# November 5, 2012

Downloaded "hmm experimentation/cluster73.hhm" from the HHOMP website. There's a link at <http://toolkit.tuebingen.mpg.de/hhomp> that says "Download HHOMP database". From there, I downloaded the file <ftp://toolkit.genzentrum.lmu.de/pub/HHomp/db/hhm_files.tar.gz>, which contained "cluster73.hhm".

Copied the file " zdiffs\gonnet\_aligned\cluster73, 2MPR as target, 1A0S as template" into this folder, as "hmm experimentation/gonnet aligned cluster73 with 2mpr and 1a0s.clu".

Downloaded the ClustalΩ readme from " <http://www.clustal.org/omega/README>".

This passage is very pertinent to doing an alignment using a downloaded HMM:

"Prior to MSA, Clustal-Omega de-aligns all sequence input (i). However,

alignment information is automatically converted into a HMM and used

during MSA, unless the --dealign flag is specifically set. Profiles

(ii) are not de-aligned."

I don't understand exactly how the program works, but it seems redundant to build an HMM from the alignment when I already have an HMM that I want to use. Also, to quote the 2009 HHOMP paper in Nucleic Acids Research, "A profile HMM was built for each cluster from a multiple alignment of its sequences". The multiple alignment I have is almost exactly the one from the cluster.

It is possible, however, that if I *remove* sequences *before* doing the alignment, I could benefit from making an HMM from the alignment. After all, if there is a group of sequences distantly related to the template with an insertion that the template doesn't have and none of its close relatives have, then there's goingto be a position where there's a high probability of transition into the insert state. But every protein I actually *want* to align has a transition to the match state there. Since that transition to the match state isn't as likely as it would be if I left out those distant sequences, the true alignment is a little less likely given the HMM. On the other hand, another possibility is that many of the distantly related proteins ahve an insertion that only a few of the closely related ones have, and an HMM that was built using the distantly related ones will align those ones better, without misaligning those without the insertion. So I'm not sure whether deleting sequences before alignment would help, it would take work, my test isn't so good so I couldn't really be sure that it helped, and it doesn't matter that much anyway. I'm not going to delete sequences before alignment.

ClustalΩ seems to crash when I try to run it with a precalculated HMM:

C:\cygwin\home\alex\beta-barrel-oligomerization\hmm experimentation>clustalo -i

"gonnet aligned cluster73 with 2mpr and 1a0s.clu" --hmm-in="cluster73.hhm" --dea

lign -o "hmm aligned cluster73 with 2mpr and 1a0s.fasta" --outfmt=fasta -v --for

ce > "alignment output for cluster73.txt"

WARNING: Ignoring line

'CONF 100.0'

in HMM cluster73

WARNING: Ignoring line

'BBDOM 92 - 511'

in HMM cluster73

WARNING: Ignoring line

'CDCVA 92 - 511 : 50'

in HMM cluster73

WARNING: Ignoring line

'CDSCR 92 - 511 : 50'

in HMM cluster73

After that, it crashes. The only output written to file is:

Using 8 threads

Read 53 sequences (type: Protein) from gonnet aligned cluster73 with 2mpr and 1a0s.clu

Dealigning already aligned input sequences as requested.

For what it's worth, that *is* the correct number of sequences:

>>> x = AlignIO.read(r'C:\cygwin\home\alex\beta-barrel-oligomerization\hmm experimentation\gonnet aligned cluster73 with 2mpr and 1a0s.clu', 'clustal')

>>> len(x)

53

However, when I ran it without an input HMM, it ran smoothly, giving no warnings. The command used:

C:\cygwin\home\alex\beta-barrel-oligomerization\hmm experimentation>clustalo -i

"gonnet aligned cluster73 with 2mpr and 1a0s.clu" -o "hmm aligned cluster73 with

2mpr and 1a0s.fasta" --outfmt=fasta -v --force > "alignment output for cluster7

3.txt"

"-v" is for "verbose" so I get a lot of output, "--force" just means it will overwrite a file that has the name I gave it as the output name.

