesis, Department of Computer Science, K.U. Leuven, Leuven, Belgium, October 2002.
23. N. Saitou and M. Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Evol. Biol*, 4(4):406–425, 1987.
24. K. Sato and Y.Sakakribara. RNA secondary structural alignment with conditional random field. *Bioinformatics*, 25(Suppl. 2):ii237–ii242, 2005.

25. T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.
26. A. Tobudic and G. Widmer. Relational IBL in Classical Music. *Machine Learning*, 2006. (To be published).
27. N. Weskamp, E. Hllermeier, D. Kuhn, and G. Klebe. Graph Alignments: A New Concept to Detect Conserved Regions in Protein Active Sites. In R. Giegerich and J. Stoye, editors, *Proceedings German Conference on Bioinformatics*, pages 131–140, 2004.

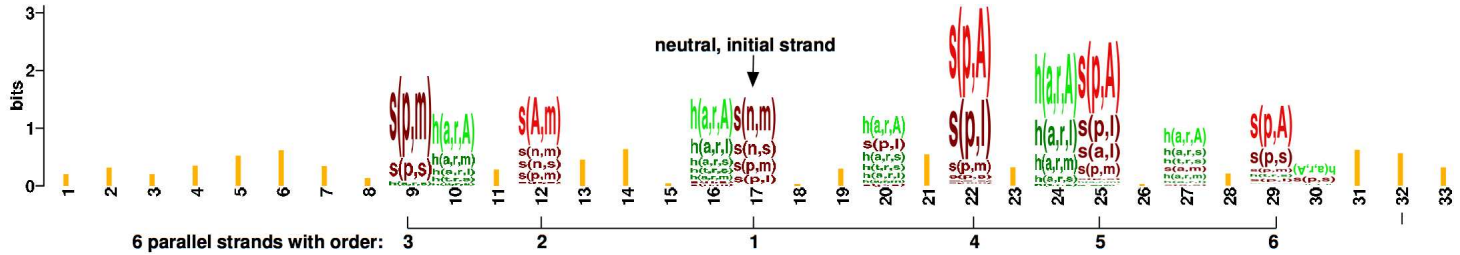
Fold: TIM barrel (1 003 001)



SCOP: contains **parallel** b-**sheet** barrel, closed; n 1/4 8; S 1/4 8; strand order **12345678**; the first six superfamilies have similar phosphate-binding sites

---

Fold: Rossmann-like (1 003 002)



SCOP: Core: three layers, a/b/a; **parallel** b-sheet of **six strands**, **order 321456**; The nucleotide-binding modes of this and the next two folds/superfamilies (1 003 003 and 1 003 004) are similar

Fig. 6. Comparison of REAL's logos to SCOP descriptions for the TIM barrel and the Rossmann-like folds. The logos are compared to the expert-like descriptions of those folds taken from the SCOP database (caption). **Bold** words denote matches.