Measuring the distance between multiple sequence alignments

Benjamin P. Blackburne and Simon Whelan*

Computational and Evolutionary Biology, Faculty of Life Sciences, University of Manchester, Manchester M13 9PT

ABSTRACT

Motivation: Multiple sequence alignment (MSA) is a core method in bioinformatics. The accuracy of such alignments may influence the success of downstream analyses such as phylogenetic inference, protein structure prediction, and functional prediction. The importance of MSA has lead to the proliferation of MSA methods, with different objective functions and heuristics to search for the optimal MSA. Different methods of inferring MSAs produce different results in all but the most trivial cases. By measuring the differences between inferred alignments, we may be able to develop an understanding of how these differences relate to the objective functions and heuristics used in MSA methods, and affect downstream analyses.

Results: We introduce four metrics to compare MSAs, which include the position in a sequence where a gap occurs or the location on a phylogenetic tree where an insertion or deletion (indel) event occurs. We use both real and synthetic data to explore the information given by these metrics and demonstrate how the different metrics in combination can yield more information about MSA methods and the differences between them.

Availability: Metal is a free software implementation of these metrics in Haskell. Source and binaries for Windows, Linux and Mac OS X are available from http://kumiho.smith.man.ac.uk/whelan/software/metal/.

Contact: simon.whelan@manchester.ac.uk

Received on July 19, 2011; revised on December 06, 2011; accepted on December 19, 2011

1 INTRODUCTION

Many methods used in bioinformatics require one or more accurate multiple sequence alignments (MSAs) as input. Each MSA arranges a set of homologous amino acid or nucleotide sequences in a matrix, where each column of the matrix corresponds to a set of characters that are homologous, functionally related or superposable in a protein structure [Edgar and Barnotka 2005]. These definitions may coincide when the sequences are closely related, but may conflict as the sequences diverge. Methods of phylogenetic tree reconstruction [Felsenstein 1981], structure prediction [Arnold et al. 2008], functional annotation [Finn 1998] and the creation of profile hidden Markov models for database searching [Krogh 2000] all depend on an alignment, and the performance of these methods depends inevitably on the accuracy of these alignments. A variety of MSA methods have been developed over the past two decades that are still in active use [Notredame 2008], including methods that co-estimate trees and alignment [Gusenman et al. 2001; Lim et al. 2009; Redelings and Suchard 2005]. These algorithms vary in their objective function and the heuristic used to find the best MSA.

The performance of MSA methods is frequently assessed by their ability to recover a reference alignment (e.g. Kishikihara et al. 2009), which may be produced from biological information, such as a conserved structure [Thompson et al. 2005]. The output of each method can then be scored against the reference alignment, either by the fraction of residue pairs in the reference alignment that are correctly identified by a given method, known as the Sum-of-Pairs (SP) score, or by the Total Column (TC) score, which describes the fraction of reference columns identified [Thompson et al. 2005].

Several studies have noted the effect of sequence alignment algorithm on, for example, phylogenetic tree topology [Cantarel et al. 2004; Hall et al. 2001; Morrison and Eddy 1994; Ogden and Rosenberg, 2006; Wu and Wu et al. 2011; Wu and Wu et al. 2009]. Many of these studies used simulated data and the parameter of interest was generally assessed against alignment quality, measured as TC or SP score against the true alignment. Neither the TC nor the SP score are true metrics because they violate the principles of symmetry and the triangle inequality (see below). Producing a valid metric would allow alignment to be projected as points in an alignment space, enabling the comparison of distances between different MSA methods or across different datasets. When these comparisons include a reference alignment, the metric would give some indication of the similarity of alignment methods as well as an accuracy score. Alternatively, such a metric could be used to characterize alignment uncertainty. Previous work on uncertainty has relied on annotating a ‘best’ alignment with uncertainty information based on the proportion of base pairs that are present when the sequences are reversed before alignment [Lundan and Grant 2007; Wu 2011], or when the guide tree is varied [Pettigrew et al. 2001]. A true metric could also provide direct comparison of equally or similarly high scoring paths through the dynamic programming matrix produced during MSA, allowing one to identify regions of high uncertainty in the final alignment.

