

# Evaluation of scoring functions for protein multiple sequence alignment using structural alignments

Sivan Yogev and Shlomo Moran \*

Department of Computer Science, Technion, Haifa, Israel

March 19, 2006

## Abstract

The process of aligning a group of protein sequences to obtain a meaningful Multiple Sequence Alignment (MSA) is a basic tool in current bioinformatic research. The development of new MSA algorithms raises the need for an efficient way to evaluate the quality of an alignment, in order to select the best alignment among the ones produced by the available algorithms. A natural way to evaluate the quality of alignments is by the use of *scoring functions*, which assigns for each alignment a number reflecting its quality. Different scoring functions for MSA were proposed over the years, which raised the need for methodological ways to assess the quality of such functions.

Few methods for assessing the quality of scoring functions for *pairwise* alignments were proposed. These methods are based on comparing alignments which are optimal for a given scoring function to *structural alignments* (alignments obtained through analysis of the 3 dimensional structures of related proteins). A main obstacle in using the above methods for evaluating scoring functions for alignments of  $k > 2$  sequences is the unavailability of efficient algorithms for computing optimal alignments (for a given scoring function) of even moderate number of sequences. We propose a framework for bypassing this difficulty, which is based on computing the correlation between suboptimal alignments.

An inherent issue that needs to be addressed in our method is the identification of an appropriate sample set of alignments to be used in the correlation test. We describe this problem, suggest a solution and report results using this solution.

Our results indicates that for most scoring functions, the addition of appropriate gap penalties improves the quality of the function. One notable exception is COFFEE, for which the average improvement after adding gap penalties was negligent in all of our experiments. COFFEE was also the best function in the average quality for the entire benchmark tested.

## Notations and Abbreviations

- MSA – Multiple Sequence Alignment
- NW – Needleman-Wunch algorithm/scoring function
- SoP – Sum of Pairs scoring function
- CoG – Center of Gravity scoring function

## 1 Introduction

The process of aligning a group of protein sequences to obtain a Multiple Sequence Alignment (MSA) of these sequences is a basic tool in current Bioinformatic research. Many algorithms and computer programs which perform this task exist, and new tools are constantly being developed and published (a recent overview of existing algorithms and description of a new one can be found in Do et al. [1]).

The multitude of available MSA tools raises the needs for a reliable method to evaluate the difficulty of aligning a given set of sequences, and compare the performance of different algorithms

---

\*Correspondence should be sent to moran@cs.technion.ac.il

on this set [10]. We focus on the comparison between alignments, in which a basic component is an appropriate *MSA scoring function* (also termed cost function or objective function). For each possible input to the MSA problem (a set of sequences) there is a set of *feasible solutions*, and the scoring function assigns for every feasible solution a numeric value (profit or cost). The scoring function should reflect the current knowledge we have from experimental data about protein MSA, and ideally we would like that the best alignment is the one with the highest score.

When only two sequences are to be aligned (pairwise alignment), a (global) alignment is commonly found by the Needleman-Wunch algorithm [12] with affine gap penalty (will be denoted NW). The scoring functions defined by this algorithm are given by two parameter sets – a substitution matrix  $S$ , and gap penalty parameters  $a$  and  $b$ , where the cost of introducing a gap of length  $l_{gap}$  into the alignment is  $a + b \cdot l_{gap}$ <sup>1</sup>. Formally, given a pairwise alignment  $A$ , let  $N_{x,y}(A)$  be the number of times the amino acid residues  $x$  and  $y$  are aligned in the same column in  $A$ ,  $N_g(A)$  be the number of gap sequences in  $A$  and  $N_i(A)$  be the number of indels in  $A$  (the sum of gap sequences lengths). The scoring function which is used in the NW algorithm can be formulated as:

$$NW(A) = \sum_{x,y} N_{x,y}(A) \cdot S(x,y) - a \cdot N_g(A) - b \cdot N_i(A) \quad (1)$$

And the complexity of finding the optimal alignment in the case of two sequences of length  $n$  is  $O(n^2)$ .

The question which parameters are the most appropriate to be used with the NW algorithm is of prime importance, and over the years many different substitution matrices were proposed (see [25] for a review). In the early 1990's the problem of selecting which matrix should be used in practice was examined in several works ([6, 9, 11, 25]). In the last 3 works the evaluation was done using pairwise alignments which are based on the three-dimensional folds of the corresponding proteins, obtained by X-ray crystallography and other methods. These alignments are called *structural alignments*, and since they reflect the experimental knowledge available for the aligned proteins they can be used as *reference alignments* – tools used to measure the quality of alignments of protein sequences.

In [25] Vogt et. al. present a systematic way to assess the quality of existing substitution matrices using structural alignments. The quality of a given substitution matrix,  $S$ , is expressed by the consistency of optimal alignments according to  $S$  with the structural alignments. This consistency was evaluated by applying the NW algorithm with  $S$  on a large number of protein sequence pairs with known structural alignment. The consistency between the resulting alignment and the structural alignment was measured for each pair of proteins, and a weighted average of the results for the given set of protein pairs was considered the overall consistency of  $S$ . This was further refined by using gap penalty parameters, which were optimized separately for each substitution matrix by testing a range of possible penalties, and selecting the combination with best overall consistency for this matrix. The results of this assessment were that the best matrix to use is the GONNET250 matrix published by Gonnet et al. [3], followed by the BLOSUM50 matrix published by Henikoff and Henikoff [5]. In a similar research, Gotoh [4] reached similar hierarchy – GONNET250 best, followed by BLOSUM62.

In this work we propose and implement a methodology for generalizing the comparative study of [25, 4] to alignments of more than two sequences. The main obstacle in such generalization is the unavailability of efficient algorithms for finding optimal alignments of  $k > 2$  sequences even for moderate values of  $k$ : the generalization of the NW algorithm to  $k$  sequences of length  $n$  has complexity of  $\Omega(n^k)$ , which is non-feasible for many of the protein sequences of interest. In fact, for the sum of pairs score (to be defined soon), it was shown in [26] and elsewhere that computing the optimal MSA score is NP-hard even when restricted to reasonable scoring matrices.

We tackle the above obstacle by considering a large number of suboptimal alignments, rather than focusing on a single optimal alignment. Specifically, we look for a scoring function which on the average has *higher correlation* with the structural score on appropriate sample set of alignments. The usefulness of this approach may depend on the correlation test used, and more than this on

<sup>1</sup>In order to avoid ambiguity in the usage of the term “gap”, *indel* will be used to describe a single insertion or deletion (represented in the alignment by the hyphen character ‘-’), while *gap* will be used to describe a sequence of adjacent indels.

right selection of the sample sets used in this test. For the correlation results to be meaningful, this set should contain only alignments which are relevant for current MSA scoring function. We develop a heuristic for selecting the sample set via a random process which selects only high scoring alignments, as described in sections 3 and 4.

Another important issue to be considered when generalizing scoring functions from pairwise to multiple alignments is that this generalization can be defined in more than one way. For given scoring parameters (substitution matrix and gap penalties), one possible generalization is to simply use equation (1) as is for multiple alignments. In another popular generalization termed Sum of Pairs (SoP), the NW score is calculated separately for every pair of sequences, and the sum of all these scores is taken. The pairwise alignments in this case are obtained by scanning the appropriate two rows of the multiple alignment, and omitting every column in which both rows contain indels. In this paper we provide an experimental evidence that the former method is usually better.

Although NW and SoP can be seen as the “natural” score functions for the MSA problem, other functions were proposed over the years, and some of them are used in current state of the art algorithms. One example is COFFEE, proposed by Notredame et al. [13]. COFFEE is based on the assumption that for every pair of sequences, there is an oracle which can provide an optimal *pairwise* alignment. Given this assumption, the best *multiple* alignment is the alignment which best keeps the optimal pairwise alignments of all sequence pairs. We implemented COFFEE using the result of the NW alignment algorithm as the optimal pairwise alignment. In order to find the best MSA scoring function, all three functions and others should be tested separately and compared. In a large percentage of the experiments in this work, COFFEE was better than NW, SoP and all of the other tested functions. In addition the influence of gap penalties was significantly lower for COFFEE in comparison to the rest of the scoring functions.

The rest of the paper is organized as follows. The following section describes the evaluation scheme in which structural alignments are used to test scoring functions using a correlation test on a sample set of alignments. Section 3 describes an alignments sampling method using random walks through an alignment “neighborhood” graph, and section 4 is dedicated to setting the endpoint of these random walks. Section 5 contains description of the correlation tests employed, and section 6 contains description of the scoring functions that were tested. The results of these tests are given in section 7, along with a discussion of the results. The last section includes a summary of this work, and a proposal of future research directions.

## 2 Evaluation of scoring functions using structural scores

An *MSA scoring function* is a function which assigns for each alignment a real number which should reflect the quality of the alignment. In order to assess different MSA scoring functions, we require two components:

1. A quantitative score of the accuracy of each alignment in accordance to experimental data,
2. A scheme which can determine how well a given scoring function approximates this quantitative score.

Once we have these, we search for a scoring function which best approximates the quantitative score.

The quantitative score in 1 above is based on current evaluation methods, which use restricted sets of structural alignments as *benchmark alignments* (see [1] and [14] for examples of a few databases of such alignments). Using the benchmark alignments, we can define *benchmark dependent scores*, which give reliable scores for this restricted input sets. An example for such benchmark of structural alignments is BALiBASE [22], which is a database that includes a total of 141 different “reference alignments”. Each reference alignment in the database contains the alignment itself, and a set of “core blocks”, which are the regions in the alignment that were reliably aligned according to the three-dimensional fold of the given proteins. For a given reference alignment  $R$ , denote the number of sequences in  $R$  by  $N$ , the number of columns in  $R$  by  $M$ , and the number of core blocks columns in  $R$  by  $m$ .  $SEQ(R)$  will represent the sequences which  $R$  is comprised of, without gaps.

