Multiple alignments get faster and better

Speech recognition and sequences

by Gordon Blackshields, Iain Wallace and Des Higgins (University College Dublin, Ireland)

The number of possible gap placements in even small sequences is vast and processing requirements grow expotentially with the number of sequences. Handling genome data needs raw computing power and clear thought.

In Figure 1, we show a multiple sequence alignment of a small set of diverse globins. These sequences have changed considerably since they diverged from each other, but they are still alignable — just. Due to insertions and deletions, as well as amino acid substitutions, you need to place gaps in some of the sequences so as to try to maintain the alignment of equivalent residues. The number of possible gap placements in even a small set of sequences like this is simply enormous and designing automatic methods for doing this presented considerable problems when first attempted in the 1970s and 1980s. The obvious thing to try was to extend the famous 'dynamic programming' methods that were applied to the two-sequence alignment problem. These methods are still widely used, and are guaranteed to give the best possible alignment if you give scores for all possible amino acid matches and mismatches, and scores for gaps of different lengths. You can certainly extend these methods to seven or more sequences and the alignments are of high quality, but they take colossal amounts of computer power to construct. Worse still, the computer requirements grow exponentially with the number of sequences.

Key words: hidden Markov model, posterior probability, probabilistic consistency, progressive alignment, sequence alignment.

The main solution has been to use what is often called 'progressive alignment', which is a short-cut approach that builds the alignment up gradually. The process involves making a quick and approximate phylogenetic tree and then adding the sequences together two at a time initially, using dynamic programming at each step. An early version of this was described by Willie Taylor in London[1], and it has become the standard method for doing this ever since, with the Clustal series of programs being especially popular[2]. These methods have the advantage of being particularly fast, and huge alignments can be constructed quickly. Of greater importance, the alignments are accurate enough to be used automatically for many applications. There is a snag, however, in that the methods do fall down in very difficult situations, such as when you have very long insertions or deletions, or when the sequences have diverged too much.

One approach that has worked especially well in many areas of bioinformatics has been to use hidden Markov models (HMMs), developed initially for speech recognition. These were applied to multiple alignments in the early 1990s, but the quality of the alignments proved to be disappointing. This was despite the clear power of HMMs in other areas of bioinformatics, and also despite the clear mathematical rigour which can be applied to this approach. A second approach that was more successful was developed by Cedric Notredame and colleagues[3] and resulted in the T-Coffee computer program. This approach used progressive alignment, but attempted to find the alignment that was most similar to a 'library' of aligned pairs of sequences. This library was generated from normal fast pairwise alignments, but could also, in principle, be derived from other sources of information such as three-dimensional structures. Most importantly, the alignments were very accurate, as measured on sets of test cases.

In this review, we wish to draw attention to three recently published multiple alignment programs. The first (3-DCoffee[4]) is a version of T-Coffee, designed explicitly to align sequences and structures. The second (MUSCLE - Multiple Sequence Comparison by Log Expectation[5]) is a turbo-charged version of progressive alignment that delivers very high quality alignments very quickly. Finally, we have ProbCons[6], which is a way of applying the T-Coffee algorithm, but using probabilities instead of simple scores and

```
Human β-globin      --------VHLTPEEKSAVTALWGKVN--VDEVGGEALGRLLVVYPWTQRFFESFGDLST
Horse β-globin      --------VQLSGEEKAAVLALWDKVN--EEEVGGEALGRLLVVYPWTQRFFDSFGDLSN
Human α-globin      ---------VLSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHF-DLS-
Horse α-globin      ---------VLSAADKTNVKAAWSKVGGHAGEYGAEALERMFLGFPTTKTYFPHF-DLS-
Whale myoglobin     ---------VLSEGEWQLVLHVWAKVEADVAGHGQDILIRLFKSHPETLEKFDRFKHLKT
Lamprey globin      PIVDTGSVAPLSAAEKTKIRSAWAPVYSTYETSGVDILVKFFTSTPAAQEFFPKFKGLTT
Lupin globin        --------GALTESQAALVKSSWEEFNANIPKHTHRFFILVLEIAPAAKDLFSFLKGTSE
                              *:  :   :   *  .           :  .:   * :   *  :   .

Human β-globin      PDAVMGNPKVKAHGKKVLGAFSDGLAHLDN-----LKGTFATLSELHCDKLHVDPENFRL
Horse β-globin      PGAVMGNPKVKAHGKKVLHSFGEGVHHLDN-----LKGTFAALSELHCDKLHVDPENFRL
Human α-globin      ----HGSAQVKGHGKKVADALTNAVAHVDD-----MPNALSALSDLHAHKLRVDPVNFKL
Horse α-globin      ----HGSAQVKAHGKKVGDALTLAVGHLDD-----LPGALSNLSDLHAHKLRVDPVNFKL
Whale myoglobin     EAEMKASEDLKKHGVTVLTALGAILKKKGH-----HEAELKPLAQSHATKHKIPIKYLEF
Lamprey globin      ADQLKKSADVRWHAERIINAVNDAVASMDDT--EKMSMKLRDLSGKHAKSFQVDPQYFKV
Lupin globin        VP--QNNPELQAHAGKVFKLVYEAAIQLQVTGVVVTDATLKNLGSVHVSKGVAD-AHFPV
                          . .:: *.  :   .                  :  *.  *  .       : .

Human β-globin      LGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVANALAHKYH------
Horse β-globin      LGNVLVVVLARHFGKDFTPELQASYQKVVAGVANALAHKYH------
Human α-globin      LSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSKYR------
Horse α-globin      LSHCLLSTLAVHLPNDFTPAVHASLDKFLSSVSTVLTSKYR------
Whale myoglobin     ISEAIIHVLHSRHPGDFGADAQGAMNKALELFRKDIAAKYKELGYQG
Lamprey globin      LAAVIADTVAAG---D------AGFEKLMSMICILLRSAY-------
Lupin globin        VKEAILKTIKEVVGAKWSEELNSAWTIAYDELAIVIKKEMNDAA---
                    :   :  .:      .      ..       .   :
```