In this study, we derive four true metrics for the comparison of MSAs: (i) a simple correction to the SP score; (ii) a metric that incorporates raw gap information; (iii) a metric that includes the position where gaps occur in a sequence; and (iv) a metric that includes the position where insertion/deletion (indel) events occur both in a sequence and on a phylogenetic tree. We proceed to demonstrate that the SP and TP score are not metrics and cannot be used to investigate many questions about how MSA methods perform. We then show the usefulness of our metrics on MSAs produced by a selection of methods on real data extracted from BAliBASE [Thompson et al. 2008], and simulated data produced using INDELible [Pettigrew and Yand 2009]. Through these analyses, we also demonstrate that the combined use of our metrics can identify similarities and differences between
MSA methods and characterize the rate of decline in alignment performance over increasing evolutionary distances.

2 METHODS

2.1 General definitions
Consider a set of n sequences \( S = \{S_1, S_2, \ldots, S_n\} \) where the \( j \)-th character in sequence \( i \) is denoted \( S_{ij} \) and the sum of the sequence lengths is given by \( c = \sum_{i=1}^{n} |S_i| \). Also consider a phylogenetic tree, \( T \), that describes the evolutionary relatedness of the sequences in \( S \) by a series of edges, \( e = \{e_1, e_2, \ldots, e_{c-1}\} \) where \( e_k \) describes an edge in the tree that splits \( S \) into two non-empty sets. Distances between MSAs should compare where in the MSAs each observable character, \( S_{ij} \), is placed in relation to the characters in the other sequences. We begin by assigning \( S_{ij} \) to a homology set, which contains the characters in the other sequences that share a common ancestor with \( S_{ij} \). Where there is no such relationship due to an indel, the set may include a character representing the gap state. When assigning such homology sets, we may also wish to include additional information in the labelling of the gap states to alter the information given by the distance. First, we can include information about the placement of the gap characters in their respective sequences. Secondly, we can include information about the likely edge that corresponds to the location of the indel event on the tree that resulted in the gap.

2.2 Recoding the insertion and deletion history of an alignment
The next step in computing a distance between alignments is to recode the alignment so that all the characters and gaps can be correctly compared. The transition from \( S \) to an alignment requires the placement of gap characters, which represent all the indels that have occurred during the history of the sequences. We assume no direction to evolution and, therefore, cannot differentiate between insertions or deletions. To incorporate information from indels, we recode gap characters in our alignments in one of four ways (see Fig. 1 for examples):

1. **SSP**; the ‘Symmetrized SP’ recoding ignores gaps and treats them as blanks in an alignment. The name reflects the similarity to the existing SP method for comparing alignments.
2. **seq**; this recoding provides a simple record of gap information and treats all gaps in a sequence equally. Each gap is simply recoded as \( G_i \), indicating it occurred in sequence \( i \).
3. **pos**; our third recoding incorporates the positional information about where a gap occurs in a sequence, but not the temporal (phylogenetic) location of the indel that produced the gap. Each gap is labelled as \( G_j \) where \( j \) is the location of the real character to the left. In Figure B, both gaps occurring after the first character in Sequence 3 are labelled as \( G_2 \). Note any gap occurring before the first character would be labelled \( G_0 \).
4. **evol**; our final recoding includes all the information of pos recoding, and also incorporates where the indel event leading to that gap occurs in a phylogenetic tree. Each column in an alignment is considered independently and the indel history inferred as the most parsimonious set of events under Dollo parsimony. Dollo parsimony enforces that the history of each column is restricted to a maximum of one insertion and any number of deletion events, with the mapping of indel events to edges the same regardless of the tree rooting. Gap characters in the column are then recoded using the nomenclature \( G_i(k) \), where \( k \) is an index used to label edges in the tree.

