Hybrid DNA Sequence Similarity Scheme for Training Support Vector Machines *

Mamoun Awad, and Latifur Khan
Department of Computer Science
University of Texas at Dallas
Richardson, Texas 75083, USA
[maa013600,lkhan]@utdallas.edu

Abstract
Similarity between two DNA sequences is based on alignment. There are different approaches of alignments; each has its own specialty of bearing different information on DNA sequence. This paper presents a study on similarity kernels based on different similarity schemes and proposes a hybrid one. Similarity Kernel is required in order to represent the distance or similarity between two DNA sequences. The different schemes of alignments and the cost of computing them, make it further more difficult to decide what scheme to use. In this study we combine different similarity schemes; each scheme is deduced based on alignment. We demonstrate that combining different similarity scheme does in fact generalize well in machine learning. The scoring scheme also turned to have impact on generalization.

1. Introduction

Many approaches have been proposed in the field of bio-informatics to classify DNA sequences. One of the most powerful techniques is Support Vector Machines, which was successfully applied on many real world problems, such as digit recognition, face detection, text categorization, and Protein/DNA classification/prediction [14,15]. Support Vector Machine showed a great deal in not over fitting features which can be explained by the VC theory [13][14]. The use of Kernel function has played an important role in machine learning in high dimensional space. Since not all problems are linear, then we can transform the data points to another high dimensional space such that the data points will be linearly separated. In the area of DNA and Protein Classification, the similarity kernel function should be chosen carefully to reflect and mimic the dot product operation, which the kernel function is based on.

It also needs to make sense in the score produced because this score will reflect how similar those two sequences/proteins are.

One of the subtle issues is this area, is how to represent an amino acid or DNA sequence. Many approaches were used including using frequencies and encoding. However, there are many problems with these methods. First the input space will be expanded unnecessarily large leaving large part of the space unused. And most importantly, second, the use of Euclidean space has no theoretic background in Biology or Chemistry and may reduce model accuracy.

Instead we can use similarity techniques, which were used in Biology to reflect a correct distance between two DNA sequences. Such approaches include global alignment, local alignment, semi-global alignment, etc.

This paper investigates the use of different similarity approaches and a hybrid approach to reflect the best score representing the distance between two DNA bases.

This paper is organized as follow. In section 2 we present related works in the field of DNA sequencing, and alignments. In sections 3 we introduce our approach of similarity. In section 4 we present our results, analysis, and rationale. In section 5 we state the conclusion, and in section 6 we show our intentions in future work.

2. Related Work

Many similarity approaches with different scoring schemes have been proposed. Among those were the global comparison, local comparison, and semi-global comparison. In global comparison, best alignment between two DNA sequences is sought by inserting spaces. In the local comparison, largest common substring between two sequences is sought. In the semi-global comparison, spaces inserted at the beginning and/or at the end of any of the two sequences are ignored. The work by Waterman [6] is a good review on optimal alignments, similarity, distance, and related algorithms. [8] by Needleman and Wunsch, is considered the first important contribution in sequence comparison from the point of view of biologists, although S.Ulam had already considered distances on sequence spaces in the 1950s. Similar phenomenon occurred with local alignments.

PAM matrices were introduced by Margaret Dayhoff and coworkers [9].

Most of the studies used only one kind of alignment to determine the similarity between two sequences. In using only one technique, we might discard information that can be useful and critical in other techniques. We investigate the use of each similarity alignment, its impact on training support vector machines, and we used a new hybrid approach.

3. Our approach

In this study, we are using different similarity approaches to find out which one best representing the similarity between two DNA sequences. In Biological point of view, similarity between two DNA sequences reflects the functional similarity, the existence of some common subsequence, or/and the existence of a mutation, which e.g. causes cancer.

We study the impact of each similarity approach not only separately, but also a combination of them, to see how well the trained support vector machine generalizes.

3.1 Problem statement.

Given a set of DNA sequences s_1, s_2... s_N, we want to find the

* This research was supported in part by gifts from SUN and NSF grants

similarity scheme, which is most suitable to represent the distance between two DNA sequences. The similarity scheme will be represented as a function, which is used as a kernel function in the support vector machine. Also we want to find out, is there any benefits from using a combination of different similarity schemes?

