2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)

Metric Learning on Biological Sequence Embeddings

Dhananjay Kimothi*†, Ankita Shukla*, Pravesh Biyani*, Saket Anand* and James M. Hogan†
* IIIT-Delhi, Delhi, India
† Queensland University of Technology (QUT), Australia

***Abstract*—Embedding techniques such as word2vec [1] have gained popularity due to their ability to represent words and their semantic variants as real valued vectors. Biological sequence analysis may also leverage unsupervised feature representations, augmented with supervised learning techniques for tasks like retrieval and classification. Algorithms that rely on distance metrics are computationally efficient and can handle large datasets, however, default distances in the embedded space often yield inadequate accuracy. In this paper, we use class labels to learn a Mahalanobis distance in the embedded feature vector space and show performance improvements over the default Euclidean metric in both retrieval and classification tasks. The approach may be readily generalised, and is and applicable to a wide range of problems in sequence analysis and others involving discrete entities or segmented data streams.**

I. Introduction

Similarity computation between entities within a data set is fundamental to clustering, classification and retrieval problems. Domains and data formats may vary considerably, from text and genomic sequences to audio and video recordings. Samples in a given data set are generally represented as feature points in a vector space and the distance computation is performed through measures like Euclidean distance or the cosine distance. The efficacy of a machine learning algorithm may crucially depend on the feature representations and the distance metrics that the feature space can support. In this paper, we investigate two learning tasks – retrieval and classification – over biological sequences, leveraging labeled data to learn an appropriate distance metric, while obtaining the underlying feature space embedding in an unsupervised manner.

Feature learning in natural language processing has recently received widespread attention thanks to the progress in word embedding techniques – representations allowing improved performance across a range of common machine learning tasks. Word embeddings essentially involve a mapping of the words in a text corpus to vectors in real number space ($\mathbb{R}^n$), where n is much smaller than the total dictionary size. A popular word embedding method, word2vec [1], uses a shallow neural network to build distributed embeddings. Word2vec is based on the distributional hypothesis [2] which states that words with similar distributional properties (i.e. those that co-occur regularly) also tend to exhibit semantic associations. Arora et. al. [3] present a theoretical discussion of embedding algorithms, while subsequent works [4], [5], [6] extend the word embeddings framework to sentence embeddings from the document corpus. Recent works [7], [8] have employed the word2vec framework in biological sequence analysis, obtaining representations for short subsequences (known as k-mers) and for entire sequences. One of the strengths of word2vec is that it is able to extract useful, low dimensional semantic embeddings even for large data sets. Note that these embedding algorithms are in some sense unsupervised, in that the embedding of a given word is independent of the label of the document it comes from.

The utility of embeddings lie in their potential as a feature set for machine learning techniques, which often rely on supervised learning to meet the downstream task requirements. While the embedded feature space roughly preserves semantic similarity, the lack of supervision limits the extent to which task-specific context is captured. For example, in a retrieval problem, the default Euclidean distance based ranking will ignore any task-conditional correlation between different feature dimensions. Similarly, a Euclidean distance based k nearest neighbor (kNN) classifier could prove to be suboptimal in a categorization task.

Using labeled data, we can learn a more appropriate distance metric in the embedded feature space, e.g., the Mahalanobis distance metric. The distance metric is learned by optimising an objective that effectively increases the separation between differently labeled data and reduces the separation between identically labeled data points. Metric learning approaches have been shown to work well for classification [9] and information retrieval tasks [10] in domains as variant as text and images.

In this work, we test the efficacy of using metric learning over the embeddings obtained for the labeled protein sequences in [8] and [7] to address retrieval and classification problems. These representations eliminate the need for specialized, often expensive, similarity measures designed for the input domain and provide a more computationally scalable alternative of simple distance computations between vectors – a critical requirement when dealing with huge bioinformatics data sets – thereby making metric learning an effective way of incorporating supervision. To perform metric learning over the protein sequence embeddings, we use the Sparse Compositional Metric Learning (SCML) algorithm [11] (please see section III). We observe that the learned distance metric over the embedded feature space improves performance for both the retrieval task – returning sequences belonging to the same protein family as the query; and for the classification

978-1-5090-3009-5/17/$31.00 ©2017 IEEE

task – predicting the species of a gene sequence. Finally, we conclude that using labeled data in effect improves the quality of embeddings for some of the learning tasks.

