# August 24, 2012

Used the scripts in "automate cluster download" to download all the HHOMP clusters. Put them in the folder "automated sequence alignment/clusters".

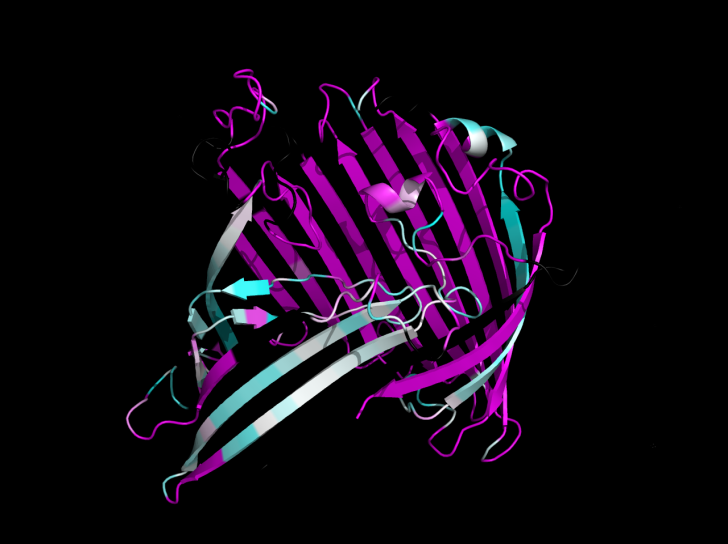
Copied the folder "aligned structures" from the folder "june 2012 lab meeting"

Copied the fodler "TMout" from the folder "bbtm derivation".

Wrote code that does not use ClustalW's "reset all gaps before alignment option". I am not sure if this is necessary but eventually I should ue this option to be safe.

# August 27, 2012

I don't know if we should actually be using BBTM. Maybe the reason the BBTM40 alignment was so off was actually because most of the protein has the substitution rate of soluble protein. This didn't used to concern me, because I don't care about aligning th eloops right. But if Clustal is sacrificing alignment score in the transmembrane region to achieve it in the loops, then we'd see those terrible misalignments. The loops might be aligned decently, though, though they're being aligned with the wrong matrix. Yeah. Oh my god they are. Here's the zdiff for bbtmout colored 0 is cyan 3 is magenta from "june 2012 lab meeting/pictures":



I don't know if I'd say it's aligning that well at the expense of the other parts though... that doesn't even really make sense. Aligning one part correctly *helps* align anothe rpart correctly.

Could it be, though, that better alignment of the less conserved loop regions guides the gonnet alignment?

This is made less likely by the fact that only like 43% of the protein is loop, determined using HHOMP's ProfTMB prediction for one of the two clusters most similar to the proteins that zdiff was calcualted for, cluster 18.1.1:

>>> x = AlignIO.read(r'C:\cygwin\home\alex\beta-barrel-oligomerization\automated\_sequence\_alignment\clusters\OMP.18.1.1.clu', 'clustal')

>>> bbpred = x[2]

>>> bbpred

SeqRecord(seq=Seq('------------------------------------------------------...I--', SingleLetterAlphabet()), id='bb\_pred', name='<unknown name>', description='bb\_pred', dbxrefs=[])

>>> tm = sum(i **in** ('U', 'D') **for** i **in** bbpred)

>>> l = sum(i **in** ('I', 'O') **for** i **in** bbpred)

>>> l/(tm+l)

0.43842364532019706

OmpA is a lot more loopy, though, it's 70% loop:

>>> x = AlignIO.read(r'C:\cygwin\home\alex\beta-barrel-oligomerization\automated\_sequence\_alignment\clusters\OMP.8.1.1.clu', 'clustal')

>>> x[2]

SeqRecord(seq=Seq('-----------------------------IIIIIII--IIII-I----------...---', SingleLetterAlphabet()), id='bb\_pred', name='<unknown name>', description='bb\_pred', dbxrefs=[])

>>> bbpred = x[2]

>>> tm = sum(i **in** ('U', 'D') **for** i **in** bbpred)

>>> l = sum(i **in** ('I', 'O') **for** i **in** bbpred)

>>> l/(tm+l)

0.7

To whatever extent using a more refined substitution matrix can help, to that same extent it hurts to have two halves of the protein with different subsitution rates! Making these BBTM matrices and automating ClustalW seems not worth it, though further testing would make it more clear. Can an HMM method capture the varying ocnditions better than a substitution matrix?