The alignments are grouped into five different reference sets according to several characteristics such as sequences length, percent of identity between sequences and others [21]. We chose to use only the reference sets that were already included in the original version (and to exclude three new reference sets that were added to the last version of BALiBASE - version 2.0), since these sets were verified and corrected in the new version.

Motivated by similar study at Raghava et al. [14], we selected BALiBASE's score  $SPS$  [22] (also denoted the developer's score  $f_D$ ) for reflecting alignment quality. The  $SPS$  score of alignment  $A$  of  $SEQ(R)$  is defined as follows. Each occurrence of a pair of amino acid residues in the same column in some core block is a *core pair* (thus each core column with no indels has  $\binom{N}{2}$  core pairs). Let  $A$  be any alignment of the sequences in  $SEQ(R)$ . Then  $N_R(A)$  is the number of core pairs which remains in the same column in  $A$ .  $SPS(A)$  is  $N_{PR}(A)/N_{PR}(R)$ , that is the fraction of core pairs of  $R$  which are preserved in  $A$ . From this point on, the term BALiBASE score will relate to  $SPS$ . The second component in the assessment was implemented in [25] and others for a given *pairwise* scoring function, say  $F$ , by taking the structural scores of alignments of best possible  $F$ -score. This approach requires the computations of *optimal* alignments according to each tested scoring function. As mentioned above, such a computation becomes prohibitively costly even for alignments of moderate number of sequences. Therefore we replaced it by an extension of the same principle: An MSA scoring function  $F$  reflects the structural BALiBASE score well if the two are in good correlation with each other. Given a reference alignment  $R$  and a set  $S_R$  of alignments of  $SEQ(R)$ , the values  $SPS(A)$  and  $F(A)$  are calculated for each alignment  $A \in S_R$ , and the correlation between these values is the score of  $F$ . The scoring function we are searching for by this approach is the one with the highest correlation to the BALiBASE score.

### 3 Selecting the Sample Set

The selection of the alignments included in  $S_R$  affects the correlation level significantly. We are interested in alignments of relatively high BALiBASE scores, as alignments of low BALiBASE scores have also low scores by the scoring functions we wish to evaluate, and adding them to  $S_R$  is likely to add "noise" into the correlation test, which may result in lower resolution between the tested functions (see next section). Therefore, we insert to  $S_R$  only alignments whose BALiBASE score exceeds certain threshold, to be specified later. The sampling is done by a random walk through the set of the alignments, which we describe now.

For each reference alignment  $R$  we define a graph  $G_R = (V, E)$ , the "neighborhood graph" of  $R$ . The vertices in the graph are the alignments of  $SEQ(R)$ , and the neighbors of an alignment  $D \in V$  are alignments obtained by performing a *switch* of a residue and an *adjacent* indel. In other words, every edge  $e \in E$  connects two alignments which differ only by a single switch of a residue with an adjacent indel. It is not hard to see that  $G_R$  is a connected graph containing all the alignments of the sequences in  $SEQ(R)$ .

Formally, a random walk through the alignment space is a Markov process on  $G_R$ , beginning in the reference alignment  $R$ . It is done by an initialization step in which a list  $L$  which includes all possible switches for the current alignment (all edges in  $E$  which touch the vertex representing the current alignment) is calculated. After that, there is an iterative process in which a single switch from  $L$  is randomly selected and applied, and  $L$  is updated. In each iteration the BALiBASE score of the current sample is compared to a predefined *termination threshold* ( $TT_R$ ), and once the current score is lower than  $TT_R$  the random walk is terminated.

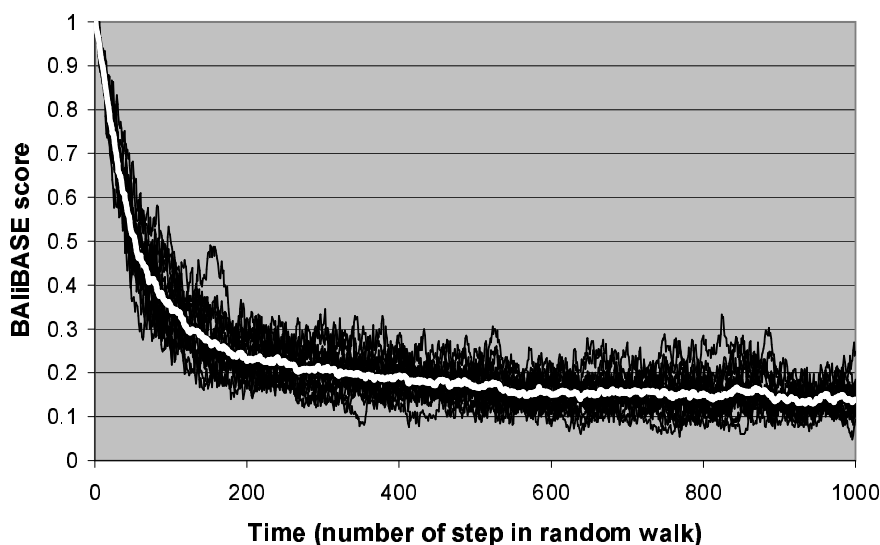
It is possible that some high scoring alignments, which are at large distance from the reference alignment, are selected by our sampling method only with a very small probability, since a random walk is likely to reach an alignment of score below  $TT_R$  before reaching such an alignment<sup>2</sup>. We tackle this problem by restricting the number of columns in the alignments at the sample set, thus excluding from this set alignments with many gaps, which are likely to be of low score. In our experiments we tested two bounds on this number: the first is  $M$  (the number of columns

<sup>2</sup>In fact, some alignments whose BALiBASE score is higher than  $TT_R$  can be reached only by passing through alignments whose BALiBASE score is lower than  $TT_R$ , and hence are never reached by our random walks. However it can be shown that the BALiBASE score of these alignments must be very close to  $TT_R$ , and the overall affect on  $S_R$  is negligible.

in the reference alignment  $R$ ), and the second is  $1.1M$ . Our results indicate that this addition of gap columns has meaningful influences on the results, in particular in cases where the structural alignment contains few gaps (see section 7.2).

As one can predict, our simulations showed that the BALiBASE scores of two adjacent alignments are often identical. As a result, the set of all alignments encountered during a single random walk contains many alignments with identical BALiBASE scores. To decrease this redundancy, rather than using all the alignments in the random walk, we take sample alignments during the walk. This is done by doing a series of switches before adding an alignment to  $S_R$ . The number of switches between samples is decided by the number of possible switches ( $\|L\|$ ) – after a sample is taken,  $\|L\|$  is calculated, and  $\|L\|/4$  consecutive switches are performed before taking the next sample. An example of the change in BALiBASE score during different random walks with this sampling method is given in figure 1.

Figure 1: BALiBASE score through random walks



20 random walks were performed for the reference alignment laboA from reference set 1. In each walk 1000 sample alignments were taken as described in the text, and the BALiBASE score of each sample was calculated. Black line represents a single random walks, where each point is an alignment for which the  $x$ -coordinate is the time (number of steps taken since beginning of random walk), and the  $y$ -coordinate is the BALiBASE score. The average of the 20 different scores for each of the  $x$  values of the walk is given in white.

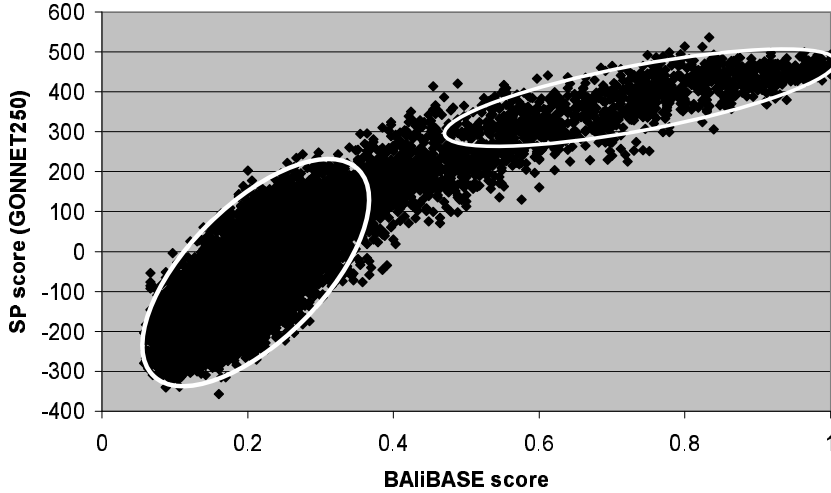
## 4 Setting the Termination Threshold

It is expected (and verified by our computations) that at the beginning of each random walk there will be a steep decrease in the BALiBASE score of consecutive samples, and after a while the decrease will become milder: In the beginning of every random walk, with high probability the BALiBASE score decreases at each random step. As the random walk continues, this probability decreases, and eventually the BALiBASE score of the samples stays in a certain restricted range. In figure 1, for example, after 200 samples the score in all the random walks gets below 0.3, and it stays approximately between 0.1 and 0.3 for many more samples. We wish to find a point where the decrease in the BALiBASE score becomes mild, meaning that the BALiBASE score reached a relatively low level.

As expected, simulations show that the correlation between the BALiBASE score and current scoring functions deteriorate as the BALiBASE score decreases. This is depicted in figure 2, which

plots the BALiBASE score ( $x$  axis) vs. the NW score with the GONNET250 matrix with both gap penalties set to 0 ( $y$  axis), for the samples included in figure 1.