3.2 Support Vector Machine

Support Vector Machines (SVM) are learning systems that use a hypothesis space of linear functions in a high dimensional feature space, trained with a learning algorithm from optimization theory. This learning strategy, introduced by Vapnik and co-workers, is a principled and very powerful method that in the few years since its introduction has already outperformed most other systems in a wide variety of applications. SVM is based on the idea of hyper-plane classifier, or linearly separability. Suppose we have N training data points $\{(x_1, y_1), (x_2, y_2), (x_3, y_3) \ldots (x_N, y_N)\}$, where $x_i \in R^d$ and $y_i \in \{+1,-1\}$. We would like to learn a linear separating hyper-plane classifier:

$$f(x) = sign(\omega \cdot \chi - b) \qquad (1).$$

Furthermore , we want this hyper-plane to have the maximum separating margin with respect to the two classes. This problem can be formalized as:

$$Minimize_{(\omega,b)} 1/2\omega^T \omega \qquad (2)$$

$$Subject\ to\ \ y_i(\omega \cdot x_i - b) \geq 1 \qquad (3)$$

Although finding the perfect classifier is what we desire, in many applications it's reasonable to allow some noise or imperfect separation. In order to do that we introduce penalty variable C, where $c < \infty$ (If C = infinity, we come back to the original perfect separation case). We introduce non-negative slack variable $\epsilon_i >= 0$ so that:

$$y_i(\omega \cdot x_i - b) \geq 1 - \varepsilon_i \qquad (4)$$

$$\varepsilon_i \geq 0 \ \text{ for all i.} \qquad (5)$$

We also add to the objective function (2) a penalizing term $C(\sum_i \varepsilon_i)^m$ giving:

$$Minimize_{(\omega,b)} 1/2\omega^T \omega + C(\sum_i \varepsilon_i)^m$$

Where m is usually set to 1, which gives us:

$$Minimize\ \ _{(\omega,b)} 1/2\,\omega^T \omega + C\sum_i \varepsilon_i \quad (6)$$

$$\text{Subject to: } y_i(\omega \cdot x_i - b) \geq 1 - \varepsilon_i \geq 0 \quad (7)$$

Where $\varepsilon_i \geq 0$ for all i.

This is a convex quadratic programming problem (in ω, b), in a convex set. Introducing Lagrange multiplier $\alpha_i >= 0$, [2], and solving the Wolfe dual instead gives us:

Maximize

$$L_D \equiv \sum \alpha_i - 1/2 \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i . x_j \qquad (8)$$

Subject to:

$$C \geq \alpha_i \geq 0, \sum_i \alpha_i y_i = 0$$

When we solve α_i we can get $w = \sum_i^N \alpha_i y_i x_i$ and we can classify a new object x with

$$f(x) = sign(w.x + b)$$

$$= sign(\sum_i^N \alpha_i y_i (x_i . x) + b) \qquad (11)$$

Note that in the objective function and the solution, the training vectors x_i occur only in the form of dot product. Also α_i are Lagrange multipliers, one of each training point. When the maximal margin hyper-plane is found, only points which lie closest to the hyper-plane have $\alpha_i \geq 0$ and these points are called support vectors. All other points have $\alpha_i = 0$. This means that the representation of the hypothesis/classifier is given only by those points which lie closest to the hyper-plane and they are the most informative patterns in the data. Their number can also be used to give an independent bound on the reliability of the hypothesis/classifier.

3.3 Similarity

Each similarity scheme assigns score to the comparison of two DNA sequences. The more similar the comparison the bigger is the score assigned to it. Similarity score between two DNA sequences can be obtained easily by aligning them. Many different alignments can occur, but the best one is the one which aligns as many matching between DNA bases as possible.
In the following sections we explain the similarity schemes we consider in this study.

3.3.1 Global similarity.

Alignment of two sequences s_1 and s_2, can be defined as the insertion of one or more spaces in arbitrary locations in the DNA sequence, such that s_1 and s_2 end up with the same size. Having the same size, the augmented sequences s_1 and s_2 can be placed one over the other, creating correspondences between bases or spaces in the s_1 and bases or spaces in s_2. In addition, no space can be aligned in one sequence with space in the other. Given alignment between two sequences, we should have 3 parameters to tune the alignment process between two character α, β and μ as figure 2 shows.

