February 1, 2012

downloaded" OMP.12.6.1.clu" from HHOMP: the complete alignment for the "OMPLA-like cluster"

Added the matrix "identity.csv", which was hand-made. It is not strictly an identity matrix, because it gives .05 for a match between X and V, for example, and .5 for B and N, for example.

Created "sorted 12.6.1.clu" using the following code, titled "order alignment.py" in today's commit:

from \_\_future\_\_ import division

from sundries import CIDict

import csv

from Bio import AlignIO

from Bio import SeqIO

from Bio.Align import MultipleSeqAlignment

# Uses CIDict

def retrieve\_matrix(csvfile):

# Create a dictionary mapping column numbers to resns

colnum\_to\_resn = dict()

# The first line is going to have an empty space as its first entry,

# so start at the second entry

for index, entry in enumerate(csvfile.next()[1:]):

colnum\_to\_resn.update({index: entry})

# Create a dictionary so dict['A']['V'] will give you score for

# alanine mutating into valine

first\_to\_second = CIDict([(resn, CIDict())

for resn in colnum\_to\_resn.values()])

for line in csvfile:

second\_resn = line[0]

for colnum, number in enumerate(line[1:]):

first\_resn = colnum\_to\_resn[colnum]

# Add number as floating point so I can sum these things up

# to get distances

first\_to\_second[first\_resn].update({second\_resn: float(number)})

return first\_to\_second

# Doesn't need any modules

def distance(seq1, seq2, matrix):

'''Get a distance score using a given matrix for a pair of aligned sequences

Or it might be a similarity score, depending on the matrix.

'''

if len(seq1) != len(seq2):

raise ValueError('Sequences of different length, must compare ' + \

'aligned sequences')

distance = 0

for pos in range(len(seq1)):

# These matrices are not necessarily symmetric, order matters

distance += matrix[seq1[pos]][seq2[pos]]

return distance

# Uses future division

def compare(seq1, seq2, matrix):

'''Comparison function I made just for this script, divides a distance

by number of positions that are not gaps in both of the sequences compared

to return a normalized distance'''

count = 0

for pos in range(len(seq1)):

if seq1[pos] != '-' or seq2[pos] != '-':

count += 1

return distance(seq1, seq2, matrix) / count

# Load matrix.

with open('identity.csv', 'rb') as f:

freader = csv.reader(f)

identity = (retrieve\_matrix(freader))

# Load multiple sequence alignment.

ompla\_msa = AlignIO.read('OMP.12.6.1.clu', 'clustal')

ompla\_pdb = ompla\_msa[5]

# I don't know what the first five entries in the multiple seq alignment

# represent, but I don't think they're actual protein sequences

ompla\_msa = ompla\_msa[5:]

# Create sorted list of alignments, with nearest to pdb sequence on

# top and furthest on bottom.

sorted\_seqs = sorted(ompla\_msa,

key=lambda seq: compare(ompla\_pdb,

seq, identity))[::-1]

# Create sorted multiple sequence alignment from list.

sorted\_msa = MultipleSeqAlignment(sorted\_seqs)

# Write the sorted alignment to a file.

AlignIO.write(sorted\_msa,'sorted 12.6.1.clu','clustal')

Retrieved 1qd5.pdb using PyMOL 1.40's fetch command

Using PyMOL's cartoon view of 1qd5 to determine where the beta strands are, examined the multiple sequence alignment. Usually only recorded first insertion or deletion.

Periplasmic tail before first strand is highly variable

First beta strand TNYLIYTQT: insertion on 9th sequence in alignment, many after the 9th sequence have the same insertion

Second strand DEVKFQLSLAFPLW: no insertions or deletions

Third strand SVLGASYTQKSWW: point deletion after like 25-30 sequences.

Fourth strand FRETNYEPQLFLGFATDYR: sequence 8 has a point deletion near the end. Like 50 sequences before you get to a two residue deletion in the more conserved membraney part.

- proline right in the middle of the strand. In FRET, RE are both interior. Sharp bend at FATD.

- the conservation characteristic of strands drops off sharply over GFATD and stays low

So not much of insertions or deletions in the interface region. What about the sides?

Strand six, the green-yellow strand, SWNRLYTRLMAEN: No insertions of deletions until a sequence in the 60s, which has an insertion like 30 or 40 amino acids long

Strand seven, WLVEVKPWYV: insertion about 40 sequences in.

- Another proline right in the middle of a strand, I thought that didn't happen much... is that true in beta barrels?

February 3, 2012

After running "order alignment.py", ran the code

for i in sorted\_msa:

print compare(ompla\_pdb, i, identity)

to produce "ompla identity scores.txt"

I examined a histogram of this data and observed that most of the sequences had identities below 40%