VerAlign: a multiple sequence alignment assessment tool

Simossis V. A. and Heringa J.

Bioinformatics Unit, Faculty of Sciences, Vrije Universiteit, De Boelelaan 1081A, 1081

HV Amsterdam, The Netherlands

(2003)

VerAlign is a program written in ANSI C that enables the comparison between two multiple alignment files in MSF format (Genetics Computer Group, 1993). The comparison is based on two criteria, the number of correctly aligned amino acid pairs and correctly aligned alignment positions. The assessment of a generated alignment is made against the corresponding “standard of truth”.

The "standard of truth"

The "standard of truth" can be defined as a structural alignment of sequences based on experimental data and 3D structures. This means that the "standard of truth" is the biologically correct alignment of the sequences. A test sequence alignment is the result of a method that uses mathematical models to align sequences. The goal of this method is to take into account as much biological information as possible and assess it in such a way that the predicted outcome will be as similar as possible to the "standard of truth".

The test examples that will be used to outline points in this discussion are produced by PRALINE (Heringa, 1999, 2000) and are compared against the current BAliBASE reference alignment database (Thompson *et al.*, 2002). We will briefly discuss what a multiple sequence alignment file in MSF format consists of.

The MSF format

MSF is the multiple sequence alignment format of the GCG sequence analysis package. A file in MSF format has certain compulsory features that define it. Keywords "MSF:", "Type:" , and "Check:" need to be in a line that ends with two periods (dots) (see figure 3.4.1). The following abbreviations are used: **MSF**: alignment length (length of longest sequence), **Type**: P for protein sequences, N for nucleotide sequences and **Check**: gives a checksum made up of the ASCII values of the sequence characters. This value can be used to check whether an alignment has been edited since it was created.

After the periods, and preceding the alignment, there is the alignment description part consisting of: **Name**: sequence name (identifier), **Len**: the sequence length, **Check**: the checksum for the sequence and **Weight**: the sequence weight.

Every sequence name has to be unique (different names for any sequence pair). No blank is accepted within a sequence name. E.g. 'Id seq a' will be interpreted as: sequence name = Id, and amino acid 1-4 = s, e, q, a. The maximal number of characters of 'Id_seq_0' can be 13. The fields "Len:" "Check:" and "Weight:" are not used by all software but are a compulsory part of the MSF format. In the case that the software one uses does not use this information any number can be inserted.

After the alignment description part there is an essential double frontslash: "//" that acts as the termination of the header list. After this the alignment is expected to begin.

The rest of the file is interpreted as alignment. Any line not starting with a sequence identifier (as given in the header!) is ignored. If a line starts with a correct identifier, say Id_seq_n, everything following the first word of this line is appended to the sequence Id_seq_n.

```
MSF of: x.hssp from:    1 to:  46
x.msf   MSF:  46  Type: P 11-Oct-00  21:17:4   Check: 5859  ..
Name: Sequence_1              Len:  46        Check:  750    Weight:  1.00
Name: Sequence_2              Len:  46        Check:  3980   Weight:  1.00
//
Sequence_1  RQLVHVVKWA KALPGFRNLH VDDQMAVIQY SWMGLMVFAM GWRSFT
Sequence_2  .QLLSVVKWS KSLPGFRNLH IDDQITLIQY SWMSLMVFGL GWRSYK
```

Figure 3.4.1 *Representation of the MSF format showing all essential keywords.*

Comparison of two alignment blocks

When comparing two alignment blocks there are two types of data; the amino acids ("ABCDEFGHIKLMNPQRSTVWXYZ") and the gaps ("."). The most common way of scoring two alignment blocks is to assign every correctly aligned case a value of 1 and increment, while assigning 0 to the erroneous. Thus, in the end we have a value for all the identities between the two alignment blocks, which when taken as a percentage of the "expected" identities if the alignments were 100% identical will give us the percentage similarity value. We can compare the alignments by complete alignment positions ("column-wise" comparison)(see figure 3.4.2a) or by alignment of amino acid pairs, the sum-of-pairs ("pairwise" comparison) (see figure 3.4.2b).