Figure 2: BALiBASE score vs. NW score through random walks



20 random walks were performed for the reference alignment laboA from reference set 1. In each walk 1000 sample alignments were taken as described in the text, and for each sample two scores were calculated: BALiBASE score and NW score using GONNET250 matrix with gap penalties set to 0. Each point represents a single alignment. The two bounding white ellipses demonstrate the correlation between the two scores at high values (top right) and low values (bottom left).

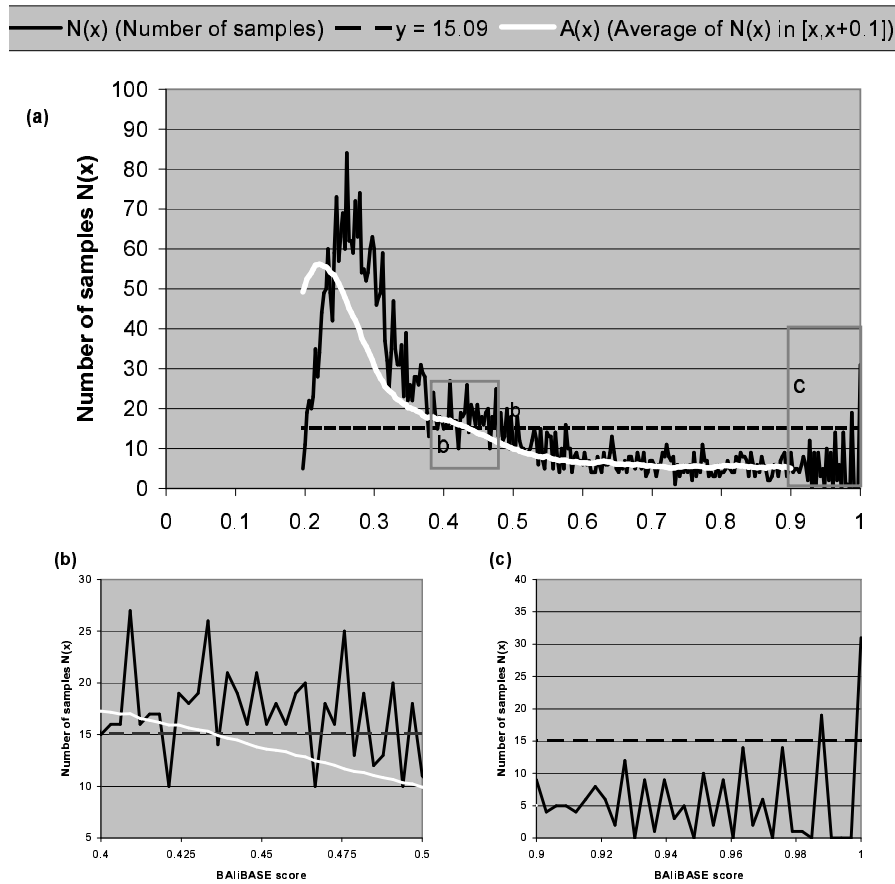
It can be seen that at about the same point where the decrease in BALiBASE score becomes milder in figure 1 ( $\sim 0.45$ ), there is also a change in the relation between the two scores. In high values of both scores (top right ellipse), the density of the points is relatively low, and their dispersion is not high - the bounding ellipse is long and narrow. In low values of both scores (bottom left ellipse), the density of the points is high, and their dispersion is higher - the bounding ellipse is shorter and wider.

Since we are interested in comparing the results of current MSA algorithms, it is important that the alignments obtained by these algorithms will not have BALiBASE scores below  $TT_R$ , in which case the correlation test is irrelevant to these alignments. It should be noted that since the average BALiBASE score when using state of the art algorithms on all the alignments in BALiBASE is approximately 0.9 [1], in most cases this restriction is not stringent.

The conclusion from the above considerations is that we should set  $TT_R$  high enough to exclude “irrelevant” alignments with too low BALiBASE score, but low enough to include all alignments whose BALiBASE score is higher than that obtained by a typical state of the art MSA scoring function. For that purpose, we performed 50 long random walks which ended at very low scores (guaranteed to be significantly lower than  $TT_R$ ), and the samples taken during these walks were combined into a large set of alignment samples. The BALiBASE score of every sample was calculated, and for each possible BALiBASE score  $0 \leq x \leq 1$ , the number of samples whose score is  $x$  was calculated, and will be denoted  $N(x)$ . Values of  $N(x)$  obtained by the same process with only 20 random walks are given in figure 3.

It can be seen that over the entire range there are local fluctuations in the values of  $N(x)$  (see Figure 3a), which are due to the discrete nature of the BALiBASE score and properties of the neighborhood graph. Since we are interested in trends rather than local phenomena, we eliminate the effect of these fluctuations by taking averages: Let  $A(x)$  denote the average of  $N(x)$  over the range  $[x, x + 0.1]$ . Then the local fluctuations in  $N(x)$  values are smoothed in  $A(x)$  (white line in 3a). Let  $x^0$  be the highest possible BALiBASE score smaller than 0.9 (i.e.

Figure 3: Number of samples  $N(x)$  in random walks



Number of samples  $N(x)$  in 20 random walks of the reference alignment laboA from reference set 1. The random walks ended when reached an alignment with BALiBASE score lower than 0.197. In (a), the entire range of possible BALiBASE score values is shown. In (c) only the range  $[0.9, 1]$  is shown.  $A(x)$ , which is the average of  $N(x)$  over the range  $[x, x + 0.1]$ , is given in white. The area in which  $A(x)$  crosses the line  $y = 3 \cdot A(x^0) = 15.09$  (dotted line), is enlarged in (b).

$x^0 = \max\{\frac{k}{SPS(R)} : k \in N \text{ and } \frac{k}{SPS(R)} < 0.9\}$ ,<sup>3</sup> Then  $A(x^0)$  represents the rate in which the BALiBASE score decreases at the beginning of the random walks. We wish to end the random walks when the rate of the decrease in the score becomes milder, i.e. when  $A(x)$  is significantly larger than  $A(x^0)$ . This is achieved by setting  $TT_R$  to be the maximal  $x < 0.9$  for which  $A(x) \geq 3 \cdot A(x^0)$ . Figure 3b includes the area at which  $A(x)$  crosses the line  $3 \cdot A(x^0)$ .

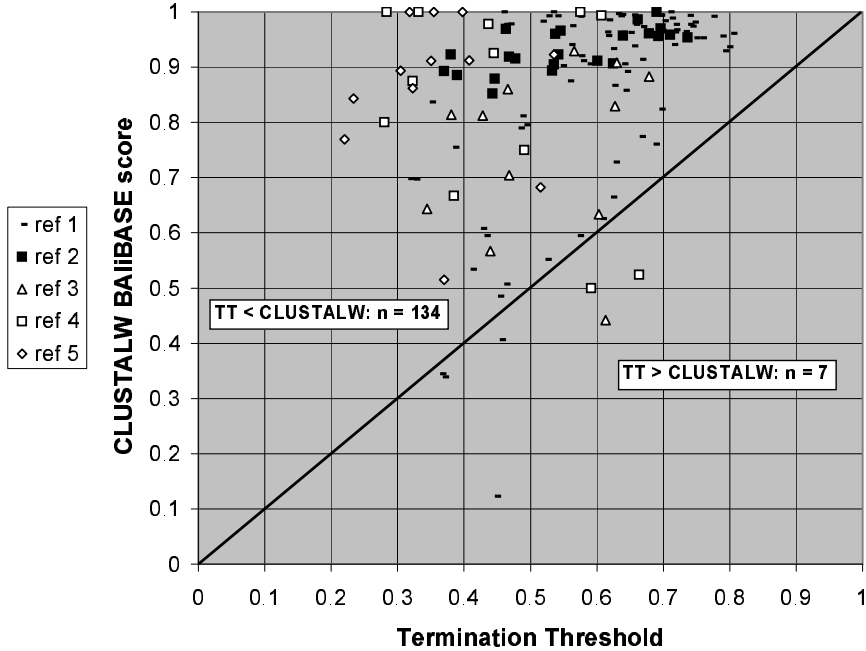
The  $TT_R$  values obtained by using several different multiplicative factors were compared with the BALiBASE score of the CLUSTALW [20] output on the input  $SEQ(R)$ . The factor of 3 was selected as a compromise between the requirement that  $TT_R$  will be high so that the sample set will not contain alignments with low BALiBASE score, and the requirement that  $TT_R$  will be lower than the BALiBASE score of the CLUSTALW output.

Given the first limit on the number of columns in the alignment, the average value for  $TT_R$  is relatively low (0.55), but on the other hand only in 7 out of the 141 alignments in BALiBASE  $TT_R$  is larger than the BALiBASE score of the CLUSTALW output. It can be seen in figure 4 that CLUSTALW finds alignments with low BALiBASE scores for these alignments, and that out

<sup>3</sup>We need  $x^0$  to be strictly smaller than 0.9 to exclude the count of samples with optimal BALiBASE score 1, which is biased upwards.

of the 7 alignments three are very close to the diagonal with  $TT_R - (\text{CLUSTALW score}) < 0.05$ , and only one is very far from the diagonal with  $TT_R - (\text{CLUSTALW score}) > 0.18$ .

Figure 4:  $TT_R$  vs. CLUSTALW BALiBASE score



Termination Threshold compared with BALiBASE score of CLUSTALW output. Each point represents one BALiBASE reference alignment. The shape of the point is according to the reference set the alignment was taken from.

Once  $TT_R$  is set for every reference alignment, the evaluation of a scoring function  $F$  is done by performing 50 random walks starting at  $R$  and ending when a score lower than  $TT_R$  is reached. For a given correlation test  $T$ , scoring function  $F$  and reference alignment  $R$ ,  $Q_T(F, R)$  is the value of the  $T$ -correlation between  $F$  and the BALiBASE score  $SPS$  on the sample set  $S_R$  (which was generated by our random walks with threshold  $TT_R$ ).