α: The match score between two aligned characters.
β: The mismatch between two aligned characters.
μ: The Score assigned in alignment involving a space.
Such that
α > β AND
β > μ AND
α, β, μ ∈ R

Figure 2: The scoring scheme

Choosing the scores should be done carefully, especially we don't want to get a negative value as a final score for aligning two DNA sequences. That's because the score obtained will be used as a kernel function to train the support vector machines to classify DNA sequences. Since the kernel function is based on

dot products, and one of the metrics reflected by the dot product is distance, hence having a negative score has little sense.
The score is simply the summation of the score for each column in the alignment. For example, if we have the scoring scheme α = 0.6, β = -0.1, and μ = -0.2. Figure 3 shows an example of the score obtained from global alignments.
We used dynamic programming to obtain the scores. It basically consists of solving an instance of a problem by taking advantage of already computed solution for smaller instances of the same problem. We leave to the reader the opportunity to look at these algorithms in the references **[1].**

GA_CGGATTAG
GATCGGAATAG

(0.6 * 9) + (- 0.1 * 1) + (– 0.2 * 1) = 5.3

Figure 3: Global alignment score

t = ATTGTTGCT
s = TTATTG
q = CTTGCGCTT
α = 0.6
β = -0.1
μ = -0.2

Figure 4: Local alignment example.

3.3.2 Local similarity.
When the compared two sequences have different lengths, we end up inserting many spaces to align them globally. Meanwhile, two different sequences can be similar locally. A local alignment between s and t is an alignment between a substring of s and a substring of t. Not only the score of the longest substring between two sequences considered, but we added the score of all common substring between them. In this study, we considered all the common substrings between two sequences. Hence the score for the local alignment between two sequences is the summation of the scores of all common substrings between them. Further more, we obtained the average, hence:

Score = $\sum_{i=1}^{N} S_i / N$

Where S_i is the score for the substring S_i, and N is the total number of common substrings. Also notice that we can restrict the length of the common substrings between two sequences. In our study, we tried all the common substring greater that 3

3.3.3 Semi-global similarity.
In semi-global comparison, we score alignments ignoring some of the end spaces in the sequences. Hence in semi-global alignment, we ignore the end spaces, when aligning two sequences, i.e. we don't charge for spaces after the last character, or before the first character of the smaller sequence. For example, all the spaces in the second sequence in the alignment below are end spaces, while the single space in the first sequence is not an end space, where Ω denotes a space.

CAGC AΩCT TGGA TTCTCGG
ΩΩΩC AGCG TGGΩ ΩΩΩΩΩΩΩΩ

Notice that the lengths of the two sequences are different, and in order to align them, we added spaces. Of course adding spaces will decrease the score given to their alignment, even though they are similar.
Given alignment between two sequences, and scores value same as in figure 4.The global score given to this alignment is 12 * -0.2 (spaces) + .6 * 6 (match) + 1 * -0.1 (mismatch) = + 1.1
Notice that this is not the only alignment between those two sequences, here is another one:

CAGC ACTTG GAT TCTC GG
CAGC ΩΩΩΩΩ GΩT ΩΩΩΩ GG

The global score give to this alignment is:
10 * -0.2 (spaces) + .6 * 8 (match) = + 2.6, which is far better that the previous one.
Although we have a better score in the second alignment, the inserted spaces scattered the sequences, which made it less interests in the biology point of view. Inserting a space or a gap is considered as if a single mutation event which removed a whole of stretch of residues, while separated spaces are most probably due to a distinct events, and the occurrence of one event is more common that the occurrence of several events.
From the previous example, we can see that the new score given to the alignment is +3.5.

3.3.4 Hybrid similarity
In this study, we added one more scheme, which is the combination of all the previous alignment, to see the effects of that on training the support vector machine. We expect that combination of similarity scheme will affect positively the training process. By combining alignment, we mean adding the score of global, local, and semi-global scores. If we add all these scores together, this will add more meaning to each sequence toward other sequences. It could be the case that one sequence has small score in global alignment, but it has better one in local or semi-global alignments.