II. Related Work

A large number of metric learning approaches have been proposed over the last decade and successfully applied in various applications. While the literature of metric learning algorithms is vast, in this section we restrict our attention to approaches which align with the theme of this paper. Several methods ([12], [13], [14]) have been proposed to learn a metric for structured data. These methods directly operate on the data and learn an edit-distance metric that measures the cost of converting one object to another. However, these methods are restricted to sequences and do not perform well with feature vector representations. For a complete review on the developments in edit distance related metric learning, interested readers can refer to the survey by Bellet [15]. Similarly, Hua et. al. [16] learn a linear transformation with the aim of reducing the distance of each data point to its p nearest neighbors and increasing the distance to its p farthest neighbors. However, this approach does not affect the inter-class separation, thus limiting clustering and classification performance. Further, with the advent of deep learning, some studies have used metric learning on top of features obtained from a deep network for applications such as person re-identification [17]. In [18], low level features are improved by performing a series of non linear transformations using a deep network. This work optimizes an objective function that comprises a loss function based on the Euclidean distance between data samples, and a regularizer on the transformation. However, our work differs significantly from these approaches. In this paper, we propose a framework that benefits from both the nonlinearity of feature embedding obtained by Seq2Vec [8] and also a linear metric learned in the embedding space that ensures that distances computed in the transformed space are semantically consistent.

III. Proposed Framework

In this section we provide a brief review of the embedding and metric learning approaches used in our proposed framework, before considering our strategy in more detail.

A. Biological Sequence Representations: Seq2Vec and ProtVec

A biological sequence can be seen as a finite string over an appropriate symbol alphabet: for DNA the four nucleic acids (Adenine (A), Thymine (T), Cytosine (C), and Guanine (G)); and for protein sequences the set of 20 amino acids. Comparison of such sequences is a key step in bioinformatics search and analysis tasks. Traditionally, biological sequences have been compared through local [19] or global [20] alignment algorithms, methods based on Dynamic Programming. Alignment methods are quadratic in the sequence length and are not robust to large scale genomic re-arrangements. These limitations have given impetus to the development of alignment free methods for comparing sequences.

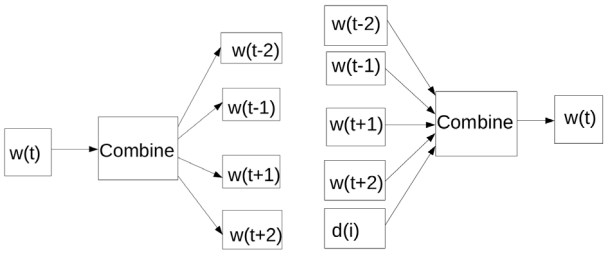

Fig. 1. Architecture for word2vec-skip gram and doc2vec-DM model. w(t) represents t^{th} word in the i^{th} document, d(i).

Alignment free sequence comparison methods (see Vinga et.al [21] for an early review) represent sequences as vectors within a vector space – usually based on a normalised count of the constituent k-mers – and employ some metric (e.g. Euclidean, d2, Kolmogrov complexity etc.) to quantify the similarity between these representations. While computationally efficient, these approaches may not prove as accurate as alignment based methods for some tasks, especially for short queries

Recent developments in word embeddings in NLP can also be exploited for representing biological sequences in a vector space. As noted above, two recent studies ([7] and [8]) employed word2vec[1] based models to learn a representation for *k-mers* extracted from the sequences. To obtain a sequence representation, [7] used a linear combination of the *k-mer* representations, while [8] used the doc2vec[4] model, naming the resulting framework *Seq2Vec*, for generating sequence representations. These models are shallow neural networks with only one hidden layer, where the vector of weights acts as a representation of the document or word. For training, these models employ a predictive task over the samples (a word-context pair) selected iteratively from the corpus of sentences. In word2vec, a word and its neighboring words (the context) make up the sample, whereas in doc2vec, the context also includes a tag associated with the document to which these words belong. Such tags are treated the same as context words, but are included only with the words coming from the corresponding document. In the classification task, the word/context is predicted given the context/word as input. The popular skip-gram (word2vec) and Distributed memory (DM;doc2vec) architectures are shown in Figure 1.