For example, when the random walks from figure 2 are terminated at the threshold  $TT_R = 0.43$  found in figure 3, only 1324 out of the original 20,000 samples are in  $S_R$ . Figure 5 plots the BALiBASE score vs. NW score for these alignments (similarly to figure 2).

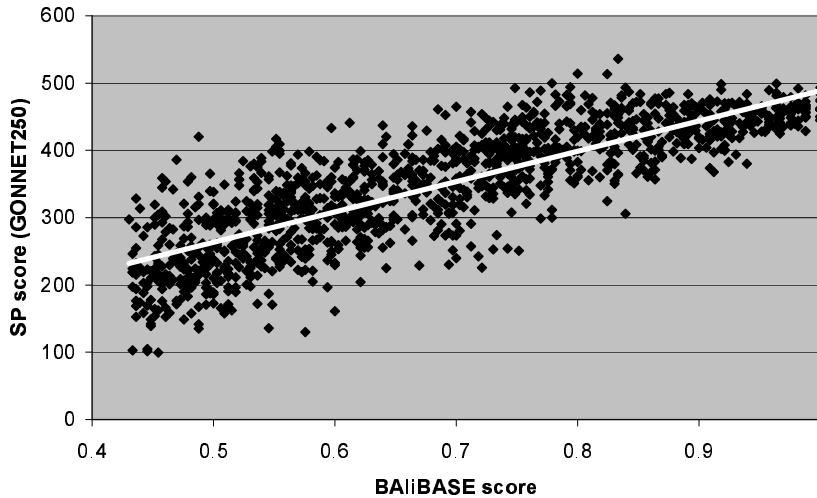
When using a group of reference alignments  $\Omega = R_1, \dots, R_n$ , the overall quality of  $F$  is the average of qualities:

$$Q_T(F, \Omega) = \frac{1}{n} \sum_{i=1}^n Q_T(F, R_i) \quad (2)$$

## 5 Correlation Tests

Given a reference alignment  $R$  and a sample set  $S_R$  of size  $N_S$ , denote the vector of BALiBASE scores of  $S_R$  by  $V_B$ , and the vector of scores obtained by applying a scoring function  $F$  on  $S_R$  by  $V_F$ . The two methods used to test the correlation between  $V_B$  and  $V_F$  are the *Pearson correlation coefficient* and the *Goodman-Kruskal  $\gamma$  (gamma) coefficient*. Because of the discrete nature of the BALiBASE score and the multiple random walks approach, ties may be abundant in the tested vectors (see for example figure 6). Other possible correlation measures such as *Kendall's  $\tau$  (tau)*, *Spearman's  $\rho$  (rho)* and *Spearman's footrule distance* were not used, since they are regarded less informative when the vectors contain many ties [18].

Figure 5: BALiBASE score vs. NW score on sample set



20 random walks were performed for the reference alignment laboA from reference set 1. Each walk was terminated at  $TT_R = 0.43$ , and for each of the resulting 1324 samples two scores were calculated: BALiBASE score and NW score using GONNET250 matrix with gap penalties set to 0. Each point represents a single alignment. The white line is the linear regression of the given points.

### 5.1 Pearson correlation coefficient

The Pearson correlation coefficient measures the normalized deviation of the samples from the linear regression between the two vectors. It is given by:

$$r = \frac{\sum V_{B_i} \cdot V_{F_i} - \frac{\sum V_{B_i} \sum V_{F_i}}{N_S}}{\sqrt{\left(\sum V_{B_i}^2 - \frac{(\sum V_{B_i})^2}{N_S}\right) \cdot \left(\sum V_{F_i}^2 - \frac{(\sum V_{F_i})^2}{N_S}\right)}} \quad (3)$$

With all summations on the range  $1 \leq i \leq N_S$ .

A confidence interval for the Pearson correlation coefficient can be calculated by transformation to Fisher's  $z'$ , which is normally distributed. After finding the confidence interval of  $z'$ , the confidence interval of the Pearson correlation coefficient is found by applying the reverse transformation [8].

### 5.2 $\gamma$ coefficient

When testing correlations, one possible approach is to ignore the actual values in the two vectors, and focus on the relative ordering of the samples. It is expected that vectors with high correlation will agree on the ordering of many pairs. Many tests based on this assumption exist, and since the data includes many ties, the most appropriate in this case is Goodman and Kruskal's  $\gamma$  coefficient.

Given  $i, j$  s.t.  $1 \leq i < j \leq N_S$ , let  $Mult(i, j) = (V_{B_i} - V_{B_j}) \cdot (V_{F_i} - V_{F_j})$ , and define:

$$Sign(i, j) = \begin{cases} 1 & \text{if } Mult(i, j) > 0; \\ -1 & \text{if } Mult(i, j) < 0; \\ 0 & \text{if } Mult(i, j) = 0. \end{cases}$$

Then, the  $\gamma$  coefficient is given by:

$$\gamma = \frac{\sum Sign(i, j)}{\sum |Sign(i, j)|} \quad (4)$$

With both summations on the range  $1 \leq i < j \leq N_S$ .

In the calculation of the  $\gamma$  coefficient, the ordering of each pair of samples in both vectors is tested separately.  $Sign(i, j)$  is 1 if the corresponding vectors agree on the relative order of samples  $i, j$ , and -1 if they disagree. Summation of  $Sign(i, j)$  over all possible  $(i, j)$  pairs gives the correlation value. In case there is a tie between a pair of samples in one of the vectors, their order in the other vector is irrelevant, and this pair of samples does not contain any ordering information. Therefore,  $Sign(i, j)$  is set to be 0 in such cases.

In order to normalize the value of  $\gamma$  to the range  $[-1, 1]$ , the sum of  $Sign(i, j)$  is divided by the number of relevant  $(i, j)$  pairs, which is the sum of the absolute values of  $Sign(i, j)$  for all possible  $(i, j)$  pairs. When there are no ties the denominator is reduced to the number of possible pairs,  $n(n-1)/2$ , and the score is identical to Kendall's  $\tau$ .

There is no robust method for calculating a confidence interval for the  $\gamma$  coefficient. Methods to find a confidence interval for the variant of Kendall's  $\tau$  termed  $\tau_b$  which is similar to the  $\gamma$  exist, but these methods are complex and result in approximated values [7]. We chose to test the robustness of the  $\gamma$  coefficient differently, as will be described in Section 7.

## 6 Scoring Functions

This section contains the definitions of the scoring functions for which the correlation to the BALiBASE score was tested. The functions are divided into two groups. The functions in the first group (4 functions) are based on pairwise alignment scoring, while the functions in the second group (5 functions) are based on column conservation measures. The last part of this section add gap penalties to each scoring function in two different ways.

### 6.1 Pairwise based functions

The two functions based on pairwise approach we tested are the functions described in section 1, NW/SoP and COFFEE, using the GONNET250 and BLOSUM62 matrices. As we do not use gap penalties in this subsection, the definition of the NW and SoP functions become identical. The NW score for MSA is a generalization of the NW objective function for the multiple sequences case. By the terminology used in equation 1, given a similarity matrix  $S$ :

$$NW(A) = \sum_{x,y} N_{x,y}(A) \cdot S(x, y) \quad (5)$$

The COFFEE function (Consistency based Objective Function For alignmEnt Evaluation) tests how good the MSA is, according to a library  $L$  of pairwise alignments. In our implementation of COFFEE, the library contains for each pair of sequences  $S_i$  and  $S_j$  the pairwise alignment  $L(S_i, S_j)$  obtained by applying the NW algorithm on  $S_i$  and  $S_j$ . We used two parameter sets for the NW algorithm to define two libraries. The first includes the BLOSUM62 matrix with gap opening penalty of 6.6 and gap extension penalty of 0.9. The second includes the GONNET250 matrix with gap opening penalty of 13.8 and gap extension penalty of 0.2. The gap penalties chosen are optimal according to Vogt et al.[25].

The pairwise projection of an MSA  $A$  on sequences  $S_i$  and  $S_j$ , to be denoted  $P_A(S_i, S_j)$ , is obtained by taking the rows of  $S_i$  and  $S_j$  from  $A$ , and removing all columns consisting two indels. Denote the length of  $P_A(S_i, S_j)$  by  $LEN_{i,j}(A)$ , and the consistency between  $P_A(S_i, S_j)$  and  $L(S_i, S_j)$  by  $SCORE_{i,j}(L, A)$ . Consistency is measured by the percentage of columns in  $P_A(S_i, S_j)$  which are identical to columns in  $L(S_i, S_j)$ . Then, the COFFEE score of  $A$  with the library  $L$  is given by:

$$COFFEE(L, A) = \frac{\sum SCORE_{i,j}(L, A)}{\sum LEN_{i,j}(A)} \quad (6)$$

With both summations on the range  $1 \leq i < j \leq N$  (recall that  $N$  is the number of sequences). Additional refinement of COFFEE through sequence pair weighting is possible (see [13]), but was not tested in this work.

## 6.2 Additional column based functions

A single MSA column represents a certain location in the protein. Assuming that mutations in locations which are important to the protein functionality are usually less beneficial to the organism, the level of conservation between the amino acids residues in the column can imply the importance of that column to the protein function. Valdar [24] thoroughly described existing functions for measuring conservation in a single MSA column. MSA scoring functions can be derived from these functions in a straightforward manner, by applying the function on every column in the MSA, and taking the sum over all columns as the final score. It should be noted that the NW/SoP score can also be viewed as such score - the column score is the sum of pair similarities according to the matrix  $S$ .

There are five categories of functions in Valdar's article. We chose to test functions from two of these categories - Mutation Data scores and Symbol Entropy scores. In addition, we tested a recently defined function which combines these two categories. The similarity matrix used in all these functions is BLOSUM62.