It produced the following output:

Using 8 threads

Read 53 sequences (type: Protein) from gonnet aligned cluster73 with 2mpr and 1a0s.clu

Input sequences are aligned. Will turn alignment into HMM and add it to the user provided background HMMs.

Input sequences are aligned. Will use Kimura distances of aligned sequences.

Using 32 seeds (chosen with constant stride from length sorted seqs) for mBed (from a total of 53 sequences)

Calculating Kimura-corrected pairwise aligned identity distances...

Pairwise distance calculation progress: 0 % (0 out of 1200)Pairwise distance calculation progress: 0 % (0 out of 1200)Pairwise distance calculation progress: 0 % (1 out of 1200)Pairwise distance calculation progress: 0 % (2 out of 1200)Pairwise distance calculation progress: 0 % (3 out of 1200)Pairwise distance calculation progress: 0 % (4 out of 1200)Pairwise distance calculation progress: 0 % (5 out of 1200)Pairwise distance calculation progress: 0 % (8 out of 1200)Pairwise distance calculation progress: 10 % (126 out of 1200)Pairwise distance calculation progress: 20 % (246 out of 1200)Pairwise distance calculation progress: 25 % (307 out of 1200)Pairwise distance calculation progress: 27 % (326 out of 1200)Pairwise distance calculation progress: 29 % (350 out of 1200)Pairwise distance calculation progress: 34 % (411 out of 1200)Pairwise distance calculation progress: 35 % (423 out of 1200)Pairwise distance calculation progress: 41 % (495 out of 1200)Pairwise distance calculation progress: 41 % (498 out of 1200)Pairwise distance calculation progress: 44 % (531 out of 1200)Pairwise distance calculation progress: 47 % (565 out of 1200)Pairwise distance calculation progress: 48 % (579 out of 1200)Pairwise distance calculation progress: 56 % (683 out of 1200)Pairwise distance calculation progress: 58 % (706 out of 1200)Pairwise distance calculation progress: 59 % (711 out of 1200)Pairwise distance calculation progress: 61 % (741 out of 1200)Pairwise distance calculation progress: 63 % (758 out of 1200)Pairwise distance calculation progress: 69 % (832 out of 1200)Pairwise distance calculation progress: 77 % (933 out of 1200)Pairwise distance calculation progress: 85 % (1029 out of 1200)Pairwise distance calculation progress: 87 % (1046 out of 1200)Pairwise distance calculation progress: 90 % (1082 out of 1200)Pairwise distance calculation progress: 93 % (1125 out of 1200)Pairwise distance calculation progress: 95 % (1147 out of 1200)Pairwise identity calculation progress done. CPU time: 0.01u 00:00:00.00 Elapsed: 00:00:00

mBed created 1 cluster/s (with a minimum of 1 and a soft maximum of 100 sequences each)

Distance calculation within sub-clusters: 0 % (0 out of 1)Distance calculation within sub-clusters done. CPU time: 0.00u 00:00:00.00 Elapsed: 00:00:00

Guide-tree computation (mBed) done.