4. Results
We used the UCI Molecular Biology datasets, specifically the Promoter DNA dataset, and the Splice DNA data sets. For each dataset we train support vector machines differently as the following sections explain. We implemented the Kernel-Adatron Algorithm, and used it for classification and testing. One advantage of this algorithm is that it does not require vector manipulation, i.e. adding, subtracting, multiplication, a vector with other scalars. Especially we are using sequences and similarity scores, which might only be used to replace a kernel function

4.1 Promoter dataset
The promoter dataset is of 106 DNA sequences, each of length 57 base-pair. These sequences belong to two classes. Either the sequence is a member/non-member of class of sequences with biological promoter activity (promoters initiate the process of gene expression). In the following tables, the name data set is noted at the top of the table. We used two types of kernel, the linear kernel similarity function, in which we just used the similarity scores. The second one is quadric kernel, in which we used the $K(x_i, x_j) = < x_i . x_j >^2 - d$. The dot product is replaced with the similarity score obtained, raised to the power of 2.
The similarity scheme is global, local, semi-global or hybrid, which is as indicated before the summation of all scores. We used two different scoring schemes to charge for matching, mismatching, and spaces. Table1 shows two different scoring values we used in our tests.

Table1: the scoring types used in our test.

Score values	Score Type1	Score Type2
A (Match score)	0.6	0.5
β (Mismatch score)	-0.1	-0.25
μ (Space charge)	-0.2	-0.5

The last column of each table is the percentage of correct classification. We followed the leave-one-out approach to calculate the error of classification. For each dataset, we trained the support vector machine with all the dataset except for one picked randomly. Then we check to see if the picked sample data was classified correctly or not. We repeated that for each data set in the promoter data set because it's relatively small.

Table 2 shows that the best result obtained in the promoter data set is by using a semi-global scheme. The local scheme did very well and hence the hybrid scheme. Since the hybrid scheme is obtained by summing all scores, it's dependent on all of them, Hence we see that it scored very well, in both score types. As the table shows the semi-global and the local schemes have the best generalization results. The hybrid approach did very well because it combined local, global, and semi-global similarity. This is as if we combined many kind of similarity from different angles, and different views.

Even though the general score using the first score type was good, the second score type is far better than the first. We think that making the difference between match, mismatch, and penalty scores large has an implication on the results. For example, in the first score type, the difference is 0.7 between match and mismatch, and 0.8 between match and space charge respectively. Meanwhile, it's 0.75 and 1.0 in the second scoring type.

Table3 which uses quadric kernel $K(xi, xj) =< xi.xj >^2 -d$, shows a similar results to table2. The quadric kernel takes the dimensionality of feature space one level up. The difference between the two kernels, linear and quadric, is that the hybrid scheme did better than the local scheme using scoring type 1, and less using scoring type 2. This shows that no matter the variation in the similarity schemes, the hybrid scheme gives in between results.

4.2 Splice data set

The Splice dataset is larger than the previous one. It's 3190 DNA sequences, and there are three classes each sequence should belong to, EI, IE, or None. Splice junctions are points on a DNA sequence at which `superfluous' DNA is removed during the process of protein creation in higher organisms. The table columns have the same meaning as explained in the previous section. However there is a different in the way we trained the support vector machine, which affected the results. Since the splice data set was large, we picked randomly 300 sample data for training. The rest were used for testing the generalization.

Table 4 shows a resemblance results to the promoter database, table 2. Even though the local alignment didn't generalize well, the hybrid kept its ordered. The generalization results are lower than the ones in the promoter data set, because of the way we trained the support vector machines. We used 10% of the data set to train the support vector machines, mean while the rest, 90%, was used for testing. The Semi-global scores the best in both scoring type, then the hybrid, the global, and the local score the least.

Table 2: The promoter data set results, using linear similarity kernel. Ordered by best results

Similarity Scheme	**Score type**	**Correct classification rate (%)**
Semi-global	1	89.5
Semi-global	2	95
Local	1	87.5
Local	2	92.4
Hybrid	1	85.5
Hybrid	2	92
Global	1	80
Global	2	86.6

Table 5 which uses the quadric kernel:
$K(x_i, x_j) =< x_i.x_j >^2 -d$
It shows similar results to table 4. The quadric kernel takes the dimensionality of feature space one level up. The difference between the two kernels, linear and quadric, is that the hybrid scheme did better than the semi-global using scoring type 1 and type 2. This shows that no matter the variation in the similarity schemes, the hybrid scheme gives in between results.

Table 3: The promoter data set results, using quadric similarity kernel.