The function from the mutation data scores category is based on the score used in the software CLUSTALX [19], which is a graphical user interface to CLUSTALW. The score, which will be denoted Center of Gravity (CoG), is similar to the one plotted under every column of the alignment in the CLUSTALX user interface, reflecting the level of column conservation. Given a similarity matrix  $S$ , each amino acid is located in a 20-dimensional space. Denote the amino acids ( $a^1, a^2, \dots, a^{20}$ ) according to the order used in  $S$ , then the coordinates in the vector  $X_S(a^i)$  are the values of the  $i$ -th column in  $S$ :

$$X_S(a^i) = \begin{bmatrix} S_{i,1} \\ S_{i,2} \\ \cdot \\ \cdot \\ S_{i,20} \end{bmatrix}$$

Given a group of amino acids ( $a_1, a_2, \dots, a_n$ ), their consensus point is the center of gravity of all amino acids in the group:

$$\bar{X}_S = \frac{1}{n} \sum_{i=1}^n X_S(a_i) \quad (7)$$

Assume that the MSA column  $A_i$  includes the  $n$  amino acids ( $a_1, a_2, \dots, a_n$ ) and an additional  $(N - n)$  gaps, then the degree of conservation of this column is:

$$CoG_S(A_i) = \frac{n}{N} \cdot \sum_{i=1}^n |X_S(a_i) - \bar{X}_S| \quad (8)$$

Using Euclidean distance.

The function norMD [23] is also based on assigning for each amino acid a point in a 20-dimensional space according to the similarity matrix. However, since the quality of norMD was consistently worse than that of CoG, the values from the tests performed on norMD are not reported.

The Symbol Entropy score tested, is Shannon's information theoretic entropy [16]. Denote the fractional frequency of amino acid  $j$  in column  $A_i$  (without gaps) by  $p_j$ , then:

$$Entropy(A_i) = - \sum_{j=1}^{20} p_j \cdot \log(p_j) \quad (9)$$

Recently, Sattath and Margalit [15] suggested a generalization of Shannon's entropy called distance-entropy ( $dEntropy$ ), an entropy measure which enables integration of data from amino acids similarity matrices. This is done by finding an ultrametric approximation  $U_S$  to the similarity matrix  $S$  using the UPGMA algorithm [2]. Using the tree representation  $T_S$  of  $U_S$ , for each node  $u \in T_S$  let  $L(u)$  be twice the length of the edge connecting  $u$  to its parent ( $L(u) = 0$  for the tree

root), and let  $Leaves(u)$  be the set of leaves in the subtree whose root is  $u$ . Given that the fractional frequency of amino acid  $j$  in column  $A_i$  (without gaps) is  $p_j$ , define  $P_{A_i}(u) = \sum_{j \in Leaves(u)} p_j$ . Then, the distance-entropy of  $A_i$  is:

$$dEntropy(A_i) = - \sum_{u \in T_S} L(u) \cdot P_{A_i}(u) \cdot \log(P_{A_i}(u)) \quad (10)$$

This score does not take into account the number of gaps in the column, therefore it will be named  $dEntropy(nogaps)$ . Since the number of gaps can add information of the conservation level, we defined two  $dEntropy$  variants. In both of these variants the ultrametric tree  $U'_S$  consists only the amino acids in the relevant column. An additional leaf representing gap is added to the tree, with the root of  $U'_S$  as its parent. Each leaf (including the gap leaf) now contains the fractional frequency *with gaps*. Given these two differences, the calculation of  $dEntropy$  is the same as above.

In order to define the two variants it is necessary to describe how the ultrametric tree is obtained. In the first variant  $dEntropy(var.)$ , the input of UPGMA is the submatrix of  $S$  containing only amino acids in  $A_i$ . The second variant,  $dEntropy(const.)$ , is based on applying the UPGMA algorithm on the entire similarity matrix  $S$  as before, and then taking the subtree induced by the amino acids in  $A_i$ .

Since for all of the additional column based functions lower scores means better conservation, to get positive correlation with the BALiBASE score we use the negated sum:

$$TotalScore(A) = - \sum_{j=1}^M ColumnScore(A_i) \quad (11)$$

### 6.3 Gap penalties

The influence of gap penalty values on the alignment accuracy in the case of pairwise alignment was demonstrated by Vogt et al. ([25], table 12). In light of their results, it is expected that MSA scoring functions which penalize for gaps will perform better. The NW score which was defined in equation 1 (section 1) is one such function:

$$NW(A) = \sum_{x,y} N_{x,y}(A) \cdot S(x,y) - a \cdot N_g(A) - b \cdot N_i(A)$$

When  $N_g(A)$  is the number of gap sequences in  $A$ , and  $N_i(A)$  is the number of indels in  $A$  (the sum of gap sequences lengths). When testing NW with a given similarity matrix  $S$  the parameters  $a$  and  $b$  may influence this quality, and they can be optimized by testing over a range of reasonable values.

The same technique of penalizing for gaps and indels can be beneficial to the quality of every scoring function. In order to take this into account, this technique was employed on each of the 9 scoring functions tested, by taking an “actual score” for  $F$ :

$$F_{actual}(A) = F(A) - a \cdot N_g(A) - b \cdot N_i(A) \quad (12)$$

Which was optimized by testing  $a$  and  $b$  over the range  $[-15, 15]$ , with steps of size 0.25, and setting  $QS_T(F, \Omega)$  as the maximal value of  $Q_T(F_{actual}, \Omega)$  over the tested range.

While the range of scores obtained in our experiments on a given reference alignment for most tested functions is 100 or more, the values of COFFEE are in the range  $[0, 1]$ . Since  $N_g(A)$  and  $N_i(A)$  are typically large integers, every value of  $a$  and  $b$  in the tested range other than 0 results in decrease of the correlation with COFFEE. In order to explore the influence of gaps in more details with the same  $a$  and  $b$  values, the COFFEE scores were multiplied by 10,000, so that the actual score is:

$$F_{actual}(A) = 10,000 \cdot F(A) - a \cdot N_g(A) - b \cdot N_i(A) \quad (13)$$

Where  $a$  and  $b$  are tested over the range previously described. As a result of this multiplication the actual COFFEE results were improved in comparison to the original scores by values similar to those obtained by the rest of the tested functions.

Another way to count gaps was described in Section 1, in the context of the *SoP* score. The idea in this gap counting method is to sum the gaps separately for the sequence pairs pairwise projections. For each pair of sequences  $S_i$  and  $S_j$ ,  $N_g(P_A(S_i, S_j))$  and  $N_i(P_A(S_i, S_j))$  are calculated, and the total gap counts are given by:

$$N'_g(A) = \sum_{i,j} N_g(P_A(S_i, S_j)), N'_i(A) = \sum_{i,j} N_i(P_A(S_i, S_j)) \quad (14)$$

These values are used as before to define:

$$F'_{actual}(A) = F(A) - a \cdot N'_g(A) - b \cdot N'_i(A) \quad (15)$$

And  $QP_T(F, \Omega)$  is the maximal value of  $Q_T(F'_{actual}, \Omega)$  over the tested range (same as the range used to define  $QS_T(F, \Omega)$ ). Again, COFFEE scores were multiplied by 10,000 to get better resolution.

Thus, for each scoring function there are three quality scores which we will compare to each other. The first,  $Q_T(F, \Omega)$ , is calculated without gap penalties. The second,  $QS_T(F, \Omega)$ , is the maximal value of  $Q_T$  over the tested range of gap penalties when gaps are counted for single sequences, as in NW. The third,  $QP_T(F, \Omega)$ , is the maximal value of  $Q_T$  over the tested range of gap penalties when gaps are counted for pair sequences pairwise projections, as in SoP.

## 7 Results and Discussion

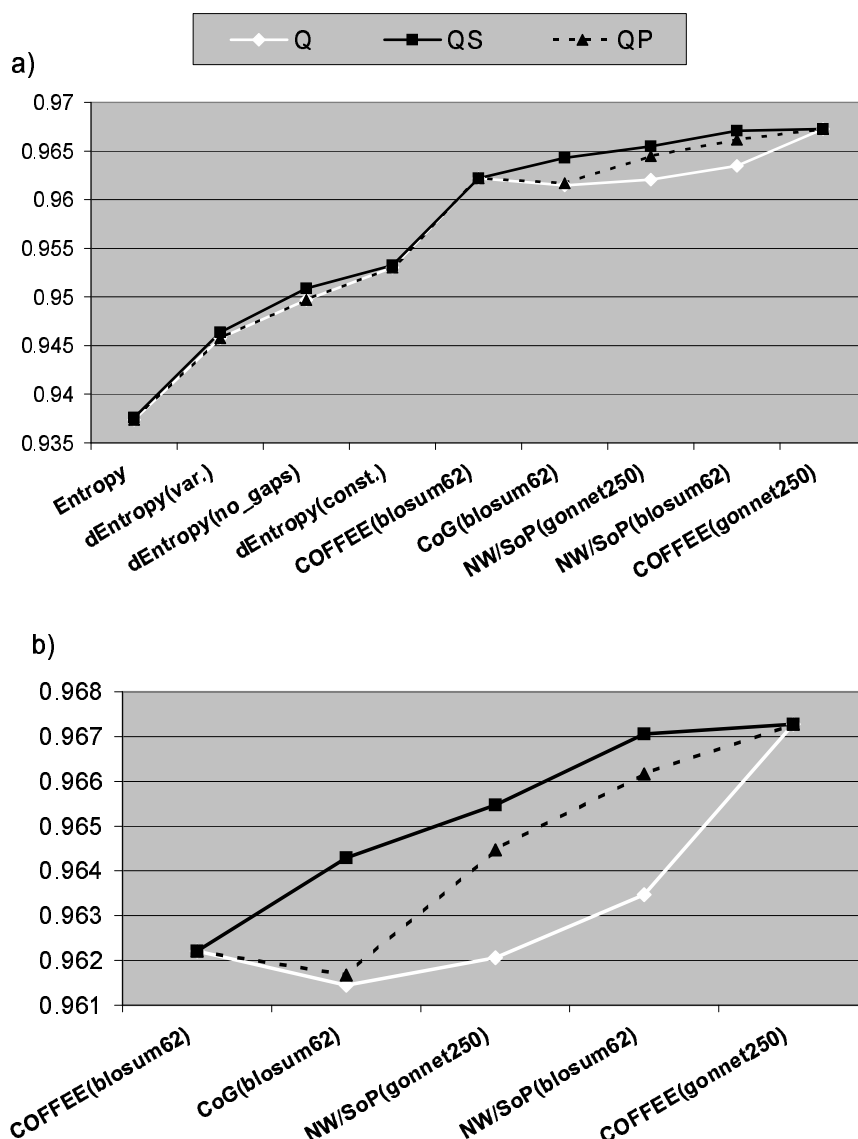
Let  $\Omega$  be the group of all BALiBASE alignments, and let  $\Omega_i$  be the different reference sets with  $i = 1, \dots, 5$ . The values of  $Q_r(F, \Omega)$ ,  $QS_r(F, \Omega)$  and  $QP_r(F, \Omega)$  for all tested functions are given in figure 6a. In addition, the confidence interval with probability 0.95 was calculated for every function and every alignment. For a set of alignments, the upper and lower limits of the interval are taken as the average of the limits over all alignments in the set.