Figure 1. A multiple alignment of seven diverse globin sequences. These have been aligned using ClustalW and the approximate locations of the α-helices, common to all of them, are shown in red.

using HMMs after all, but in this case to generate the initial alignments and residue–residue weights. This program now looks like it is the most accurate method yet.

3-D Coffee

3-D Coffee is a computer program for making multiple alignments of protein sequences that incorporate structural information from three-dimensional structures, when any structures exist. It is a common occurrence to have a set of sequences that one wishes to align, and to have one or two structures available. In principle, you can align very distantly related sequences when you know the three-dimensional structures of all of them because you can see which structural elements (α-helices and β-strands) align with which, even when the sequences have diverged well into the twilight zone. The latter is the situation where the amino acid sequences are less than 25% identical and therefore difficult to align. In Figure 2, we have the structures corresponding to the globin example from Figure 1, and, clearly, these are very easy to match up when you use the structures. Such structural information is clearly important and useful, but it is not necessarily easy to exploit in simple multiple alignment programs. Various complex packages and algorithms have been developed that can mix sequences and structures in alignments, ranging from full multiple structure superpositions to sequence/structure threaders and aligners.

Here, we wish to draw your attention to a fast, simple and accurate way to incorporate structures in an alignment, and to derive an improvement in overall alignment accuracy, even when you only have one or two structures available. The method exploits the ability of the T-Coffee alignment program to incorporate heterogeneous alignment information. T-Coffee, by default, generates pairs of sequence alignments between all of the input sequences and finds a multiple alignment that is most compatible with these. If you have one structure, you could, in principle, match each sequence to the structure, using a threading package and convert the output of each threading into a two-sequence alignment. These two-sequence alignments contain information about how the structure aligns to each sequence, and, indirectly, how the sequences align to each other. These alignments can be fed into T-Coffee, and if you do this, the accuracy of the multiple alignments increases.

The difficulty is that many threading packages are difficult to install or have heavy computational requirements. 3-DCoffee does all of this for you automatically. It uses the FUGUE[7] package, which carries out sequence alignment by dynamic programming, but by using information about local features of the structure. You do not even have to have FUGUE installed, as the program will take your structure and a sequence, and will pass these to the FUGUE server and will request an alignment before collecting the results and incorporating them automatically. All you need is a connection to the Internet, a copy of 3-DCoffee and some sequences, including at least one with a structure in the Protein Data Bank (PDB). If installing the package locally scares you off, you can just use the 3-DCoffee server and do everything online.

If you have two or more structures, these can also be matched using a full structure superposition package, such as SAP[8]. The more structures you include, the better (more accurate) will be your alignments. Furthermore, you are not restricted to using just SAP and FUGUE (though the 3-DCoffee server uses these programs by default). You can use any outside software that can convert a pair of structures or a sequence and a structure into an alignment or a set of alignments.

MUSCLE

A major issue in alignment program development has been the trade off between alignment accuracy and computational complexity. Up until this year, the most accurate method was T-Coffee. A further advantage of T-Coffee was its ability to combine data from heterogeneous sources (as described above for structures). Nonetheless, ClustalW continues to be extremely popular, partly due to its numerous interface features and menu options, but also partly because it is much faster than T-Coffee, especially if you try to align more than 50 sequences.

