Relational Sequence Alignment

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 information extraction, music mining, biological sequence analysis, and other 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 protein secondary structures, however, exhibit a 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 propose to use well-established ILP distance measures within alignment methods. Although straight-forward, our initial experimental results show that this approach performs well in practice and is worth to be explored.

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, however, exhibit internal structure. The elements of such sequences can be seen as atoms in a relational logic. 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.

The main contribution of the present paper is a general approach to align relational sequences, i.e., sequences of ground atoms. In particular, we propose to use well-established ILP distance measures within traditional alignment methods. Although straight-forward, our preliminary experimental results show that this approach performs well in practice and is worth to be explored.

We proceed as follows. After briefly reviewing alignment algorithms in Section 2, we discuss relational sequences and relational distance measures in Section 3. Before concluding, we present experimental results.

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 [15]. 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 [3] or BLOSUM [6] 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 [19]. 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 [18].

A good overview of alignment algorithms, including construction of multiple alignments and the generation of phylogenetic trees, can be found in Durbin *et al.* [4].

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. Many sequences occurring in real-world problems such as in computational biology, planning, and user modeling, however, exhibit internal structure. The elements of such sequences can elegantly be represented as objects in a relational logic (see e.g. [13] for an introduction to logic). For example, 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 of a certain type and length, `he`(*HelixType*,*Length*), and strands of a certain orientation and length, `st`(*Orientation*,*Length*). The symbols `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, 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 $\texttt{h}(\texttt{right},\texttt{X})$, `medium`, and $\texttt{he}(\texttt{h}(\texttt{right},\texttt{X}),\texttt{medium})$. A *ground term* is one that does not contain any variables such as $\texttt{st}(\texttt{null},\texttt{short})$, $\texttt{he}(\texttt{h}(\texttt{right},\texttt{alpha}),\texttt{long})$, ...

Relational sequence alignment simply denotes the alignment of sequences of such structured terms. More precisely, let $\mathtt{x} = \mathtt{x}_1, \ldots, \mathtt{x}_n$, $n > 0$, and $\mathtt{y} = \mathtt{y}_1, \ldots, \mathtt{y}_m$, $m > 0$, two sequences of logical objects and $d(i, j)$ a similarity measure indicating the score of aligning object $\mathtt{x}_i$ with object $\mathtt{y}_j$. 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.

Indeed, the only required task needed is to define the similarity measure $S_{i,j}$ in Equation (1). We propose to use one of the many distance measures developed within ILP [14]. As en example, consider one of the most basic measures proposed by Nienhuys-Cheng [16]. 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:

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

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). 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 [17] for a nice review of them.

4 Preliminary Experiments

Our intention here is to investigate to which extent relational sequence alignment is useful in real-world data sets. More precisely, we investigated the following two questions: **(Q1)** *Does the Nienhuys-Cheng measure provide better and more interesting alignments of sequences than a propositional one?* **(Q2)** *Is it possible to use relational sequence alignment for prediction purposes?* To this aim, we implemented the alignment method and the Nienhuys-Cheng distance measure in Python. In the following, we will describe some preliminary experiments carried out to investigate **Q1** and **Q2** and present their results.

4.1 Alignment of Protein Sequences

Here, we considered as real-world application the same data set as by Gutmann and Kersting [5], representing the five most populated folds in the SCOP class *Alpha and beta proteins (a/b)*. 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). We have performed the experiments on the complete set of example proteins, as well as on a subset of proteins which do not share more than 40 per cent amino acid sequence identity (*cut 40*). This subset was generated using the ASTRAL database for the SCOP version 1.63 [1]. Overall, there are 2082 example sequences.

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

Seq4	-	-	he_r_a_m	st_n_m	he_r_a_m	he_r_a_m	st_p_s	he_r_a_s	he_r_a_l	st_p_l
Seq3	he_r_a_l	he_r_3_s	he_r_a_l	st_n_s	he_r_a_s	he_r_a_m	st_p_s	he_r_a_l	st_p_s	he_r_a_s
Seq2	he_r_3_s	st_n_s	he_r_a_m	st_n_m	-	-	-	-	-	-
Seq1	st_n_m	st_p_m	he_r_a_l	-	st_p_m	he_r_a_m	st_p_s	he_r_a_m	st_p_s	he_r_a_s
(a)	*	*	*	*	*	*	*	*	*	*
(b)	*	*	**he(r,a,*)**	*	*	*	*	*	*	*

Seq4	-	-	he(r,a,m)	st(n,m)	he(r,a,m)	he(r,a,m)	st(p,s)	he(r,a,s)	-	-
Seq3	he(r,a,l)	he(r,3,s)	he(r,a,l)	st(n,s)	he(r,a,s)	he(r,a,m)	st(p,s)	he(r,a,l)	st(p,s)	he(r,a,s)
Seq2	he(r,3,s)	st(n,s)	he(r,a,m)	st(n,m)	he(r,3,s)	he(r,a,l)	st(p,m)	he(r,a,l)	-	-
Seq1	st(n,m)	st(p,m)	he(r,a,l)	st(p,m)	-	he(r,a,m)	st(p,s)	he(r,a,m)	st(p,s)	-
(c)	*	*	**he(r,a,*)**	**st(*,*)**	*	**he(r,a,*)**	**st(p,*)**	**he(r,a,*)**	*	*

Table 1. An alignment of four sequences using **(a)** the flat, **(b)** the back-translated flat, and **(c)** the relational approach. All symbols are abbreviated and the original alignment is truncated. The relational approach captures the conserved region much better than the flat ones as shown by the *lgg*-consensus sequences denoted in bold.