![Fig. 1. Labelling of sequence alignments in order to determine distances by different metrics. Gap states are given in bold to highlight the difference between the four methods.](https://academic.oup.com/bioinformatics/article-abstract/28/4/495/212883/)

![Fig. 2. Two alternative representations of the same alignment. Before our recoding procedure, the columns of representation B would be re-ordered to give representation A.](https://academic.oup.com/bioinformatics/article-abstract/28/4/495/212883/)

2.3 Enforcing a unique representation of an alignment
When comparing MSAs, the first step is to ensure that any given assignment of gaps has a unique representation in alignment space. When gaps are considered as independent events, the order of some columns in an alignment may be arbitrary. For example, the last two columns in Figure B could be placed in any order because they contain no overlapping bases. To enforce a unique alignment representation, we reorder sequences alphabetically by their names. We then sort the non-overlapping columns so that the leftmost column contains a gap in a higher row than the column to the right.

2.4 Including evolutionary information
The labelling step for the evol recoding requires a tree in order to label each gap by the inferred indel event that caused it, assuming that each position in a multi-site gap can be considered independently. Although this approach is not ideal from a biological perspective, it does allow efficient and simple computation of where indels occur on the tree. Allowing gaps to span multiple columns makes identifying their placement on the tree more difficult, because there may be many equally parsimonious solutions for gap placement (Simmons and Ochoterena, 2000), each of which may induce a different homology assignment to a site and result in a different distance between alignments. Our approach may be viewed as similar to using a linear gap penalty when identifying where an indel occurs.

2.5 Comparing homology sets
Once a pair of alignments are recoded appropriately the process of calculating the distances between them can begin. The first step is to calculate the site-wise homology set for each alignment under each metric, so for alignment \( A \) and metric \( X \in\{SSP, seq, pos, evol\} \) we have \( H_X(A) = \{H_X(A)_j\} \). To create the homology set \( H_X(A)_j \), one identifies the column in alignment \( A \) with character \( S_{ij} \) and store all the other characters in that column, including gaps if they are labelled. Note that gaps do not have a homology set of their
own because they do not represent observable data and their presence varies between alignments. For example, the \( \text{evol} \) recoding from Figure 2b shows the homology set for sequence character \( S_2 \) is \( H_{\text{evol}}(A_2) = \{ \{2\}, \{4\} \} \). We define the \( d_{\text{hom}} \) metric as the Jaccard distance on the homology sets \( H_{\text{hom}} \). Previous research has shown the Jaccard distance to be a valid metric. Let \( d_{\text{SSP}} \) be

\[
d_{\text{SSP}}(A,B) = 1 - \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} H_{i}^{(1)} \cap H_{j}^{(2)}}{\sum_{i=1}^{n} \sum_{j=1}^{m} H_{i}^{(1)} \cup H_{j}^{(2)}}
\]

We call this measure of distance between alignments our Symmetrized SP (SSP) metric, reflecting its inspiration from the original SP score. This metric has a mixture of desirable and undesirable properties. It has a direct link to the information used by most methods of sequence analysis (homology recoding every homology set to the metric is less clear). By explicitly encoding gap characters, we can extend the metric to include other information that may be considered important.

For the remaining three metrics, we assume them such that they take advantage of the fact that the homology set sizes remain constant for any given alignment. In Figure 2, one can see that for \( \text{seq}, \text{pos} \), and \( \text{evol} \) recoding every homology set is of size \( n - 1 \), in contrast to SSP where the size of homology sets varies between alignments. We take advantage of this consistency to use the following to compute the distance metric \( d_{\text{X}}(A,B) \) for \( X \in \{\text{seq}, \text{pos}, \text{evol}\} \):

\[
d_{\text{X}}(A,B) = \frac{1}{\sum_{i=1}^{n} \sum_{j=1}^{m} d_{\text{X}}(i,j)}
\]

The numerator of this equation is the symmetric difference, which in our case is equivalent to the Hamming distance, as each member of \( H_{\text{X}}(A)^{j} \) has a corresponding member in \( H_{\text{X}}(B)^{j} \). Previous research has shown the Hamming distance to be a true metric. We define the distance matrices using the heatmap.2 package from the gplots R package.