MUSCLE is a new multiple sequence alignment package that is extremely fast, indeed faster than ClustalW, but which also delivers alignments as accurate as T-Coffee. It is a progressive alignment program, but it is very highly optimized for speed, and uses a neat profile-to-profile alignment method to deliver accuracy.

Initially, the goal is to rapidly generate a rough draft of the alignment, emphasizing speed over accuracy. Crude evolutionary distances between pairs of input sequences are estimated by *k*-mer or *k*-tuple (short exact matches of fixed length) counting using a compressed alphabet. The 20 standard amino acids are divided into seven subgroups sharing biochemical similarity. These distances are clustered to give an initial tree, which is then used to construct a progressive alignment between the sequences. At each stage of the progressive alignment, groups of sequences (profiles) are aligned. One important improvement with MUSCLE is to use the Log Expectation (LE) score, which has been shown to outperform other functions in correctly aligning pairs of columns:

$$\mathrm{LE}^{xy} = (1-f_G^x)\ (1-f_{\mathrm{G}}^y)\log\sum_i\sum_j f_i^x f_j^y \frac{\mathrm{P}_{ij}}{\mathrm{P}_i\mathrm{P}_j}$$

In this function, *i* and *j* are different amino acid types, P_i is the background probability of *i*, P_{ij} is the joint probability of *i* and *j* being aligned. f_i^x is the observed frequency of amino acid type *i* in column *x* of the first profile, f_G^x is the observed frequency of gaps at that column (similarly for column *y* in profile 2). The factor $(1-f_G^x)$ encourages more highly occupied columns to align to each other.

The next stage in the process attempts to refine the rough draft. A second tree is constructed, this time using a more sophisticated distance model, which is more accurate, but needs an alignment as input. A second progressive alignment is generated using this refined tree. For added speed, new pairwise profile alignments are calculated only for those sub-trees that changed relative to the initial tree.

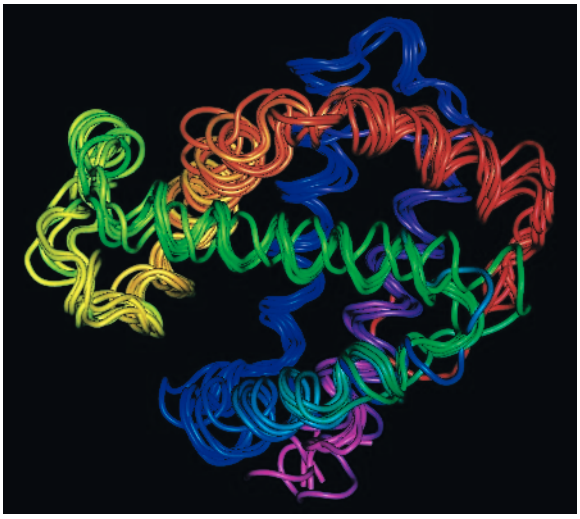

Figure 2. A multiple structure superposition of the structures corresponding to the sequences in Figure 1. These have been superimposed using the VAST algorithm and displayed using Cn3D, both available at http://www.ncbi.nlm.nih.gov/Structure/. The α-helices are shown as spirals of different colour.

An iteration step is included to improve the alignment quality further. Within the tree, an edge connects two sub-trees; in order of decreasing distance from the root, each edge is visited and deleted, splitting the tree in two. The profiles of these two sub-trees are realigned. If the alignment score is improved, the alignment is kept, otherwise it is discarded. This step can be repeated until no further improvements can be made.

During development, MUSCLE was assessed using several alignment databases, including the BAliBASE[9] benchmark, on which it achieved the highest ranking of any method at the time of publication.

On a typical PC, T-Coffee is unable to align more than 100 typical length sequences, while ClustalW can easily manage hundreds. Beyond this, however, the computational complexity becomes increasingly prohibitive. The largest analysis described for MUSCLE, contained 5000 computer-generated sequences; MUSCLE was able to complete the task in 7 minutes (if the iteration step was ignored), while it was estimated that for ClustalW, if left running continuously, a full year would have been required.