B. Mahalanobis Distance Metric Learning

Given the vectors $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^n$, the Mahalanobis distance between them, $d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j)$, is parametrized by the $n \times n$ matrix $\mathbf{M}$ and is given by

$$d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M} (\mathbf{x}_i - \mathbf{x}_j)}. \quad (1)$$

When $\mathbf{M} \succeq 0$ is Positive Semi-Definite (PSD), this distance function qualifies as a pseudo-metric[1] and satisfies the

[1]For a 'semi'definite $\mathbf{M}$, the condition $d_{\mathbf{M}}(\mathbf{x}_1, \mathbf{x}_2) = 0 \not\Rightarrow \mathbf{x}_1 = \mathbf{x}_2$, thus making $d_{\mathbf{M}}(\cdot)$ a 'pseudo' metric.

three properties: non negativity ($d_{\mathbf{M}}(\mathbf{x}_1, \mathbf{x}_2) \geq 0$), symmetry ($d_{\mathbf{M}}(\mathbf{x}_1, \mathbf{x}_2) = d_{\mathbf{M}}(\mathbf{x}_2, \mathbf{x}_1)$) and the triangle inequality ($d_{\mathbf{M}}(\mathbf{x}_1, \mathbf{x}_3) \leq d_{\mathbf{M}}(\mathbf{x}_1, \mathbf{x}_2) + d_{\mathbf{M}}(\mathbf{x}_2, \mathbf{x}_3)$). The goal of Mahalanobis distance metric learning is to learn a PSD matrix $\mathbf{M}$ such that $(d_{\mathbf{M}}(\cdot, \cdot))$ is small when the data points have the same label and large otherwise. We use the method proposed in [11] for metric learning and also briefly highlight the key points that make it suitable for our framework.

1) SCML: Sparse Compositional Metric Learning (SCML) [11] is a recent development in Mahalanobis distance metric learning. It constructs a PSD matrix by learning a sparse combination of rank-one, locally discriminative metrics extracted from the training data

$$\mathbf{M} = \sum_{i=1}^{K} w_i \mathbf{b}_i \mathbf{b}_i^\top \tag{2}$$

Here, $w_i \geq 0$ are the nonnegative weights corresponding to rank one metrics formed by the basis elements $\mathbf{b}_i \in \mathbb{R}^n$ extracted from several local regions of the training data. Typically, $K >> n$ and the set $B = \{\mathbf{b}_i\}_{i=1}^{K}$ forms an over-complete basis for the subspace spanned by the training samples.

SCML optimizes the following objective function that comprises a loss term for the distance constraints imposed on the labeled data and a regularizer on the weights

$$\min_{\mathbf{w}} \quad \frac{1}{|\mathcal{T}|} \sum_{(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k) \in \mathcal{T}} [1 + d_{\mathbf{w}}(\mathbf{x}_i, \mathbf{x}_j) - d_{\mathbf{w}}(\mathbf{x}_i, \mathbf{x}_k)]_+ + \beta \|\mathbf{w}\|_1 \tag{3}$$

Here, the first term is a large margin based hinge loss function $[\cdot]_+ = max(0, \cdot)$, $\mathcal{T}$ is the set of triplets, where distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ should be smaller that the distance between $\mathbf{x}_i$ and $\mathbf{x}_k$. Here $\mathbf{x}_i$ and $\mathbf{x}_j$ have same label while $\mathbf{x}_i$ and $\mathbf{x}_k$ have different labels and $\beta \geq 0$ is a regularization parameter.

The highlight of this approach is that the basis vectors are made available from the training data, thus reducing the number of learning parameters to K. This approach is suitable for embeddings as opposed to other state of the art techniques, where number of parameters increases quadratically with the dimension of the data. Moreover, most approaches perform a projection onto the PSD cone at every step, requiring eigenvalue decomposition and incurring an additional cost of $\mathcal{O}(n^3)$, thus limiting their applicability in high dimensions.

C. Metric Learning on Embeddings

Most of the protein sequences of a family share similarity over the entire length or contain smaller similar regions. But some sequences do not share strong similarity with other members of the family. The presence of such diverse sequences makes it difficult to obtain similar representations for all the members of family, even using the ProtVec and Seq2vec models, as these rely on shared patterns among sequences. In such cases, learning a suitable metric by using the labeled data has proven to measure the similarity effectively. Figure 2 shows the complete pipeline of our proposed framework;