Similarity Scheme	**Score type**	**Correct classification rate (%)**
Semi-Global	1	83.8
Semi-Global	2	93.3
Hybrid	1	84.3
Hybrid	2	95.6
Local	1	95.6
Local	2	91.3
Global	1	79
Global	2	80.9

Table 4: The splice data set results, using linear similarity kernel ordered by the best results

Similarity Scheme	**Score type**	**Correct classification rate (%)**
Semi-Global	1	83.2
Semi-Global	2	87
Hybrid	1	81.2
Hybrid	2	87.5
Global	1	77.8
Global	2	86
Local	1	75
Local	2	80.3

Table 5: The splice data set results, using quadric similarity kernel. Ordered by the best results

Similarity Scheme	Score Type	Correct classification rate (%)
Hybrid	1	86
Hybrid	2	90
Semi-Global	1	86
Semi-Global	2	88.6
Global	1	85.7
Global	2	88.8
Local	1	78.5
Local	2	84.3

5 Conclusion and Further work

In this paper, we proposed a different similarity scheme study, where we have tested the impact of each on the generalization of the support vector machine. Interestingly enough, the hybrid similarity turned to generalize very well even when some other schemes did not. The scoring numbers assigned for aligning characters has also an impact on the results. Having the difference between the matching and mismatching/space bigger turned to be better. The experiment using the Splice data set showed that support vector machine generalize very well, even though the number of data set picked for training, was far less than the testing data. We will investigate more hybrid similarity scheme, more than just summing up different similarity schemes. We are also planning to study the effects of randomly choosing training data, and test data, as well as shuffle the training data set.

6. References

[1] Joao Carlos Setubal and Joao Meidanis, Introduction to Computational Molecular Biology. PWS Publishing Company. [1993].

[2] Colin Campbell, and Nello Cristianini Simple Learning Algorithms for Training Support Vector Machines.

[3] Promoter and splice datasets websites: http://www.ics.uci.edu/~mlearn/MLSummary.html

[4] N. Cristianini and J. Shawe-Tayler, Introduction to Support Vector Machines. Cambridge University Press 2000 ISBN: 0 521 78019 5

[5] W.R. Pearson and W.Miller. Dynamic programming algorithms for biological sequence comparison. In L. Brand nd M.L. Johnson, editors ,Numerical Computer Methods, volume 210 of Methods in Enzymology, pages 575-606. New York: Academic Press, 1992.

[6] M.S. Waterman, editor. Mathematical Methods for DNA sequences. Boca Raton, FL:CRC Press, 1989.

[7] G.von Heijne. Sequence Analysis in Molecular Biology: Treasure Trove or Trivial Pursuit? New York: Academic Press, 1987.

[8] 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, 48:443-453

[9] M. Dayhoff, R.M.Schwartz, and B.C. Orcutt. A model of evolutionary change in proteins. In M. Dayhoff, editor, Atlas of Protein Sequence and Structure,

[10] S.F Altschul. Amino acid substitution matrices from an information theoretical perspective. Journal of Molecular Biology, 219:555-565, 1991.

[11] J.Meidanis. Distance and similarity in the presence of non increasing gap-weighting functions. In proceedings of the Second South American Workshop on Sting Processing, Valparaiso, Chile, Apr, 1995.

[12] T.F Smith, M.S. Water, and W.M. Fitch. Comparative bio-sequence metrics. Journal of Molecular Evolution. 1981.

[13] Vapnik, V. (1995) The Nature of Statistical Learning Theory, Springer Verlag.

[14] Cortes, C. (1995) Prediction of Generalization Ability in Learning Machines. P.hD Thesis, Department of Computer Science, University of Rochester.

[15] LeCun, Y., Jackel, L. D., Bottou, L., Brunot, A., Cortes, C., Deker, J.S., Drucker, H., Guyon, I., Muller, U.A., Sackinger, E., Siard, P. and Vapnik, V., (1995)Comparison of Learning algorithms for handwritten recognition , Internatinal Conference on Artificial Neural Networks, Fogelman, F. and Gallinari, P. (Ed.), pp. 53-60.

[16] Batlett P., Shawe-Taylor J.,(1998). Generalization Performance of Support Vector Machines and other Pattern Classifiers. 'Advances in Kernel Methods- Support Vector Learning', Bernhard, Scholkopt, Christopher J.C. Burges, and Alexander J.Smola (eds), MTT Press, Cambridge, USA.