In our tests, the 4 entropy-based functions were inferior to the other functions, with Shannon's entropy worst. Since d-entropy uses the same information as Shannon's entropy with additional data from similarity matrix, it is expected that the d-entropy functions will get better scores. The ranking of the three d-entropy functions reveals that it is better to use the ultrametric tree obtained by applying UPGMA on the entire similarity matrix, since both functions using this tree perform better than  $dEntropy(var.)$  in which a different tree construction is employed. The best d-entropy function is  $dEntropy(const.)$  – using the UPGMA tree with the entire similarity matrix, with incorporation of the number of gaps. Since for all entropy based functions the confidence interval was approximately  $\pm 0.0025$ , this ranking is significant.

The five functions with higher quality measures are given in better resolution in figure 6b. There is a large overlap between confidence intervals of the different functions, when all results of these functions are within the range  $[0.961, 0.968]$ , and the confidence intervals size is approximately  $\pm 0.0018$ . This means that it is not possible to make comprehensive conclusions regarding the ranking of these functions, and therefore in the remaining experiments we tested the correlations of the scoring functions in each reference group separately.

If we compare the three different gap penalties scoring, in many cases the difference between them is  $\sim 0.001$ , smaller than the confidence intervals. Statistically, the difference between these scores is insignificant, and when the test is done with the entire database none of them is preferable over the others. However, it can be seen that the non-gapped score ( $Q$ ) is always improved when gaps are introduced through single sequence counting ( $QS$ ), while with sequence pairs counting ( $QP$ ) the improvement is significant only in the SoP scores. In light of the above, from this point  $QS$  will be used as the primary score for comparison between scoring function.  $Q$  will be used to test the properties of different correlation tests, and  $QP$  will not be used. Because of the observed similarity between the scores, and since the calculation of  $Q$  is less complex than that of the other measures, a possible conclusion of our test is that when the computational aspect is important,  $Q$  is preferable over  $QS$ . This conclusion should be made with caution, since because of the limit on the number of columns in the reported tests, it can be applied only when the number of columns in the tested alignments is close to that of an unknown structural alignment.

Figure 6: Quality values for entire BALiBASE, using Pearson correlation



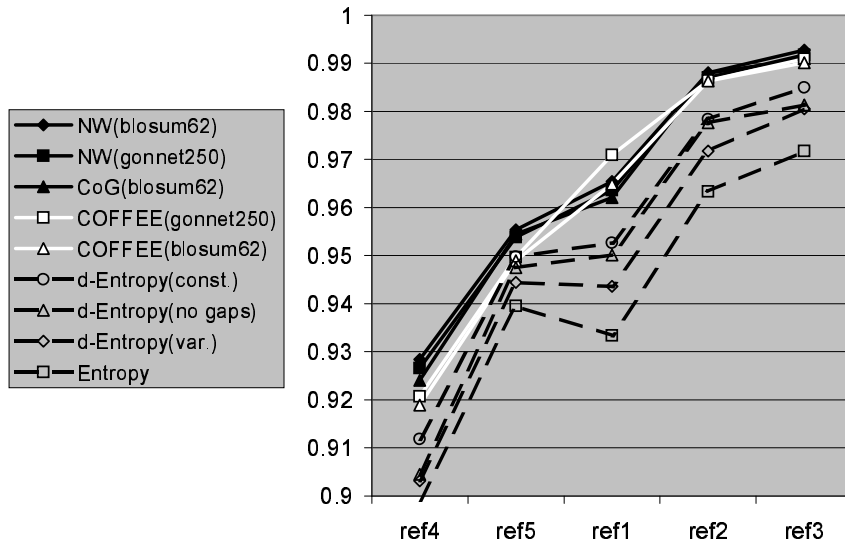
(a) The quality of the 9 tested functions according to the three quality measures  $Q_r(F, \Omega)$ ,  $QS_r(F, \Omega)$  and  $QP_r(F, \Omega)$ , (b) The best 5 functions with higher scores are given in better resolution.

## 7.1 Comparisons over distinct groups of structural alignments

Intuitively, the characteristics of the structural alignments should affect the performance of scoring functions in different ways. This was demonstrated when we tested the scoring functions separately for each BALiBASE reference set. Figure 7 includes the results of these tests.

As expected, the results for different reference sets are quite different – while the largest value for reference set 4 is smaller than 0.93, the smallest value for reference set 3 is larger than 0.97. In addition, the overlap of confidence intervals is significantly reduced, which means these results are more reliable. Some general features observed in figure 6 are not changed. The entropy-based functions are still outperformed by all the others (with one exception in reference set 5), and the ranking of these functions remain the same as before. The rest of the functions can now be divided into two groups: COFFEE vs. NW and CoG. In all but reference set 1, the latter group (NW and

Figure 7:  $QS_r$  for BALiBASE reference sets

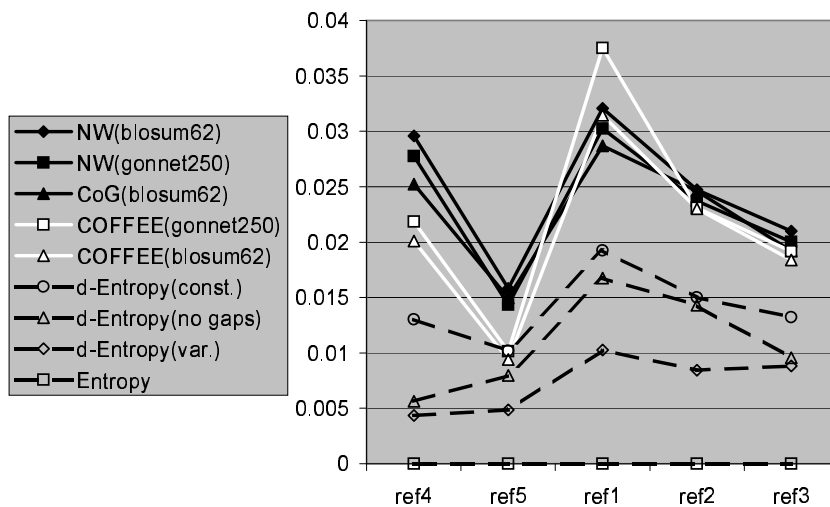


$QS_r(F, \Omega_i)$  is given in column “ref $i$ ” for each of the tested scoring functions. The functions are grouped by line color and shape according to their performance on the different reference sets.

CoG) has better results than the two COFFEE functions.

The internal differences among the two distinct groups are very small. In order to improve the visibility of these differences, and since we are mostly interested in the ranking of the functions, in each reference set the  $QS$  results were adjusted to reveal this ranking. The adjustment was made for each reference set separately, by subtracting the minimal value obtained by any of the functions in that reference set. In this particular example, this means subtracting  $QS_r(Entropy, \Omega_i)$  from each  $QS_r(F, \Omega_i)$ . The adjusted  $QS$  values are given in figure 8.

Figure 8: Adjusted  $QS_r$  for BALiBASE reference sets

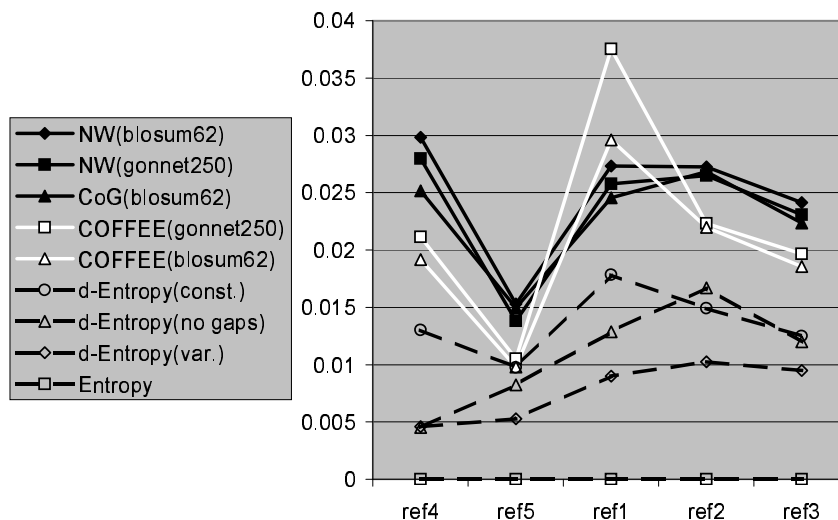


In this more detailed figure it is possible to observe that in the COFFEE group the GONNET250 precedes BLOSUM62 in all reference sets. Since COFFEE depends on the quality of the library

of pairwise alignments, it is expected that the GONNET250 matrix which was already found to perform better in pairwise alignments [25] would be better. On the other hand, in the NW functions, BLOSUM62 is slightly better. This could be due to the origin of the two matrices – while the GONNET250 matrix was derived from pairs of aligned sequences, BLOSUM62 was derived from non-gapped blocks of multiple alignments.