Column-wise comparison is the stricter of the two methods. The alignment blocks are scanned in columns (i.e. amino acid 1 of all sequences, then 2, 3, 4 etc.). If there is even one amino acid aligned differently in a column, the score assigned is 0. If all amino acids and gaps are the same then the score assigned is 1.

Pairwise comparison compares all possible residue pairs in the test alignment with all the corresponding pairs in the "standard of truth". This method is more exhaustive compared to the sum of columns approach because it gives a score of 1 for every pair that has been properly aligned whether that is between sequence 1 and 2 or 1 and 40, and assigns a score of 0 only to misaligned pairs.

A) "Columnwise" Comparison

Column n°	0.....	1.....	2.....	3.....	4.....
Sequence_1	RQLVHVVKWA	KAFFGFRNLH	VDDQMAVIQY	SWMGLMVFAM	GWRSFT
Sequence_2	.QLLSVVKWS	KSLPGFRNLH	IDDQIT..QY	SWMSLMVFGL	GWRSYK
Sequence_3	...MSVVKWS	CSUYGFRNLH	GDDSYT..QY	SWMSLMVFFU	FSATRE
Sequence_1	RQLVHVVKWA	KAFFGFRNLH	VDDQMAVIQY	SWMGLMVFAM	GWRSFT
Sequence_2	QLLS.VVKWS	KSLPGFRNLH	.IDD.QITQY	SWMSLMVFGL	GWRSYK
Sequence_3	..MS.VVKWS	CSUYGFRNLH	GD.DSYT.QY	SWMSLMVFF.	UFSATR
Scores	0000011111	1111111111	0000000011	1111111100	000000

B) "Pairwise" Comparison

Column n°	0.....	1.....	2.....	3.....	4.....
Sequence_1	RQLVHVVKWA	KAFFGFRNLH	VDDQMAVIQY	SWMGLMVFAM	GWRSFT
Sequence_2	.QLLSVVKWS	KSLPGFRNLH	IDDQIT..QY	SWMSLMVFGL	GWRSYK
Sequence_3	...MSVVKWS	CSUYGFRNLH	GDDSYT..QY	SWMSLMVFFU	FSATRE
Sequence_1	RQLVHVVKWA	KAFFGFRNLH	VDDQMAVIQY	SWMGLMVFAM	GWRSFT
Sequence_2	QLLS.VVKWS	KSLPGFRNLH	.IDD.QITQY	SWMSLMVFGL	GWRSYK
Sequence_3	..MS.VVKWS	CSUYGFRNLH	GD.DSYT.QY	SWMSLMVFF.	UFSATR
Scores	1111033333	3333333333	1100000133	3333333331	111111

Figure 3.4.2. *(a) Representation of scoring system when "columnwise" comparison is applied, (b) representation of scoring system when "pairwise" comparison is applied. In (a) the maximum score (100%) will be given if all columns (46 for this example) match, while for (b) the maximum score (100%) will be given if all possible pairs (138 for this example) match. An example of a pair that seems to be correctly aligned but does not conform to both amino acid type and sequence positions is in red boxes.*

An alignment block can be represented as a series of columns and rows, so actually it is a matrix with co-ordinates. The column values are the amino acid positions in each sequence (e.g. M^{13}) and the row values are the sequence numbers (e.g. Seq1). Therefore, any given amino acid in our matrix has a unique identity, e.g. (Seq2, A^{44}) will correspond to alanine 44 of sequence 2. This is extremely important, because if there was a string of 10 alanines in two sequences being aligned and a gap was inserted between A^5 and A^6 of sequence 1, A^6-A^{10} would be shifted one position to the right with respect to the corresponding alanines of sequence 2. As a result, when compared to the "standard of truth", it could still appear that the all the matching alanines were correctly aligned, but actually they would not be correctly aligned since for some, the sequence position would be different. The "identity" of each amino acid allows for the detection of such a situation and can then assign the correct score. So, in order for a column or a pair of amino acids to be assigned an identity score of 1, they must meet two criteria; identical amino acid type and sequence position (see figure 3.4.2-red boxes).

The VerAlign algorithm

VerAlign reads in the input MSF files, regarding the first one in the command line as the "standard of truth".