2.7 Alignment programs examined

To demonstrate the performance and usefulness of our four metrics, we use them to compare the alignments produced from a range of popular alignment programs with their default options on two different datasets. We consider the progressive aligners ClustalW (Larkin et al., 2007), Muscle (Edgar, 2004), T-Coffee (Notredame, 2000), ProbCons (Do et al., 2005), and the L–INS–I algorithm of MAFFT; and the phylogenetically aware aligner Prank, with the recommended ‘-V’ option (Gouy and Guindon, 2008). Although other non-default parameters have the potential to improve the quality of alignments, our aim is not to judge alignment quality but to show how the metrics presented here reveal differences in the alignments.

We examine two datasets, one synthetic, and one real (described below). For each dataset examined, we align the sequences using each of the procedures detailed above. We also include the true or reference alignment for comparison. Given these alignments, we can calculate the all-against-all set of distances for each of the four metrics proposed here. We visualise mean distance matrices using the heatmap.2 package from the gplots R package.

2.7.1 Test data 1: BAliBASE

We examine alignment performance on three sections of the BAliBASE 3 database, which has been widely used to assess the performance of alignment algorithms at recovering reference alignments derived from protein structure information. We use sections RV12 (medium-to-divergent sequences with 20–40% identity), RV30 (subfamilies) and RV50 (internal insertions), with 20 alignments randomly chosen from each section for subsequent analysis. Results are also presented for the truncated versions of the same sequences, containing only the homologous regions. Alignments contain between 4 and 142 sequences (mean of 34.9).

2.7.2 Test data 2: synthetic data

Simulated data provides a useful test of alignment where all properties of the data are known, at the cost of not reflecting the complexity of real data. In this study, we use synthetic data to assess the effect of evolutionary distance on alignment. We generate data under a balanced tree topology with eight leaves, where all 14 branches are of equal length and the total tree branch length is one of \( 0, 1, 2, 3, 4, 5, 6, 7 \). We also generate a second dataset with an unbalanced tree where each internal branch has equal length and the overall tree length is scaled the same as the corresponding balanced tree. All trees used are clock-like. INDELible v1.03 (Fletcher and Yang, 2009) was used to simulate data under the WAG substitution matrix, with an indel rate parameter of 0.1 and a power-law indel length distribution with parameter \( \alpha = 1.7 \). The root sequence has length 1000, and 20 replicates were made of each run.

3 RESULTS

3.1 The problem with existing scores

A non-negative function, \( d(x,y) \), is a metric if it obeys the following conditions:

1. \( d(x,y) = 0 \) if and only if \( x = y \) (identity of indiscernibles).
2. \( d(x,y) = d(y,x) \) (symmetry).
3. \( d(x,z) \leq d(x,y) + d(y,z) \) (triangle inequality).

497
Fig. 3. The process of analysing the alignment of two toy sequences with three sequence aligners, using the SP score. See Section 2 for discussion. (A) Perform sequence alignments; (B) calculate SP differences; (C) choose one reference and compare; (D) use mean difference as pseudo-distances.

We demonstrate that existing methods for comparing alignments are not metrics, starting with the widely used SP similarity. Thompson et al. (2002) recorded a dissimilarity, $d_{SP}$. Using our labelling definitions (Fig. 1), the dissimilarity $d_{SP}(A,B)$ between a test alignment $A$ and a reference alignment $B$ is:

$$d_{SP}(A,B) = 1 - \frac{\sum_i |H_{SSP}(A)_i| \cap |H_{SSP}(B)_i|}{\sum_i |H_{SSP}(B)_i|}$$

This dissimilarity score is not a true metric because it fails to satisfy the conditions of symmetry and the identity of indiscernibles. The asymmetry occurs because the denominator may be different for SP (and TC) score using one of the alignments as a reference across the three alignments, but because we need to normalize the programs. In Figure 3B, we calculate SP differences between the three alignments, but because we need to normalize the SP (and TC) score using one of the alignments as a reference.