In figure 9 the  $Q$  values are adjusted in the same way as  $QS$  (i.e. by subtracting the minimal value in each reference set). It is evident that there is a clear order between the two groups, when the NW/CoG group performs better than the COFFEE group with the exception of reference set 1 (in which the order is reversed). The main difference between the alignments in reference set 1 in comparison to the other reference sets is that in this set the ratio of identity between every pair of sequences in the alignment is kept above certain threshold. In other reference sets, in every alignment there are pairs of sequences which are remotely related, so that the NW results for these pairs are less informative. Since COFFEE still takes these sequence pairs in account, this could cause the decrease in the quality COFFEE on these alignments. This hypothesis should be tested by using weighted COFFEE, in which closely related sequence pairs have larger influence on the score than remotely related sequence pairs [13].

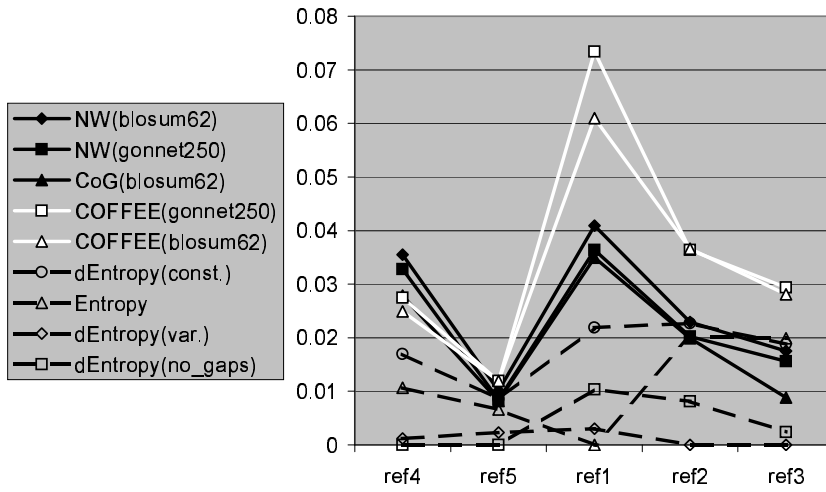
Figure 9: Adjusted  $Q_r$  for BALiBASE reference sets



Since when using the Pearson correlation coefficient some of the confidence intervals overlap, we tried other correlation tests, and  $Q_\gamma(F, R)$  was calculated for every reference alignment  $R$ . While  $Q_r$  values are in the range  $[0.9, 0.99]$  (figure 7), the  $Q_\gamma$  values are smaller and more dispersed – in the range  $[0.75, 0.95]$  (data not shown). However, much of the observations made on the basis of  $Q_r$  are consistent with the results for  $Q_\gamma$ . In figure 10, which includes the adjusted  $Q_\gamma$  values, it can be seen that there are many similarities between the adjusted  $Q_\gamma$  values and the adjusted  $Q_r$  values (figure 9).

The main difference between  $Q_r$  and  $Q_\gamma$  is that the NW and CoG values are relatively lower for  $Q_\gamma$ , so that in most cases they are worse than the COFFEE values, and in some cases they are equal to the best entropy based functions. Another important difference is that in contrast with the situation in  $Q_r$  where there is a clear order between the entropy-based functions, in  $Q_\gamma$  the order between these functions changes in different reference sets. We currently do not have an explanation to both dissimilarities. As we are not aware of accurate tools to compute confidence intervals for  $Q_\gamma$ , we estimated the significance of the ranking as follows: Each sample set was partitioned into five randomly selected sets of equal size,  $Q_\gamma$  was calculated separately for each set, and the results were compared. The maximal change in the values given in figure 10 was approximately 0.001, which in most cases is smaller than the difference in the rankings of the corresponding functions, and the ranking of the functions was not changed. This seems to

Figure 10: Adjusted  $Q_\gamma$  for BALiBASE reference sets

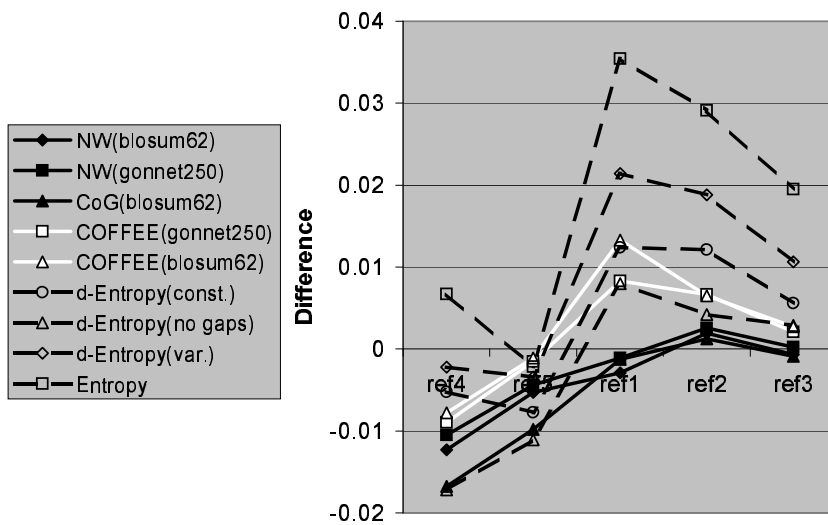


indicates that the ranking is reliable.

## 7.2 The effect of adding gap columns

As mentioned in section 3, in order to test whether the restriction on the number of columns in the alignment is too stringent and causes bias in the results, the whole evaluation process was repeated with the addition of gap columns to the reference alignment. A gap column was added before and after the alignment, and also following each tenth column inside the alignment. The resulting alignment was used as the starting point of the random walks.  $QS_r$  and  $Q_r$  with the additional gap columns will be denoted  $QS'_r$  and  $Q'_r$ , respectively. The change in  $QS_r$  (i.e.  $QS'_r - QS_r$ ) is given in figure 11.

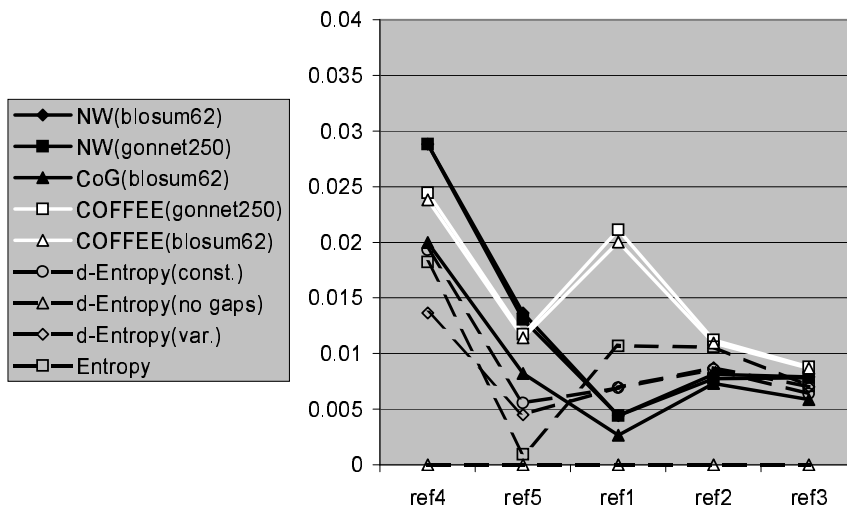
Figure 11:  $QS'_r - QS_r$  for BALiBASE reference sets



For each function  $F$ ,  $QS'_r(F, \Omega_i)$  and  $QS_r(F, \Omega_i)$  were calculated (with and without additional gap columns, respectively).  $QS'_r(F, \Omega_i) - QS_r(F, \Omega_i)$  is given in column "ref $i$ " for each of the tested scoring functions.

The addition of gap columns has different affect on the tested scoring functions. This is best depicted for reference sets 1, 2 and 3, in which the reference alignments do not contain many gaps. In these sets  $QS_r$  for the NW/CoG group of functions is almost unchanged, while it is significantly improved for the other functions, in particular Shannon's Entropy and dEntropy with varying tree. On the other hand, for reference sets 4 and 5 in which the reference alignments have many gaps, this difference is reduced, and  $QS_r$  is smaller (with a single exception). The immediate conclusion from these results is that if the percentage of gaps in the reference alignment is too low, the measurement of the scoring functions quality in our method is distorted. The addition of gap columns is therefore recommended.