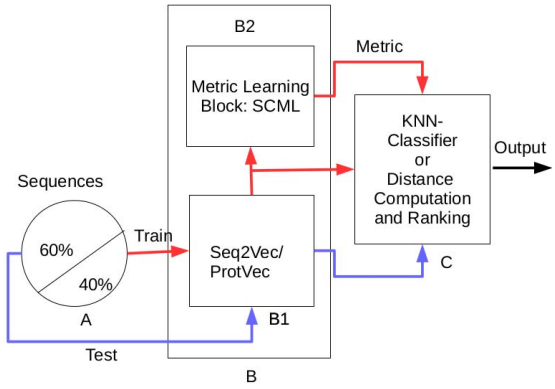

Fig. 2. Proposed Framework: Block A – Dataset(sequences), Block B(B1 and B2) – Seq2Vec/ProtVec (module for representation learning) and Metric Learning module (for learning metric over training feature vectors), Block C – ML technique for classification or retrieval

block A represents the available sequences that are divided into training-40% and testing-60% set, discussed in detail in section IV; the following block B represents our strategy and is explained with two blocks: Representation learning block (B1) and metric learning block (B2). Vector representations for all the sequences are learned either by Protvec or Seq2vec in block B1. The features corresponding to the training sequences are then passed to B2 to learn a distance metric. In the testing phase, the representations of the test sequences from block B1 are passed to block C, along with the metric learned from block B2 and the training sequences. For classification using the kNN classifier, block C outputs the class of the test sequence, whereas in the retrieval task it gives the ranked list of training sequences, based on distances from the test vector as computed with the metric learned from the training vectors.

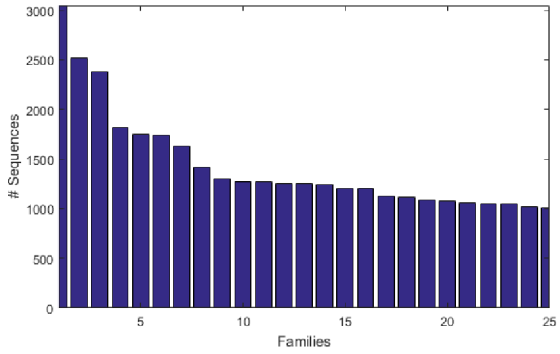

Fig. 3. Distribution of number of sequences

IV. Experiments and Results

As noted earlier, we evaluate the proposed framework on two problems in bioinformatics, namely, protein family prediction (classification) and homologous sequence retrieval.

	2 classes	4 classes
Euclidean Metric	1.47 ± 1.44	4.39±0.22
Learned Metric	1.10 ±1.01	3.45 ±0.21

TABLE I
10-NN CLASSIFICATION ERROR (IN %) FOR DIFFERENT NO OF CLASSES FOR SEQ2VEC REPRESENTATION WITH AND WITHOUT METRIC LEARNING. ALL RESULTS ARE AVERAGED OVER 5 RANDOM SPLITS OF DATA

A. Experimental Setup

For performance evaluation and comparison, we select a data set of 25 protein families (labeled classes) from the meta data provided by [7] and [22]. These families contain a varying number of sequences, ranging from 1000 to 3084. We use the Seq2Vec [8] method to generate the feature vectors of dimensions 100 and 250, while we utilize already learned feature vectors of 100 dimensions from ProtVec[2]. For training the Mahalanobis matrix $\mathbf{M}$ using SCML, the number of basis vectors K and the regularization parameter β are fixed to 5000 and 10^{-5} respectively. For each training sample in the training set, we identify 3 nearest neighbors (using Euclidean distance) having *same* label and 10 nearest neighbors having *different* labels. We use all combinations of these points and generate 30 triplet constraints for each training sample.

B. Protein family prediction

Identifying the family of an unlabeled protein sequence is an important classification problem in bioinformatics. To train the Mahalanobis distance matrix $\mathbf{M}$ using the SCML algorithm, we divide the data into 40% training set (annotated) while keeping the remaining 60% of the data as a test set (unannotated sequences). Also, in order to consider the uneven distribution (figure 3) of number of sequences across families, we selected 40% of the sequences from each family for training, while the remaining 60% sequences from each family were reserved for the test set.

We report results averaged over 5 such random splits of the data. Classification of sequences into protein families is performed using the NN algorithm. Distances are computed using the learned Mahalanobis matrix $\mathbf{M}$, and we compare it with the default of $\mathbf{M} = \mathbf{I}$ for both ProtVec and Seq2Vec based representations. Classifier performance is reported below for randomly selected sets of 2, 4 and 25 classes as noted above. Results in Table IV-B validate that the classification accuracy using the NN classifier increases when distances are calculated using the learned $\mathbf{M}$. The trends of increased accuracy continue to hold (table II) for the case where the number of classes was increased to 25. These results also suggest that metric learning consistently improves the classification accuracy as we vary the dimension of the representation.