Progressive alignment progress: 1 % (1 out of 52)Progressive alignment progress: 3 % (2 out of 52)Progressive alignment progress: 5 % (3 out of 52)Progressive alignment progress: 7 % (4 out of 52)Progressive alignment progress: 9 % (5 out of 52)Progressive alignment progress: 11 % (6 out of 52)Progressive alignment progress: 13 % (7 out of 52)Progressive alignment progress: 15 % (8 out of 52)Progressive alignment progress: 17 % (9 out of 52)Progressive alignment progress: 19 % (10 out of 52)Progressive alignment progress: 21 % (11 out of 52)Progressive alignment progress: 23 % (12 out of 52)Progressive alignment progress: 25 % (13 out of 52)Progressive alignment progress: 26 % (14 out of 52)Progressive alignment progress: 28 % (15 out of 52)Progressive alignment progress: 30 % (16 out of 52)Progressive alignment progress: 32 % (17 out of 52)Progressive alignment progress: 34 % (18 out of 52)Progressive alignment progress: 36 % (19 out of 52)Progressive alignment progress: 38 % (20 out of 52)Progressive alignment progress: 40 % (21 out of 52)Progressive alignment progress: 42 % (22 out of 52)Progressive alignment progress: 44 % (23 out of 52)Progressive alignment progress: 46 % (24 out of 52)Progressive alignment progress: 48 % (25 out of 52)Progressive alignment progress: 50 % (26 out of 52)Progressive alignment progress: 51 % (27 out of 52)Progressive alignment progress: 53 % (28 out of 52)Progressive alignment progress: 55 % (29 out of 52)Progressive alignment progress: 57 % (30 out of 52)Progressive alignment progress: 59 % (31 out of 52)Progressive alignment progress: 61 % (32 out of 52)Progressive alignment progress: 63 % (33 out of 52)Progressive alignment progress: 65 % (34 out of 52)Progressive alignment progress: 67 % (35 out of 52)Progressive alignment progress: 69 % (36 out of 52)Progressive alignment progress: 71 % (37 out of 52)Progressive alignment progress: 73 % (38 out of 52)Progressive alignment progress: 75 % (39 out of 52)Progressive alignment progress: 76 % (40 out of 52)Progressive alignment progress: 78 % (41 out of 52)Progressive alignment progress: 80 % (42 out of 52)Progressive alignment progress: 82 % (43 out of 52)Progressive alignment progress: 84 % (44 out of 52)Progressive alignment progress: 86 % (45 out of 52)Progressive alignment progress: 88 % (46 out of 52)Progressive alignment progress: 90 % (47 out of 52)Progressive alignment progress: 92 % (48 out of 52)Progressive alignment progress: 94 % (49 out of 52)Progressive alignment progress: 96 % (50 out of 52)Progressive alignment progress: 98 % (51 out of 52)Progressive alignment progress: 100 % (52 out of 52)Progressive alignment progress done. CPU time: 8.12u 00:00:08.11 Elapsed: 00:00:08

Alignment written to hmm aligned cluster73 with 2mpr and 1a0s.fasta

The alignment was very quick (eight seconds, according to the log), so there is really no point in using a precalculated HMM. It is probably not quite the same alignment that I would have gotten with the HMM downloaded from HHOMP, because I began with a Gonnet alignment made with ClustalW, rather than one of the Kalign alignments that HHOMP used.

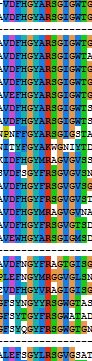
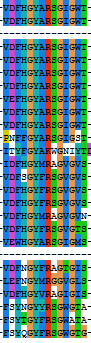
The new ClustalΩ alignment is visibly different from the old ClustalW one. At the beginning, the ClustalW alignment has a few proteins with a long insertion at the beginning, and then another group of proteins with a short insertion at the beginning that aligns to the middle of the long one. The ClustalΩ alignment instead puts this short insertion *after* the long insertion.

It would be interesting to see if the alignments are different in the strand regions, and especially to see if the alignments are different in the strands that ClustalW misaligned.

# November 9, 2012

I'm opening 2MPR in PyMOL, and using PyMOL's sequence viewer to find the sequences of each strand. Then, I'm visualizing these strands in each of the two alignments (the one made with ClustalW using the Gonnet series, and the one made with ClustalΩ)

First strand of 2MPR. Gonnet on the left, HMM on the right. 2MPR is on top, V to T is the beta sheet region according to PyMOL's definition of secondary structure.



Second strand, T to E

|  |  |
| --- | --- |
| Gonnet | HMM |
|  |  |

Third strand, K to V

|  |  |
| --- | --- |
| Gonnet | HMM |
|  |  |

Wow. They are *almost exactly the same*. This is unexpected. Based upon the misalignments in the zdiff comparisons, you'd expect like maybe one in twenty sequences to be misaligned in a strand, and perhaps aligned correctly by a better sequence alignment program.

(h1) Maybe when a strand is misaligned, it messes up the alignment for *all* sequences?

(h2) The other reason I can think of that I wouldn't be seeing any misalignments is that, due to the missing loops, sequences pulled from crystal structures align unusually poorly, and that's what I was evaluating the zdiff on.