To answer question **Q1** we aligned sequences from one *fold* into a multiple alignment. Here we used the global alignment algorithm Needleman-Wunsch with affine gap penalties and seek to find *conserved* regions, i.e. subsequences in the multiple alignment, which express close similarity over all examples. 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: gap opening cost 1.5, gap extention cost 0.5, and padding cost 0.25.

To visualize the conserved regions, we have extracted the *consensus sequence* in form of the *lgg* (*least general generalization*) of all atoms in each particular position in the multiple alignment. An example of such a multiple alignment and the consensus sequence can be found in Table 4.1 **(c)**. Clearly, the consensus sequence reflects a conserved region of the four sequences.

The complete alignment possessed two conserved regions with a number of mismatches and gaps between. These conserved regions were not discovered when treating the sequence propositionally, i.e., each structured symbol $\texttt{st}(\texttt{null}, \texttt{short})$ is treated as a flat symbol `st_null_short`. In this case, using the Nienhuys-Cheng distance, only exact matches and pure mismatches are possible. Because none of the aligned flat symbols matches exactly, the resulting consensus sequence consist only of variables $*$, cf. Table 4.1 **(a)**. Even, when treating each of the aligned flat symbols as structured symbols again, the *lgg* consensus sequence does reveal much information, cf. Table 4.1 **(b)**. This affirmatively answers **Q1**.

The more informative consensus sequences, however, come at an expense: even apparent unrelated sequences get higher similarity scores. For instance, in our data set, we found sequences from different folds, where the relational alignment score is 4.75 times higher than the flat one. This could explain the

slightly lower predictive accuracy of the relational approach: a 10-fold cross-validated nearest neighbour classification (k=7) yielded 90.17% accuracy for the relational and 93.86% accuracy for the flat alignment approach for the complete dataset.

In any case, the predictive performances themselves are interesting. They are comparable to more sophisticated statistical relational learning results on similar data: LoHHMs 74.0% [8], Fisher kernels 84% [9], CRFs 92.96% [5]. This tends to affirmatively answer **Q2**.

For the *cut 40* subset, i.e. proteins in the five most populated classes not sharing more than 40 % amino acid sequence identity, the predictive performance decreases substantially for both representations: for the flat representation to 74.33 % and for the relational to 68.01 %. The reason for the decrease are obviously in the missing of close homologues in the cut 40 subset.

4.2 Alignment of Natural Language Sentences

Automatically paraphrasing sentences is of great practical importance for text-to-text NLP systems. Applications include text summarization and translation. For this task, Barzilay and Lee [1] proposed to use multiple (propositional) sequence alignment within clusters of similar sentences. Consider the following five sentences adapted from the example given by Barzilay and Lee:

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.

The underlined sub-structures show the conserved regions computed by a propositional sequence alignment using the same gap costs as in the protein experiment; the bold parts denote the conserved regions of the relational sequence alignment; and italic parts denote the use of *lgg*s. The relational representation allows to encode additional information for each sentences. In particular, we used 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, to annotate each word with its part of speech tag. This yielded sequences such as

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

Decreasing the gap opening costs to 0.5 resulted in

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.

In both cases, the consensus regions of the propositional sequence alignments are proper sub-regions of the relational ones. This affirmatively answers **Q1**.

5 Related Work and Conclusions

Surprisingly few works investigated sequences of complex objects. Ketterlin [11] considered the clustering of sequences of complex objects but did not employ logical concepts. Likewise, Weskamp *et al.* [21] proposed an alignment algorithm for graphs. Lee and De Raedt [12] and Jacobs [7] introduced ILP frameworks for reasoning and learning with relational sequences. Recently, Tobudic and Widmer [20] used relational instance-based learning for mining music data, where sequential information is employed. To the best of our knowledge, however, the present paper proposes the first alignment approach for relational sequences, i.e., sequences of logical objects. The preliminary experimental results indicate that the relational sequences alignment reveals useful information in practice for different domains. That they are indeed more informative has been recently confirmed by Kersting and Karwath [10] using an information-theoretic, empirical argument on the protein data set.

The approach presented 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.

Acknowledgments: The authors thank Luc De Raedt for his support and Ross King for helpful discussions. The research was 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. 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.
4. R. Durbin, S. Eddy, A. Krogh, and G. Mitchinson. *Biological Sequence Analysis*. Cambridge University Press, 1998.
5. 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).
6. S. Henikoff and J. G. Henikoff. Amino acid substitution matrices from protein blocks. *Proc. Natl Acad. Sci.*, 89:10915–10919, 1992.
7. N. Jacobs. *Relational Sequence Learning and User Modelling*. PhD thesis, Computer Science Department, Katholieke Universiteit Leuven, Belgium, 2004.
8. K. Kersting, L. De Raedt, and T. Raiko. Logial Hidden Markov Models. *Journal of Artificial Intelligence Research (JAIR)*, 25:425–456, 2006.
9. 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.
10. K. Kersting and A. Karwath. On Relational Sequence Alignments and Their Information Content. Short paper to be presented at the 16th International Conference on Inductive Logic Programming (ILP06), 2006.
11. 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.
12. 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.
13. J. W. Lloyd. *Foundations of Logic Programming*. Springer, Berlin, 2. edition, 1989.
14. S. H Muggleton and L. De Raedt. Inductive Logic Programming: Theory and Methods. *Journal of Logic Programming*, 19(20):629–679, 1994.
15. 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.
16. 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.
17. J. Ramon. *Clustering and instance based learning in first order logic*. PhD thesis, Department of Computer Science, K.U. Leuven, Leuven, Belgium, October 2002.
18. N. Saitou and M. Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Evol. Biol*, 4(4):406–425, 1987.
19. T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.
20. A. Tobudic and G. Widmer. Relational IBL in Classical Music. *Machine Learning*, 2006. (To be published).
21. 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.