A more detailed picture of classification accuracy for varying neighborhood size for the kNN classifier is provided in figure 5. The decrease in classification performance with increased neighborhood size may be attributed to the low inter class distances in embedded space. The superior performance observable in each case for the learned metric (the upper curve in each pair) suggests a possible increase in the inter class distances in the linearly transformed feature space.

C. Homologous sequence retrieval

Retrieving homologous sequences *i.e.* sequences which belong to the same family as that of the given query (sequence), is a challenging problem.

We evaluate our framework for this task. For data partitioning we follow the same setup as discussed in section IV-B, and report average results over 5 random splits of the data. In this experiment, for each test sequence (query), the training samples are ranked based on their distance from the query. Ideally, we expect to retrieve all sequences homologous to the query at the top of the ranked list. To evaluate the retrieval performance, we calculate precision at 11 recall levels, i.e. 0.0, 0.1,.....1; for 0.0 recall, the precision is considered to be 1.0 by definition. To bound the effort of searching for a relevant sequence, we limit our search to the top 1500 results. Precision at any given recall level is set to 0.0 if the relevant sequences at that level are not found in this group. The precision values for all possible queries for each of the classes are then averaged. We report these results in figure 4 for Seq2Vec based representations and ProtVec for the learned metric. It is evident from figure 4 that using a learned metric with these unsupervised embeddings performs better than Euclidean metric for similarity computation. Further, we observe that while the improvement in classification accuracy is not significant, the retrieval performance improves by a significant margin. This may be due do the discriminative nature of the distance constraints imposed in the SCML framework, that aims to reduce the distance between similar pair as compared to dissimilar pair by a given margin. This effectively improves the data distribution in the embedded space and allows one to retrieve sequences from same family over others given a query.e note in particular that the learned Seq2Vec metric provides strong early precision and maintains a clear advantage over other methods across the spectrum, although there remains some scope for improved precision at higher recall levels.

V. CONCLUSION

Word embeddings provide a meaningful mapping to a real valued vector space in a manner that preserves the proximity of features representing semantically similar words. Such embeddings can also capture the similarity between higher level structures such as sentences, and between entities in other domains such as bioinformatics, in which we are concerned primarily with similarity between biological sequences. In this work we add a layer of supervision into these earlier mappings.

[2]http://dx.doi.org/10.7910/DVN/JMFHTN.

	Seq2vec (250 dim)	Protvec (100 dim)	Seq2vec (100 dim)
Euclidean Metric	8.21 ±0.05	14.00 ±0.13	13.85 ±0.21
Learned Metric	7.56± 0.06	12.88 ±0.23	12.58± 0.22

TABLE II
10-NN CLASSIFICATION ERROR (IN %) FOR BOTH SEQ2VEC AND PROTVEC REPRESENTATIONS WITH AND WITHOUT METRIC LEARNING. ALL RESULTS ARE AVERAGED OVER 5 RANDOM SPLITS OF DATA

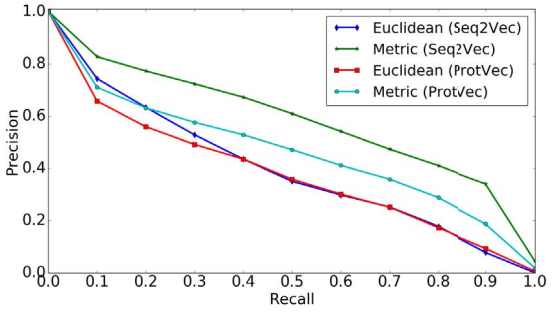

Fig. 4. Precision calculated at every 10% recall for Seq2Vec based representations and ProtVec, with and without metric learning. All results shown, are averaged over 5 random splits of data.