I can test h1 by looking at the misaligned strands.

I opened a zdiff colored structure of 2MPR using the "zdiff\_report" function of "zdiffs/color.py". This function retrieves the structure of a prediction target, and colors it by how close the predicted z value is to the real z value, when making predictions from homology models built with the Gonnet series alignments.

Strand 8 (first misaligned strand), from K to L

|  |  |
| --- | --- |
| Gonnet | HMM |
|  |  |

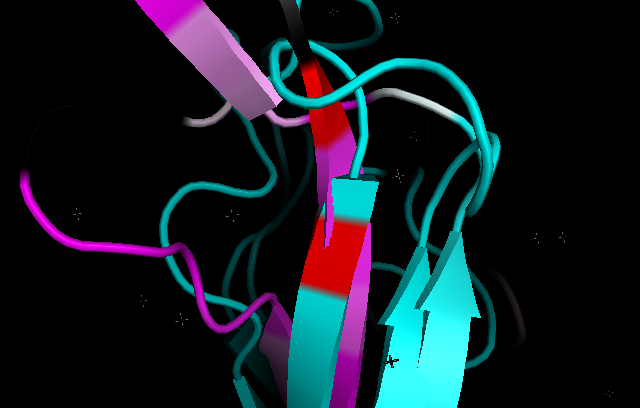
No, it's still not that different. The ClustalΩ has less gaps and is probably better. There really are not insertions or deletions in strands, probably, that would mess everything up for the barrel. Or maybe it wouldn't. But these alignments are chemical rather than evolutionary to a certain extent: if half ot hese β barrels *did* have an insertion in a strand since the common ancestor, the insertion probably evolved to have the same frequency profile as the residues it displaced. So, when I see gaps in a strand region, I'm inclined to say "misalignment". But what do I really know about that anyway, I've never really tested my understanding of the mechanics of sequence alignment.

I'm going to expand the view a bit so I can see to what extent the template, 1A0S, was aligned differently. The correct sequence of that strand of 1A0S is VQNYILTMNHF. And remember we're looking at the same strand of 2MPR as we were before: KDTANDVFDVRL.

|  |  |
| --- | --- |
| Gonnet | HMM |
|  |  |

This is *very* surprising. Nothing prepared me for this. The Gonnet alignment seems to have aligned them correctly, though this is the very same Gonnet alignment from which I produced the zdiffs that say that this strand was aligned wrong! And, in the *other* alignment, it's aligned wrong! It's enough to make me wonder whether I accidentally switched the names somehow. So, I went back to the original file: "zdiffs\gonnet\_aligned\cluster73, 2MPR as target, 1A0S as template". And it's definitely the one labeled "gonnet aligned" in this folder: they both have the characteristic feature in the beginning, where there are, near the top, sequences surrounded on both sides by long insertions.

Actually, the explanation is simple. The initial K that I've been considering the start of the strand in 2MPR is a small distance away from the initial V that I've been considering the start of the strand in 1A0S, in PyMOL's structural alignment. Here is an image in which strand 8 of 2MPR is purple, strand 8 of 1A0S is cyan, and the K and V are colored red. Since they're apart in the structural alignment, we figure they're also at different depths in the membrane.



Really, the Gonnet and HMM alignments are kind of different for strand 8. Most of the proteins of unknown structure have at least *part* of the strand aligned the same way. But the gaps are different in 8 out of the 28 I put in that original screenshot. In the first three strand I looked at, I only saw one in ttoal that was different at *all*. This definitely supports the idea that there are some strands that the two alignment programs align differently for many of the sequneces; the similarity in thefirst three strands cannot be taken as representative, even though with so many sequences looked at it seems like a lot of data. This is a lack of independence: since we know that similarity of the alignment is a strand-by-strand thing, then when we see for a particular strand that the first half of the sequences were aligned the same way, we already then expect the second half to be aligned the same way, and when we see that they are it doesn't really give us any more information. You need to look at a lot of strands to judge the similarity of the alignments, not just a lot of sequences.