March 5, 2012

"figures notes.docx" incidicates that "inclusive\_moment.csv" and "exclusive\_moment.csv" were produced with "generate moments.py". In this file are the lines of code

with open('beta\_selections.csv', 'rb') as f:

reader = csv.reader(f)

beta\_selections = ezb.selections\_by\_resi(reader)

exclusive\_selections = CIDict()

for name, selection in beta\_selections.items():

exclusive\_selections.update(((name, list(set(selection).intersection(

set(inclusive\_selections[name])))),))

and also

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

exc\_centers = ezb.load\_centers(csv.reader(f))

and also

moments('exclusive\_moment.csv',exclusive\_selections, exc\_centers)

In other words, "exclusive\_moments.csv" was made using "beta\_selections.csv" as the selection and "exc centers.csv" as the centers.

This makes sense only if "exc centers.csv" are the centers for "beta\_selections.csv": that is, for each protein, if all the atom positions in a protein are projected onto the plane of the membrane, and a unit vector is drawn from the center in "exc centers.csv" to the c-alpha of every residue listed for that protein in "beta\_selections.csv", the sum of these unit vectors is the zero vector.

"figures notes.docx" tells how "beta\_selections.csv" was created, as follows:

*Made beta\_selections.csv using a function in selectors.py that exports a selection. I ran it with the parameter ".exp\_sheets & n. ca".*

That selection stands for "exposed sheets". The code that generates this selection, in "./pymol/ezbeta\_api.py", is

cmd.select(name + '.exp\_sheets', name + ' & ' + \

selectors.percent\_sasa\_and\_ss(group.ez\_data,.2,'E'))

In other words, this selection has greater than .2 change in SASA and is in a beta sheet as determined by DSSP.

This is disturbing, because it gives no indication that core residues were removed. The possibility comes to mind that core residues are not in beta sheets, and thus not in the ".exp\_sheets" selection, but this is not the case, as attested to by "exc centers/exc centers.odt", which reads

*I used the 'take\_intersection' and 'make\_exp\_sheets' functions in 'make selections and coordlists.py' to find the intersection of the exp\_sheets selection and the cored\_1 selection, from 'cored\_1.csv'. I verified that the intersection was smaller than the exp\_sheets selection for 3EFM:*

*PyMOL>print len(exp\_sheets['3EFM'])*

*147*

*PyMOL>print len(intersection['3EFM'])*

*145*

"make\_exp\_sheets" must mean "get\_exp\_sheets", as there is no function in that file called "make\_exp\_sheets". "get\_exp\_sheets" is the following code:

def get\_exp\_sheets():

stored.exp\_sheets = dict()

for name in groupdict.keys():

stored.exp\_sheets.update(((name, list()),))

cmd.iterate(name + '.exp\_sheets & n. CA',

'stored.exp\_sheets["%s"].append(resi)' % name)

return stored.exp\_sheets

The important thing to note is that it also uses the ".exp\_sheets" selection.

All this establishes that the centers used to generate "exclusive\_moment.csv" are the centers of a selection that includes fewer amino acids than the selection used to generate "exclusive\_moment.csv". This could have severe effects on the moments generated. For any protein that had core residues removed, there is no reason to expect the moment in "exclusive\_moments.csv" to be accurate, since it uses the wrong centers.

Actually, further examination of my really poorly commented code leads me to revise that judgment. Generate moments.py takes the intersection of the selection in "beta\_selections.csv" with the selection being used for the inclusive moment, which is created using the code

with open('cored 1 selections with 1qd5.csv', 'rb') as f:

reader = csv.reader(f)

inclusive\_selections = ezb.selections\_by\_resi(reader)

The name of that file indicates that it is simply the selections in "cored 1.csv", used to create the selections that were used to create the centers. This is a strange and roundabout way of doing things, but it is entirely correct; it would result in generating each moment using a selection and the center of that selection. I recall that I was sleeping about three hours a night at the time I wrote the code to generate the exclusive moments.

To be sure, I'm going to regenerate them using the selections using the selections used to generate " exc centers.csv", which is "exc centers/exc selections.csv"

Ran the following code. It was copied and pasted from "generated moments.csv", and then modified.

import sys

sys.path.append(r'C:\cygwin\home\alex\beta barrels\non normalized moments\calculator modules')

from os import chdir

from sundries import one\_letter

from useful import CIDict

import csv

import warnings

import numpy as np

from Bio.PDB import PDBParser

from biodata import file\_dict

import ezb

reload(ezb)

#chdir('/home/davis/Desktop/work stuff/final paper/figures')

#chdir(r'C:\Users\Nanda Lab\Desktop\Alex\final paper\figures')

# Use Biopython's pdb file parser to create a dictionary called

# "structures" containing Biopython's pdb structure objects,

# created from the files in the folder "structures with 1qd5"

structure\_files = file\_dict('structures with 1qd5', ['aligned\_(.\*).pdb'])

parser = PDBParser()

# Daniel's aligned pdb files are missing b factors, and perhaps

# some other information, which causes Biopython to give hundreds

# or thousands of warnings when loading them. The following code

# suppresses warnings

with warnings.catch\_warnings():

warnings.simplefilter('ignore')

structures = [(name, parser.get\_structure(name, path)) \

for name, path in structure\_files.items()]

structures = CIDict(structures)

# Load the selections that should have been used

with open('exc centers/exc selections.csv', 'rb') as f:

reader = csv.reader(f)

beta\_selections = ezb.selections\_by\_resi(reader)

right\_selections = CIDict()

for name, selection in beta\_selections.items():

right\_selections.update(((name, selection),))

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

exc\_centers = ezb.load\_centers(csv.reader(f))

def moments(path, selections, centers):

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

reader = csv.reader(f)

new\_calc = ezb.Calculator(reader, normalize = True)

moments = CIDict()

for name in structures.keys():

if name.upper() == '1QD5':

continue

moments.update({name:

ezb.moment(structures[name], selections[name],

centers[name], new\_calc,

paramless\_option = '.5',

old\_style\_gly = True) })

with open(path, 'wb') as f:

target = csv.writer(f)

for name, moment in moments.items():

row = [name.upper()] + [str(component) for component in moment]

target.writerow(row)

moments('right selection exc moments.csv',right\_selections, exc\_centers)

print('done')

The resulting file, "right selection exc moments.csv", is absolutely identical to "exclusive\_moment.csv"