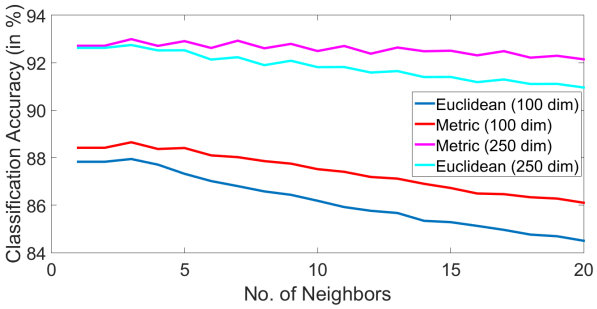

Fig. 5. Comparison of Classification Accuracy (in %) of Seq2vec representation with different embedding dimensions

By including labels of the data points, and using them to learn a task specific metric, we are able to demonstrate clear advantages over purely unsupervised embeddings for retrieval and classification tasks. In closing, we note that the approach is not limited to the tasks explored in this paper, and it appears straightforward to generalise these ideas to encompass other tasks in sequence analysis and to other problems involving discrete entities or segmented data streams.

REFERENCES

[1] T. Mikolov, G. Corrado, K. Chen, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *Proceedings of the International Conference on Learning Representations (ICLR 2013)*, pp. 1–12, 2013. [Online]. Available: http://arxiv.org/pdf/1301.3781v3.pdf

[2] J. R. Firth, "A synopsis of linguistic theory 1930-55." vol. 1952-59, pp. 1–32, 1957.

[3] S. Arora, Y. Li, Y. Liang, T. Ma, and A. Risteski, "Random walks on context spaces: Towards an explanation of the mysteries of semantic word embeddings," *CoRR*, vol. abs/1502.03520, 2015. [Online]. Available: http://arxiv.org/abs/1502.03520

[4] Q. Le and T. Mikolov, "Distributed Representations of Sentences and Documents," *International Conference on Machine Learning - ICML 2014*, vol. 32, pp. 1188–1196, 2014. [Online]. Available: http://arxiv.org/abs/1405.4053

[5] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler, "Skip-thought vectors," in *Advances in neural information processing systems*, 2015, pp. 3294–3302.

[6] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," *arXiv preprint arXiv:1503.00075*, 2015.

[7] E. Asgari and M. R. K. Mofrad, "Continuous Distributed Representation of Biological Sequences for Deep Proteomics and Genomics," *PLOS ONE*, vol. 10, no. 11, p. e0141287, nov 2015. [Online]. Available: http://dx.plos.org/10.1371/journal.pone.0141287

[8] D. Kimothi, A. Soni, P. Biyani, and J. M. Hogan, "Distributed representations for biological sequence analysis," *arXiv preprint arXiv:1608.05949*, 2016.

[9] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *Journal of Machine Learning Research*, vol. 10, no. Feb, pp. 207–244, 2009.

[10] P. Jain, B. Kulis, and K. Grauman, "Fast image search for learned metrics," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[11] Y. Shi, A. Bellet, and F. Sha, "Sparse compositional metric learning," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*. AAAI Press, 2014, pp. 2078–2084.

[12] H. Saigo, J.-P. Vert, and T. Akutsu, "Optimizing amino acid substitution matrices with a local alignment kernel," *BMC bioinformatics*, vol. 7, no. 1, p. 246, 2006.

[13] A. Bellet, A. Habrard, and M. Sebban, "Learning good edit similarities with generalization guarantees," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2011, pp. 188–203.

[14] ——, "Good edit similarity learning by loss minimization," *Machine Learning*, vol. 89, no. 1-2, pp. 5–35, 2012.

[15] ——, "A survey on metric learning for feature vectors and structured data," *arXiv preprint arXiv:1306.6709*, 2013.

[16] K. Hua, Q. Yu, and R. Zhang, "A guaranteed similarity metric learning framework for biological sequence comparison," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 13, no. 5, pp. 868–877, 2016.

[17] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Deep metric learning for person re-identification," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*. IEEE, 2014, pp. 34–39.

[18] J. Hu, J. Lu, and Y.-P. Tan, "Discriminative deep metric learning for face verification in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1875–1882.

[19] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences." *Journal of molecular biology*, vol. 147, no. 1, pp. 195–7, mar 1981. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/7265238

[20] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins." *Journal of molecular biology*, vol. 48, no. 3, pp. 443–53, mar 1970. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/5420325

[21] S. Vinga and J. Almeida, "Alignment-free sequence comparison–a review," *Bioinformatics*, vol. 19, no. 4, pp. 513–523, 2003. [Online]. Available: http://bioinformatics.oxfordjournals.org/content/19/4/513

[22] U. Consortium *et al.*, "Uniprot: a hub for protein information," *Nucleic acids research*, p. gku989, 2014.