Figure 12: Adjusted  $QS'_r$  for BALiBASE reference sets

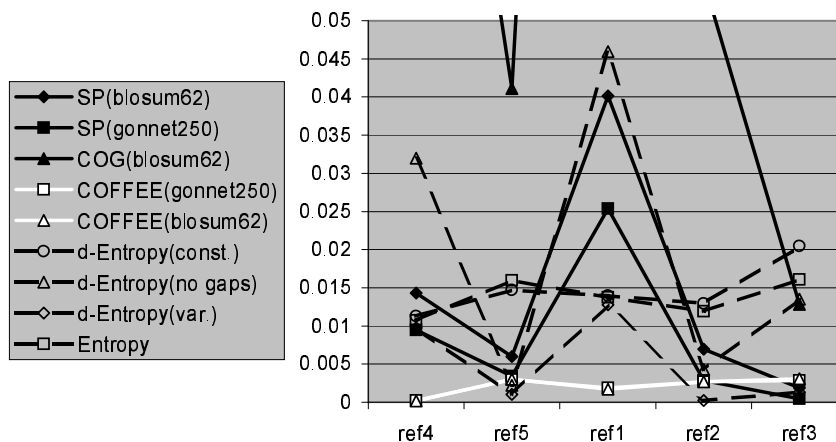


Other observations can be drawn from adjusted  $QS'_r$  which is given in figure 12, and comparing it to adjusted  $QS_r$  (figure 8):

- In  $QS_r$ , the NW functions were better than the COFFEE functions in all reference sets except reference set 1. In  $QS'_r$  this situation is retained only for reference sets 4 and 5, while in all other reference sets both COFFEE functions are better than NW.
- CoG which was tightly coupled with the NW functions previously, has lower  $QS'_r$  than these functions, in all of the reference sets.
- There are two differences related to the entropy based functions. The first is that the Shannon's entropy is not the worse function – it is replaced by dEntropy without gaps, and in some of the reference sets it is the best entropy based function. The second difference is that in reference sets 1, 2 and 3 the difference between the entropy based functions and the other functions is reduced, and three of the four entropy based functions perform better than NW and CoG.

The last important observation that can be made regarding the addition of gap columns is that it raises the importance of gap penalties. While without additional gap columns the maximal improvement achieved by introducing gap penalties for any BALiBASE reference set ( $QS_r - Q_r$ ) was less than 0.007 (data not shown), it can be seen in figure 13 that the addition of gap columns changes the picture, as the  $QS'_r - Q'_r$  values (the improvement achieved by introducing optimal gap penalties) are in many cases much higher than 0.007, especially for the CoG function which reaches a value of 0.264 in reference set 1.

The practical conclusion from figure 13 is that we should incorporate gap penalties in the scoring function we use in order to get results which are near optimal. However, there is an exception to this observation, since the  $QS'_r - Q'_r$  values of COFFEE functions are below 0.004 (with maximal

Figure 13:  $QS'_r - Q'_r$  for BALiBASE reference sets

value for a single alignment below 0.0178), similar to the  $QS_r - Q_r$  of these functions. As it seems, although the COFFEE scores were multiplied by 10,000 in order to enable an influence of the gap penalties, on average this influence is not significant. We conclude that almost all of the information that can be obtained from the gaps in the alignment is incorporated in COFFEE, and therefore COFFEE can be used directly, without the addition of gap penalties. It should be noted that the norMD function [23] includes a normalization step which is designed to reduce the influence of gaps on the overall score. As expected, norMD is less sensitive to the addition of gap penalties than most functions, but it is more sensitive than COFFEE.

## 8 Conclusion

The goal of this work was to define an evaluation scheme for MSA scoring function using structural alignments. The main concept in the suggested scheme is the application of a correlation test between the tested scoring function and a structural score. In order to perform such correlation test, a sample set is selected through a stochastic process. This process is designed to select samples which are relatively similar to the structural alignment, including alignments with structural score similar to that of the output of existing MSA algorithms.

After performing a series of experiments using the proposed scheme, there are several important conclusions:

- As in the pairwise alignment problem, for many of the MSA scoring functions tested the use of gap opening and extension penalties causes significant improvement to the scoring function quality, and when the gaps are counted for single sequences the improvement is larger than with pairs of sequences. However, the COFFEE function is less influenced by gap penalties, which makes it more attractive in many of the possible applications of scoring functions.
- Some alignment characteristics such as the minimal similarity between sequences and the gap percentage influence the results, in both absolute values and the ranking of different scoring functions.

Besides conducting more experiments in the current settings, there are several improvements which can be introduced to the scheme. In [17] it is suggested that the BALiBASE score does not utilize all the information in the structural alignment, and new scores are being developed which attempt to include more information. It should be tested if these scores improve the evaluation scheme. In the calculation of  $TT_R$  a single parameter is used in all reference sets, and other ways to determine  $TT_R$  should be tested, including differentiation between reference set or alignments. There are a few additional parameters in the scheme which can be varied, such as the number of

random walks and others. The effects of changing these parameters on the results should also be tested.

The last and most needed expansion to the current scheme is to use additional databases of structural alignments, such as OXBench [14] and others. This should be combined with an option to test only parts of the alignment, which corresponds to the local alignment problem, in which the goal is to find a partial alignment which does not necessarily include the entire sequences. This addition can be especially beneficial when developing programs such as NorMD [23], which uses a “sliding window” to evaluate the quality of different regions in the alignment.

The functions tested in this work are only the tip of the iceberg, and many additional scoring functions can be proposed – either modifications of the tested functions or functions with different theoretical basis. The next step in the search of the best functions is a comprehensive survey of these functions, followed by a comparative study.

## Acknowledgements

The authors wish to thank Paul Feigin and Ayala Cohen for their assistance with statistical aspects of this work.

## References

- [1] Do C.B., Mahabhashyam M.S.P., Brudno M. and Batzoglou S., **ProbCons: Probabilistic consistency-based multiple sequence alignment**, *Genome Research* 15 (2005) 330–340.
- [2] Felsenstein J., **Inferring Phylogenies**, *Sinauer Associates*, Sunderland, Mass. (2004).
- [3] Gonnet G.H., Cohen M.A. and Benner S.A., **Exhaustive matching of the entire protein sequence database**, *Science* 256 (1992) 1444–1445.
- [4] Gotoh O., **Significant improvement in accuracy of multiple protein sequence alignments by iterative refinement as assessed by reference to structural alignments**, *J. Mol. Biol.* 264 (1996) 823–838.
- [5] Henikoff S. and Henikoff J.G., **Amino acid substitution matrices from protein blocks**, *Proc. Nat. Acad. Sci.* 89 (1992) 10915–10919.
- [6] Henikoff S. and Henikoff J.G., **Performance evaluation of amino acid substitution matrices**, *Proteins* 17 (1993) 49–61.
- [7] Hollander M. and Wolfe D.A., **Nonparametric statistical methods**, 2nd edition, *Wiley Series in Probability and Statistics*, New York, NY (1999).
- [8] HyperStat Online Contents, **Confidence interval on Pearson’s correlation**, <http://davidmlane.com/hyperstat/B8544.html>.
- [9] Johnson, M.S. and Overington, J.P., **A structural basis for sequence comparisons. An evaluation of scoring methodologies**, *J. Mol. Biol.* 233 (1993) 716–738.
- [10] Lassman T. and Sonnhammer E.L.L., **Automatic assessment of alignment quality**, *Nucleic Acids Res.* 33 (2005) 7120–7128.
- [11] McClure, M.A., Vasi, T.K. and Fitch, W.M., **Comparative analysis of multiple protein-sequence alignment methods**, *Mol. Biol. Evol.* 2 (1994) 572–592.
- [12] Needleman S.B. and Wunch C.D., **A general method applicable to the search for similarities in the amino acid sequence of two proteins**, *J. Mol. Biol.* 48 (1970) 443–453.
- [13] Notredame C., Holm L. and Higgins D.G., **COFFEE: an objective function for multiple sequence alignments**, *Bioinformatics* 14(5) (1998) 407–422.

- [14] Raghava G., Searle S.M., Audley P.C., Barber J.D. and Barton G.J., **OXBench: A benchmark for evaluation of protein multiple sequence alignment accuracy**, *BMC Bioinformatics* 4:47 (2003).
- [15] Sattath S. and Margalit H., **Measuring Amino Acid Conservation by Distance Sensitive Entropy**, unpublished.
- [16] Shannon C.E. **A mathematical theory of communication**, *The Bell System Technical Journal* 27 (1948) 379–423, 623–656.
- [17] Schwartz A.S., Myers E.W. and Pachter L., **Alignment Metric Accuracy**, <http://arxiv.org/abs/q-bio.QM/0510052>.
- [18] Siegel S. and Castellan N.J., **Nonparametric statistics for the behavioral sciences**, *McGraw-Hill*, New York, NY (1998).
- [19] Thompson J.D., Gibson T.J., Plewniak F., Jeanmougin F. and Higgins H.D., **The CLUSTALX windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools**, *Nucleic Acids Res.* 25 (1997) 4876–4882.
- [20] Thompson J.D., Higgins H.D. and Gibson T.J., **CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice**, *Nucleic Acids Res.* 22 (1994) 4673–4680.
- [21] Thompson J.D., Plewniak F. and Poch O., **BAlIbASE: a benchmark alignment database for the evaluation of multiple alignment programs**, *Bioinformatics* 1 (1999) 87–88.
- [22] Thompson J.D., Plewniak F. and Poch O., **A comprehensive comparison of multiple sequence alignment programs**, *Nucleic Acids Res.* 27(13) (1999) 2682–90.
- [23] Thompson J.D., Plewniak F., Ripp R., Thierry J.C. and Poch O., **Towards a reliable objective function for multiple sequence alignments**, *J. Mol. Biol.* 314 (2001) 937–951.
- [24] Valdar W.S.J., **Scoring Residue Conservation**, *PROTEINS: Structure, Function, and Genetics* 48 (2002) 227–244.
- [25] Vogt G., Etzold T. and Argos P., **An assessment of amino acid exchange matrices in aligning protein sequences: the twilight zone revisited**, *J. Mol. Biol.* 249 (1995) 816–831.
- [26] Wang, L. and Jiang T., **On the complexity of multiple sequence alignment**, *J. Comput. Biol.* 1 (1994) 337–348.