This folder contains work towards determining which BBTM matrices ot use at which sequence identities. It very much overlaps with the work in "deriving bbtm matrices" , "automated sequence alignment", and "ziff module": these four folders are all part of my effort to create and verify the usefulness of alignments for the hmomology modeling that goes into deriving Ezβ assymmetric.

# August 29, 2012

Downloaded TMout.zip from an e-mail David jimenez-Morales sent me on August 23. This is the same TMout folder that is seen in the "deriving bbtm matrices" folder and "automated sequence alignment" folder.

Here's a quote from the 1994 ClustalW paper:

“The ranges of distances and tables used with the PAM series of matrices are: 80-100% :PAM20, 60-80%:PAM60, 40-60%:PAM120, 0-40%:PAM350. The range used with the BLOSUM series is: 80-100%: BLOSUM80, 60-80% :BLOSUM62, 30-60% :BLOSUM45, 0-30%:BLOSUM30.”

Which makes me wonder: what expected seq identities do those PAM and BLOSUM matrices correspond to? How do they choose a representative matrix from the range?

I wrote some code that, given a PAM or BBTM distance, would give the expected sequence identity at that distance:

***import numpy as np***

***import scipy***

***import matrices***

***class MatrixMapping(dict):***

***'''A dictionary such that if x is a MatrixMapping, (n\*x)[key] == n\*(x[key])***

***'''***

***def \_\_rmul\_\_(self, other):***

***output = copy.deepcopy(self)***

***for key in output.keys():***

***output[key] \*= other***

***return output***

***def \_\_lmul\_\_(self, other):***

***return self.\_\_rmul\_\_(self, other)***

***published\_bbtm\_ordering = ['A', 'R', 'N', 'D', 'C', 'Q', 'E', 'G', 'H',***

***'I', 'L', 'K', 'M', 'F', 'P', 'S', 'T', 'W',***

***'Y', 'V']***

***# Background frequencies, e-mailed to me by David Jimenez-Morales***

***# on August 20 2012. I consider it a birthday present***

***pi\_out = {'A': 0.103414, 'C': 0.000253, 'E': 0.003965, 'D': 0.010899,***

***'G': 0.071558, 'F': 0.088656, 'I': 0.064497, 'H': 0.016882,***

***'K': 0.009315, 'M': 0.018249, 'L': 0.168981, 'N': 0.016985,***

***'Q': 0.023042, 'P': 0.018898, 'S': 0.025996, 'R': 0.012083,***

***'T': 0.050352, 'W': 0.045422, 'V': 0.115135, 'Y': 0.135606}***

***# Amino acid frequencies from http://www.tiem.utk.edu/~gross/bioed/webmodules/aminoacid.htm retrieved August 27 2012***

***pi\_ver = dict({'A': 7.4e-2,***

***'R': 4.2e-2,***

***'N': 4.4e-2,***

***'D': 5.9e-2,***

***'C': 3.3e-2,***

***'E': 5.8e-2,***

***'Q': 3.7e-2,***

***'G': 7.4e-2,***

***'H': 2.9e-2,***

***'I': 3.8e-2,***

***'L': 7.6e-2,***

***'K': 7.2e-2,***

***'M': 1.8e-2,***

***'F': 4.0e-2,***

***'P': 5.0e-2,***

***'S': 8.1e-2,***

***'T': 6.2e-2,***

***'W': 1.3e-2,***

***'Y': 3.3e-2,***

***'V': 6.8e-2})***

***pam1 = 10\*\*-4 \* matrices.parse('pam1.txt')***

***def parse\_david(path):***

***'''Open the matlab format matrix files that David Jiminez-Morales***

***sent me (in the "pout" folder)'''***

***with open(path, 'r') as f:***

***output = MatrixMapping()***

***for row\_resn, line in zip(published\_bbtm\_ordering, f):***

***for col\_resn, entry in zip(published\_bbtm\_ordering,***

***line.split()):***

***output.update({(row\_resn, col\_resn): float(entry)})***

***return output***

***def p\_retrieve(t):***

***'''Return one of David Jimenez-Morales's transition probability***

***matrices, corresponding to the given time t, from the matrices***

***he sent me on August 23 2012.'''***

***return ~~matrices.parseparse\_david~~('TMout/pout/MTMout{0}.p'.format(t))***

***def id\_given\_time(t, mat\_name='bbtm'):***

***'''Given a time and the name of a matrix family (bbtm and pam***

***currently available) return the expected value of %identity between***

***a sequence and the same sequence after a time t has passed,***

***given that the amino acid frequencies in the original sequence***

***are equal to the background frequencies'''***

***if mat\_name == 'pam':***

***pam\_at\_t = matrices.matrix\_power(pam1, t)***

***return 1 - matrices.expected\_changes(pam\_at\_t, pi\_ver)***

***if mat == 'bbtm':***

***bbtm\_at\_t = p\_retrieve(t)***

***return 1 - matrices.expected\_changes(bbtm\_at\_t, pi\_out)***

(fixed an error in p\_retrieve, which was not actually used for the PAM results)

It's the last function that a user of this code actually runs, to find an expected sequence identity.

Pam20, 80% to 100%:

***>>> id\_given\_time(20, mat\_name='pam')***

***0.81964237460071687***

Pam60, 60% to 80%:

***>>> id\_given\_time(60, mat\_name='pam')***

***0.57486688878682646***

Pam120, 40% to 60%:

***>>> id\_given\_time(120, mat\_name='pam')***

***0.37126323726146515***

Pam350, 0% to 40%:

***>>> id\_given\_time(350, mat\_name='pam')***

***0.13774262286766792***

I am confused, because the expected identities are *outside the ranges for which they are meant to be used. Maybe they use different amino acid frequencies? The ones I'm using are the vertebrate amino acid frequencies. PAM is used for plant, bacterial, everything.*

*I don't care a whole lot. This was an interesting exercise and I'm glad I got to see and interpret the results, but the importance of this was really to put me in a good position to find the correct BBTM matrices for a variety of ranges of sequence identity, and I think I can do that now.*