Fig. 4. Distance matrix computed using $d_{evol}$. The upper part of the matrix gives the (mean) distance between two alignments over all of BALiBASE. RV12, RV30 and RV50, scaled as percentages. The lower part gives the same information but for the homologous regions only. Darker shades of grey indicate increasing distance.

alignment, the distance (e.g.) $A \rightarrow B$ is different to the distance $B \rightarrow A$, demonstrating that the SP (and TC) violates the symmetry requirement for a distance.

To demonstrate how the choice of reference alignment affects these scores, Figure 3B demonstrates the different conclusions drawn when each MSA is taken to be the reference. If we take $A$ to be the reference, we infer that $B$ and $C$ are equally similar to $A$, perhaps concluding that $B$ and $C$ are equally good aligners. However, if we take $B$ to be the reference alignment, we find $A$ and $B$ produce the same alignment, whereas $C$ is maximally different. In this case, we would conclude $B$ is an excellent method and $C$ is a poor method.

One may try to solve these problems by taking the mean of the score $A \rightarrow B$ and $B \rightarrow A$. This approach enforces symmetry, but the resultant distances do not obey the triangle inequality. In Figure 3B, we demonstrate that averaged distances result in a triangle between $A$, $B$ and $C$ that cannot exist in Euclidean space. These theoretical failings expose a major limitation of existing metrics: although they can measure a similarity to a reference alignment, they cannot be used for all-against-all comparisons, with or without a reference. This limitation is illustrated in Figure 4 where we show that existing metrics prevent any meaningful comparison between a set of toy MSAs. In Figure 4B, we align two short sequences using three alignment programs. In Figure 4B, we calculate SP differences between the three alignments, but because we need to normalize the SP (and TC) score using one of the alignments as a reference.
To investigate the performance of our metrics on alignments with unbalanced trees are similar. For a subset of methods for accuracy, we examine the average distance from the true alignment generated by INDELible to its alignment model closely resembles that used by INDELible to simulate the data. Prank performs well. The performance of Prank may be expected because it has a noticeable tendency to have high distances to other aligners (43.5–49.4%). Prank and ClustalW, for example, show the greatest dissimilarity (49.4%), whereas T-Coffee and ProbCons give the most similar alignments (26.6%). Prank has a noticeable tendency to have high distances to other aligners (43.5–49.4%).

These results suggest that although the different MSA methods are producing alignments that are roughly of equal similarity to the BAIBASE alignment, the way they achieve these alignments can be quite different. BAIBASE also provides alignments where only regions of identified homology are included. Results for these datasets are plotted below the diagonal of Figure 4. Trends are similar for both the full sequences and the homologous regions, although absolute distances are larger for the full sequences.

### 3.3 Comparing alignment methods using data simulated by INDELible

To investigate the performance of our metrics on alignments with a known origin and history, we apply the metrics to data simulated using INDELible on a balanced and an unbalanced tree, each with tree length of 1.4. The mean values of $d_{seq}$ are plotted in a matrix in Figure 5 across the balanced (upper matrix) and unbalanced (lower matrix) datasets. These figures show that the differences between methods are more pronounced in simulated data relative to real data, with ClustalW performing noticeably worse than other methods. In both examples, the phylogenetically aware aligner, Prank, has the best performance, although MAFFT L-INS-i also performs well. The performance of Prank may be expected because its alignment model closely resembles that used by INDELible to generate the data.