Precautions:

First of all, a routine checks that the input format is the correct one and if not, explains the problem and suggests a course of action to the user. However, in many cases the files may have the required format but the data being compared, for example a series of protein sequences, could be in different orders. Unless checked, the program will read in the data as it comes and if the sequences are in different orders, then the data will be falsely labelled, processed and the result will also be false because the data from the test file will not be compared to similar data in the "standard of truth". For example, let us take sequences 1, 2 and 3. If the order in the test file is 3-1-2 and that in the "standard of truth" is 1-2-3, then sequence 3 will be compared to 1, 1 to 2 and 2 to 3. The resulting assessment will be false and extremely misleading. To avoid the confusion that could be caused by unordered files, we have also included a step that finds "the standard of truth" sequence names and amino acid sequence order and for each sequence finds and links

them to the corresponding sequences in the test file. As a result, sequence 1, being the first sequence in the "standard of truth" will always be linked to sequence 1 in the test file even if it is third or last in the test file. Therefore, given any randomly ordered set of sequences within an MSF file, VerAlign can process the information correctly. In addition, sequences that are not shared by both files are ignored, thus not overloading the system with information that will not be used. More importantly, it allows VerAlign to inform the user that the sequence content of the files is not identical and also to inform which sequences are the odd ones out. This is very important because when a multiple alignment is created the outcome is highly dependent on the information available. In general, the more information a program has the better it performs because there are more examples of similar sequences to derive conserved regions from. However, sometimes-distant sequences will make the alignment worse because of the lack of homology. The result could be interpreted in a misleading way, so it is good to know how much "extra" each file has. Optimally the files compared should contain exactly the same sequences for an accurate comparison. Added to this, after isolating the common data, it is also important to make sure that the sequence names as well as the sequences themselves are the same in both the test and the "standard of truth" files. It is possible that a name could be duplicated or one of the sequences could be corrupted in one of the files (see figure 3.4.3). Name duplication would cause a labelling error in data storage and also cause errors in the data isolation routine mentioned earlier, since the name will be found twice. In addition, it could mean that a sequence has been entered twice which would mean that the alignment would have to be repeated as the presence of identical information would have biased the alignment towards that sequences conformation. If the sequence is corrupt then it is possible that the original sequence (i.e. no gaps) is not identical in the test file and the "standard of truth" or even that the name does not correspond to the same sequence. If there are such differences, then the user is informed of which sequence and in which part of the sequence or which names are inconsistent.

MSF of: x.hssp from: 1 to: 46

x.msf **MSF:** 46 **Type:** P 11-Oct-00 21:17:4 **Check:** 5859 **..**
Name: Sequence_1 **Len:** 46 **Check:** 750 **Weight:** 1.00
Name: Sequence_2 **Len:** 46 **Check:** 3980 **Weight:** 1.00
//

```
Sequence_1  RQLVHVVKWA KAFFGFRNLH VDDQMAVIQY SWMGLMVFAM GWRSFT
Sequence_1  .QLLSVVKWS KSLPGFRNLH IDDQIT..QY SWMSLMVFGL GWRSYK
```

MSF of: x.hssp from: 1 to: 46
x.msf **MSF:** 46 **Type:** P 11-Oct-00 21:17:4 **Check:** 5859 **..**
Name: Sequence_1 **Len:** 46 **Check:** 750 **Weight:** 1.00
Name: Sequence_2 **Len:** 46 **Check:** 3980 **Weight:** 1.00
//

```
Sequence_1  RQLVHVVKWA KALPGFRNLH VDDQMAVIQY SWMGLMVFAM GWRSFT
Sequence_2  .QLLSVVKWS KSLPGFRNLH IDDQITLIQY SWMSLMVFGL GWRSYK
```

Figure 3.4.3 *Representation of corrupted MSF file that could cause the assessment to be inaccurate (Examples of corrupted sequence are in the red boxes and of name duplications are in the blue boxes).*

Comparison of alignments

Once checked, the sequence information is stored in arrays and the multiple sequence alignment is stored in a 2D matrix for each file. A third matrix is created (the *maparray*) maintaining the column identity of the longest alignment. Every amino acid in the "standard of truth" matrix is looked-up in the test matrix and the position value (column number) is stored in the *maparray* at co-ordinates matching those of the "standard of truth" matrix (see figure 3.4.4). The result of this is a 2D matrix, the *maparray*, which contains all amino acid positions in the "standard of truth" alignment at the column positions where they were found in the test alignment. Matched and unmatched gaps are given a value of -2 and -1, respectively.

