# 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". deleted August 28 2012

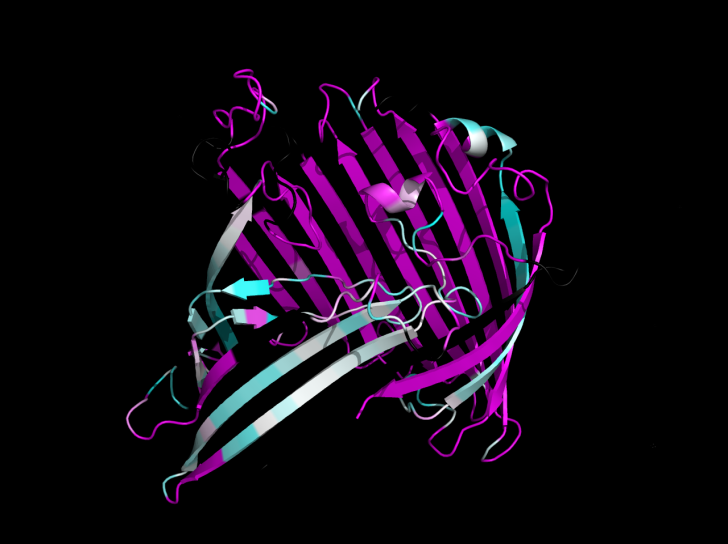
Copied the folder "aligned structures" from the folder "june 2012 lab meeting" renamed "zdiff test structures" August 28 2012

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?

# August 28, 2012

Changed name of "aligned structures" to "zdiff test structures".

Today Daniel sent me a draft of his thesis. It contains the following table as Table 4:

Table 4 - Unique clusters associated with Proteins in our Dataset

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| number | PDB\_ID | Cluster Name | subclusters | # of homol. seqs in MSA |
| 1 | 1A0S | cluster73 | 18.1.1  18.1.2 | 51 |
| 2 | 1AF6 |  | 18.1.1 | n/a |
| 3 | 1E54 | cluster28 | 16.2.1-16.2.5  16.2.8 | 319 |
| 4 | 1FEP | cluster8 | 22.4.2  22.4.4 | 81 |
| 5 | 1I78 |  | 10.1.1 | 15 |
| 6 | 1K24 |  | 10.2.1 | 2 |
| 7 | 1KMO |  | 22.2.4 | 31 |
| 8 | 1P4T | cluster144 | 8.1.5  8.6.3  8.6.2 | 76 |
| 9 | 1QD6 |  | 12.6.1 | 85 |
| 10 | 1QFG |  | 22.1.4 | 224 |
| 11 | 1QJ8 |  | 8.3.1 | 31 |
| 12 | 1QJP | cluster75 | 8.1.1  nn.31.1 | 78 |
| 13 | 1T16 |  | 14.1.1 | 162 |
| 14 | 1THQ |  | 8.5.1 | 14 |
| 15 | 1TLY | cluster108 | 12.5.1  12.5.2 | 36 |
| 16 | 1UYN |  | 12.1.6 | 57 |
| 17 | 2ERV |  | 8.4.1 | 44 |
| 18 | 2F1C |  | 14.2.1 | n/a |
| 19 | 2F1V |  | 8.2.1 | 141 |
| 20 | 2GUF |  | 22.4.5 | 61 |
| 21 | 2J1N | cluster99 | 16.1.1  16.1.2 | 77 |
| 22 | 2O4V |  | 16.4.2 | 75 |
| 23 | 2POR | cluster131 | 16.2.1  16.2.3 | n/a |
| 24 | 2QDZ | cluster53 | nn.5.1  nn.5.2  nn.5.4  cluster43 | 152 |
| 25 | 2VQI |  | nn.2.2 | 206 |
| 26 | 2WJR |  | nn.36.1 | 35 |
| 27 | 3BS0 | cluster71 | 14.1.5  14.1.7  14.1.1  cluster62 | 194 |
| 28 | 3CSL |  | 22.4.6 | 67 |
| 29 | 3DWO |  | 14.1.1 | n/a |
| 30 | 3DZM | cluster165 | 8.1.1 | n/a |
| 31 | 3EFM | cluster18 | cluster6  22.1.7  22.1.4  22.1.5  22.1.3  22.1.6 | 531 |
| 32 | 3EMN |  | nn.54.1 | 25 |
| 33 | 3FHH |  | 22.4.6 | n/a |
| 34 | 3JTY |  | nn.9.1 | 172 |
| 35 | 3PRN | cluster131 | 16.2.1  16.2.3 | n/a |

I copied this table to an Excel document, and removed the red rows: 35, 33, 30, 29, 23, 18, and 2. I saved it as "structure dataset and clusterguide.csv".

I deleted the "clusters" folder - too many clusters in it. Made a new folder "~~ezbeta~~ clusters", which I will now opulate with just the clusters required.

Used this code tomake a file "~~ezbeta~~ clusters/clusters.csv" containing all the above clusters:

**import** csv

pdbid\_clusterid = list()

with open('structure dataset and clusterguide.csv', 'rb') as f:

first = True

**for** row **in** csv.reader(f):

# Skip the first line

**if** first:

first = False

**continue**

# Get the pdbid

**if** row[1] != '':

pdbid = row[1]

**else**:

**continue**

# Get the clustername from the Cluster Name column

**if** row[2] != '':

cluster = row[2]

# But maybe this structure wasn't located to a supercluster

# In that case, get it from the subcluster column.

**else**:

cluster = row[3]

pdbid\_clusterid.append((pdbid, cluster))

with open('ezbeta clusters/clusters.csv', 'wb') as o:

csv.writer(o).writerows([[cluster] \

**for** pdbid, cluster **in** pdbid\_clusterid])

The resulting file has 28 lines - exactly the right number, because the original figure is labeled 1 to 35, and I removed 7 lines.

I then ran this code to download, from HHOMP, every cluster named in "~~ezbeta~~ clusters/clusters.csv"

**import** urllib

**import** re

**import** csv

**class** UnrecognizedClusterName(Exception):

**pass**

url\_base = 'http://toolkit.tuebingen.mpg.de/hhomp\_ali/'

files = list()

**try**:

f = open('clusters.csv', 'r')

**for** line **in** csv.reader(f):

**if** len(line) == 0 **or** line[0] == '':

**continue**

**elif** re.match('cluster\d+', line[0]) **is** **not** None:

files.append(line[0] + '.clu')

**elif** re.match('OMP\..+?\.\d+\.\d+', line[0]) **is** **not** None\

**or** re.match('NodT\.\d', line[0]) **is** **not** None\

**or** re.match('TolC\.\d', line[0]) **is** **not** None:

files.append(line[0] + '.clu')

**elif** re.match('.+?\.\d+\.\d+', line[0]) **is** **not** None:

files.append('OMP.' + line[0] + '.clu')

**else**:

**raise** UnrecognizedClusterName(line[0])

**finally**:

f.close()

**for** filename **in** files:

urllib.urlretrieve(url\_base + filename, filename=filename)

**print**('done')

The code is saved as "~~ezbeta~~ clusters/download.py". However, I often modify files containing python code; the above code, printed in this log, should be referred to for any questions about my methods.

I copied all the .pdb files from "pymol/structures" into a folder called "ezbeta aligned structures"