To explore the effect of increasing evolutionary distance on accuracy, we examine the average distance from the true alignment for a subset of methods for $d_{seq}$ and $d_{vol}$ under a range of evolutionary divergences on balanced trees. The trends for unbalanced trees are similar. For $d_{seq}$, we find that as the evolutionary divergence increases, the distance between alignments inferred by the MSA methods also begins to increase, but it reaches a maximum distance when the tree length is ~4, and then plateaus or declines (Fig 5b). In contrast, for $d_{vol}$, the distance between inferred MSAs and the true MSA steadily increases as evolutionary divergence increases (Fig 5c). Curves for $d_{pos}$ and $d_{gap}$ are similar to $d_{vol}$, reaching 1.0 at or after a tree length of 7. The differences between $d_{seq}$ and $d_{vol}$ in Figure 5 are due to the way they treat gaps in the divergent sequences. For $d_{seq}$, the decrease in alignment distance for divergent sequences occurs because the majority of amino acids are aligning with gap characters, which cannot be distinguished from one another. On the other hand, $d_{vol}$ includes evolutionary and positional information about gaps, which allows the metric to correctly identify falsely inferred indel events.

Equation 1 makes it possible to calculate how differently a pair of MSAs treat a single sequence, potentially highlighting sequences that are aligned particularly inconsistently between two methods. We investigate this approach by examining the performance of the different MSA methods on simple balanced and unbalanced tree topologies. We expect that each sequence on the balanced tree will have a similar distance under the same method, but for the unbalanced tree the most distantly related sequences will be harder to align than the more closely related sequences. In Figure 6, we use simulation to investigate the distance from the true alignment of individual sequences for equal length balanced and unbalanced trees using the $d_{vol}$ metric. The results match our expectations. For balanced trees, all leaf nodes are topologically identical, and the observed distances between sequence are broadly similar. For the unbalanced tree, we observe it is harder to align sequences from long terminal branches than those from short terminal branches.

### 3.4 Relationship between metrics

The structure of our metrics enforces the relationship $d_{seq} \preceq d_{vol} \preceq d_{gap}$ with the lack of gap information in $d_{gap}$ making it somewhat different from the other three metrics. Our results demonstrate that incorporating positional and evolutionary information can provide useful information to a bioinformatician. BAIBASE, example BB12038, contains one sequence that is much longer than the others,
Fig. 7. The effect of tree topology on sequence-wise distances from the true alignment. Sequence-wise distances ($d_{\text{evol}}$) are similar (within a margin of error) on the balanced tree, but increase with terminal branch length for the unbalanced trees.

Fig. 8. Distance matrix computed using $d_{\text{evol}}$, as Figure 4 but for BALIBASE alignments BB12038 (lower) and BB12020 (upper).

with a length of 2314 aa relative to the next longest of 437 aa. The extreme nature of this alignment makes it a useful case study. Results comparing alignment methods on BB12038 are plotted in Figure 8 below the diagonal. For comparison, scores for a more typical alignment are plotted above the diagonal. By considering each of the four metrics in turn, using the distance from the reference sequence to ClustalW and Prank, we can evaluate the information encoded by each metric. In the case of $d_{\text{SSP}}$, the distance between ClustalW and the reference is 0.37, so of all the base-base pairs across the two alignments, 63% of them are present in both. For Prank, the distance is larger (0.44). Substantially lower distances are seen under $d_{\text{seq}}$ (Prank = 0.19 and ClustalW = 0.14). The $d_{\text{SSP}}$ distance is not nested in the other distances, so care should be taken when comparing the other distances with $d_{\text{SSP}}$. In this case, we interpret the large decrease as evidence that the alignments are quite gappy and those locations aligned to gaps in the reference alignment also tend to be aligned with gaps by the MSA methods. Under $d_{\text{pos}}$, the distance increases only slightly for ClustalW (0.17) but Prank increases to 0.47, indicating that many of the gap locations differ between the Prank and reference alignments. A small further increase under $d_{\text{seq}}$ is observed (ClustalW = 0.18, Prank = 0.50). Prank frequently aligns bases to gaps located differently from the reference alignment, but only occasionally corresponding to different events in time.