At this point the assessment has conformed to the amino acid type and sequence position conditions and can proceed in two ways:

Column-wise Assessment: The maparray is scanned column-by-column checking that all the rows in each column have the same value or –2 (matched gap). If all values are the same in a column, this means that this alignment position (column) in the test alignment has been observed in the "standard of truth". For every column that is found to be matched, the score increases by one. Even if one of the sequences (rows) has a misaligned amino acid position, the score is set to 0 for the whole column. The score is

expressed as a percentage of matched columns (all rows having the same value) over total number of columns:

Percentage similarity (%) = (N^o of matched columns/ N^o of columns) * 100.

Pairwise Assessment: For this type of assessment, the *maparray* is scanned column by column, each time checking all possible residue pairs. For every identity, i.e. for every pair of numbers occurring in the same column, the score increases by one, giving the number of matched pairs. After all possible pairs have been checked, the score is expressed as a percentage of matched pairs over total number of possible pairs:

Percentage similarity (%) = (N^o of matched pairs/ N^o of all pairs) * 100.

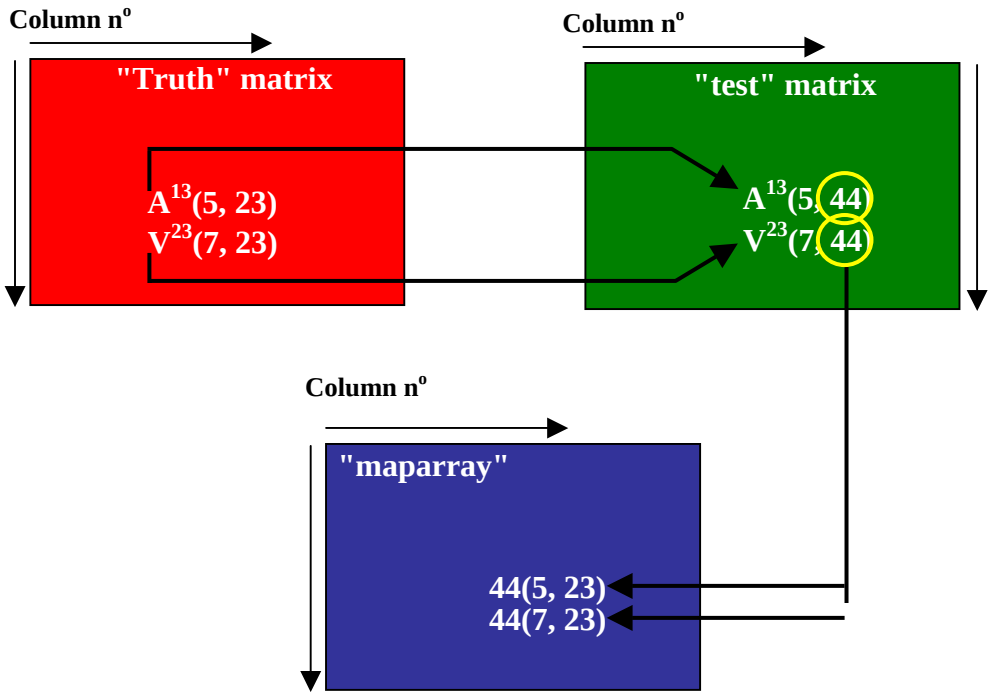

Figure 3.3.4. *Schematic representation of the method used in VerAlign to compare between alignments. The alanine 13 and valine 23 examples are shown with their co-ordinates in the two alignment matrices in brackets, where the first number is the row (sequence) number and the second is the column (amino acid position) number. The "standard of truth" alignment co-ordinate (5, 23) denotes that the alanine 13 of sequence 5 is found in column 23(this suggests the presence of 10 gaps). When alanine 13 is looked up in the "test" alignment matrix it is found in sequence 5 but in column 44. Its position is saved in the "maparray" matrix with the "standard of truth co-ordinates. This is repeated for all amino acids in the alignment and then scored by either of both ways discussed.*