Table 1. *URLs for multiple alignment packages*

3-DCoffee	http://igs-server.cnrs-mrs.fr/Tcoffee/tcoffee_cgi/index.cgi
MUSCLE	http://www.drive5.com/muscle/
ProbCons	http://probcons.stanford.edu/
FUGUE	http://www-cryst.bioc.cam.ac.uk/fugue/
ClustalW	http://www.ebi.ac.uk/clustalw/

ProbCons

T-Coffee derives some of its accuracy from its ability to take alignment information from mixtures of input alignments. It derives further accuracy from a consistency step where the input aligned pairs of amino acids are compared with each other. Pairs of aligned residues that agree or are consistent with other pairs get up-weighted. The weights that are used are simple scores, related to the similarity of the parent sequences that the pairs come from. ProbCons, uses a very similar approach, but with one major difference: the scores that determine which pairs of amino acids should be aligned are probabilities. The probabilities are generated as the posterior alignment probabilities from pairwise HMMs. Furthermore, the up-weighting of consistent pairs of amino acids is now done by multiplying these posterior probabilities. After this, ProbCons, aligns the sequences using progressive alignment, just as in T-Coffee. ProbCons is now the most accurate multiple alignment method as benchmarked using BAliBASE. In their paper, Do et al.[6] show that ProbCons performs best in all five of the reference sets that make up BAliBASE. It also finds the unique best alignment in 46.1% of the BAliBASE cases, as well as the joint best alignment in 66.7% of the cases.

The first step in the ProbCons algorithm is to align all of the sequences with each other using a pair-HMM. By modelling an alignment as a pair-HMM, Do et al.[6] were able to calculate the posterior probability, $P(x_i{\sim}y_j|x,y)$ for each pair of amino acids. This is the probability that residue i in sequence x is matched with residue j in sequence y in the final alignment. At this stage, the expected accuracy of each pairwise alignment is also calculated, which is defined as the sum of the posterior probabilities for each alignment.

The second step is to calculate the probabilistic consistency (hence the name of the program). If a third homologous sequence, z, is available, a better estimate of the posterior probability of $P(x_i{\sim}y_j)$ can be obtained using the information about how x and y align with z. This is defined as $P(x_i{\sim}y_j|x,y,z)$ and may be calculated using a three sequence HMM, but that would be an $O(L^3)$ calculation. A cubic running time was considered undesirable, so the following heuristic was implemented which can be solved in approximately constant time:

$$P(x_i \times y_j x,y,z) \approx \sum_k P(x_i \times z_k|x,z)P(y_j \times z_k|y,z)$$

The multiple alignment is generated by using a progressive alignment scheme. The guide tree is calculated by clustering the sequences based on their expected accuracy. Sub-alignments are combined using a sum-of-pairs scheme, in which the score of the multiple alignment is calculated by summing all posterior probabilities for all pairs of sequences present. The final alignment is then subjected to an iterative refinement protocol. The alignment is randomly split into two groups and realigned. This is repeated 100 times.

All of these packages are available online and/or can be downloaded for local use. The URLs are given in Table 1.

References

1. Taylor, W.R. (1988) J. Mol. Evol. **28**, 161–169
2. Thompson, J.D., Higgins, D.G. and Gibson, T.J. (1994) Nucleic Acids Res. **22**, 4673–4680
3. Notredame, C., Higgins, D.G. and Heringa, J. (2000) J. Mol. Biol. **302**, 205–217
4. O'Sullivan, O., Suhre, K., Abergel, C., Higgins, D.G. and Notredame, C. (2004) J. Mol. Biol., **340**, 385–395
5. Edgar, R.C. (2004) Nucleic Acids Res. **32**, 1792–1797
6. Do, C.B., Brudno, M. and Batzoglou, S. (2004) Bioinformatics **20** (Suppl. 1), in the press
7. Shi, J., Blundell, T.L. and Mizuguchi, K. (2001) J. Mol. Biol. **310**, 243–257
8. Taylor, W.R. and Orengo, C.A. (1989) J. Mol. Biol. **208**, 1–22
9. Thompson, J.D., Plewniak, F. and Poch, O. (1999) Nucleic Acids Res. **15**, 87–88

Gordon Blackshields graduated in 2003 with an honours degree in Biochemistry from University College Cork. He is currently working towards a PhD in University College Dublin, focusing on the evaluation of multiple sequence alignment programs and their benchmark databases. This work is funded by Science Foundation Ireland.

Gordon.blackshields@ucd.ie

Iain Wallace graduated from Trinity College Dublin in 2001, with a degree in Computational Chemistry and subsequently obtained a masters in High Performance Computing. Currently he is investigating new multiple alignment methods for his PhD at University College Dublin. This work is funded by the Science Foundation Ireland.

Iain.Wallace@ucd.ie

Des Higgins obtained his primary and PhD degrees from Trinity College Dublin in the 1980s and has worked on bioinformatics, especially sequence alignment methods ever since. He started as a bioinformaticist during a 4 year post-doc. with Paul Sharp in Dublin in 1985 and then worked at the EMBL Data Library in Heidelberg for four years and the European Bioinformatics Laboratory in Hinxton for a further two. He was a member of the Biochemistry Department in University College Cork from 1997 until 2003. He is currently Professor of Bioinformatics in the Conway Institute, University College Dublin.

Des.Higgins@ucd.ie