To explore further the similarities and differences between our metrics, we investigate the distances they produce on BALIBASE RV12. We plot the distances between all pairs of aligners across all data from RV12 as scatterplots. Figure 9 shows comparisons between $d_{\text{SSP}}$ and our gap-aware metrics $d_{\text{seq}}$ (Fig. 9A) and $d_{\text{pos}}$ (Fig. 9B). The inclusion of non-specific gap information in $d_{\text{seq}}$ means that it tends to produce lower distances than $d_{\text{SSP}}$ as characters aligned with gaps in both alignments reduce the distance. This observation reinforces the result presented in Figure 6A, where $d_{\text{seq}}$ tends to decrease after a certain degree of divergence. The comparison between $d_{\text{SSP}}$ and $d_{\text{pos}}$ shows no such clean relationship, with $d_{\text{SSP}}$ sometimes being greater than or less than $d_{\text{pos}}$. This variation is because for $d_{\text{pos}}$ the location of the gap is important, so if two methods align gappy regions similarly $d_{\text{pos}}$ is lower than $d_{\text{SSP}}$, whereas for inconsistent gappy regions $d_{\text{pos}}$ is greater than $d_{\text{SSP}}$.

The substantial differences between $d_{\text{pos}}$ and $d_{\text{seq}}$ are also illustrated in Figure 8. In common with the $d_{\text{SSP}}$ and $d_{\text{seq}}$ comparison, these differences arise due to the treatment of gaps. The definitions enforce the relationship $d_{\text{pos}} \geq d_{\text{seq}}$, but sometimes $d_{\text{pos}}$ is substantially higher than $d_{\text{seq}}$ and we infer that such alignments are gappy with much variation in the base-gap homology assignments.
Measuring the distance between MSAs

Figure 9. Scatterplots of distances calculated according to different metrics on the RV12 BAliBASE dataset. Each circle corresponds to a single distance computed on one set of homologous sequences from RV12 between alignments performed by two of ClustalW, Muscle, ProbCons, T-Coffee, Prank, MAFFT L-INS-i and MAFFT FFT-NS-i. A solid line is plotted to indicate $s=y$.

4 DISCUSSION

In this study, we have defined four metrics to compare MSAs. Our metrics are superior to the most commonly used scores (i.e. the SP and TC scores) in two ways. First, our scores are valid metrics, whereas existing methods, such as the SP and TC scores are not. Secondly, our methods can incorporate indel information, including their location and when they occur during evolution. In contrast to existing scores, our metrics also allow the simple comparison of sets of sequence alignments, allowing us to investigate how similar methods of sequence alignment compare to one another rather than how well individual alignments compare to a reference alignment. This is an important difference; we find that for real data, different alignment methods tend to produce alignments with similar magnitude in distance to the BAliBASE reference alignment, but that these inferred alignments are frequently at least as different as they are from the reference alignment.

A multitude of studies have examined the effect of sequence misalignment on, for example, phylogenetic inference (Cantarel et al., 2000; Italia et al., 2004; Morrison and Hillis, 1999). Ogden and Rosenberg (2004) Wang et al. (2011) Wang et al. (2008), detection of positive selection Fletcher and Vendredi 2004, Markova-Raina and Petrov, 2013, Schneider et al. 2009; Wong et al. 2008, detection of co-occurring sites in proteins Dickson et al. 2010, studies of non-coding DNA Poliandri et al. 2009 and protein structure prediction Nakken et al. 2009. The metrics introduced here will permit several important analyses that may benefit such studies. First, our metrics allow the direct comparison of alternative alignments and their outcomes, which may be especially important as different algorithms may be subject to similar kinds of bias (Gelubchik et al. 2007). Second, comparing the differences between our metrics allows one to examine how variation in an alignment correlates with differences observed in downstream analyses. Finally, comparing sets of optimal or near-optimal alignments may allow one to discriminate between the effects brought about by uncertainty in an alignment and bias introduced by the alignment algorithms. By providing valid metrics for the direct comparison of alignments, we hope future studies will be able to more thoroughly understand the similarities and differences between alignment methods, and make progress in the task of disentangling alignment heuristics from their effect on downstream inference.

ACKNOWLEDGEMENTS

We thank two anonymous reviewers for comments that improved the manuscript.

Funding: Biotechnology and Biological Sciences Research Council (UK) (grant number BB/H000445/1).

Conflict of Interest: none declared.

REFERENCES


