**# For coloring “all atoms” according to a new vector (deltaE\_i used here); all atoms are obtained from the variable “order” in pdb file in MATLAB; check mapping\_crystal\_structure\_pymol.m for constructing the vector.**

cmd.load("4mwf\_cC\_only.pdb")

set\_view (\

-0.021376820, 0.326919585, 0.944810331,\

0.901856959, -0.401571989, 0.159355357,\

0.431505799, 0.855490267, -0.286250412,\

0.000000000, 0.000000000, -155.433761597,\

55.899662018, -11.721086502, 39.145759583,\

125.366844177, 185.500671387, -20.000000000 )

hide all

show cartoon

bg\_color white

#seq\_view on #not sure what the command for this is

*# open the file of new values (just 1 column of numbers, one for each alpha carbon)*

inFile = open("dE2\_crystal\_structure\_4mwf\_allatoms.txt", 'r')

*# create the global, stored array*

stored.newB = []

*# read the new B factors from file*

for line in inFile.readlines(): stored.newB.append( float(line) )

*# close the input file*

inFile.close()

*# clear out the old B Factors*

alter 4mwf\_cC\_only, b=0.0

*# update the B Factors with new properties*

alter 4mwf\_cC\_only, b=stored.newB.pop(0)

*# after running color\_b.py from pymol*

run color\_b.py

***# coloring (not used)***

# color\_b gradient=bwr, nbins=20, sat=1, minimum=0, maximum=12

***#coloring with a user defined gradient [blue white red (easy escape pathways = blue and high cost = red); colors from brewermap]***

color\_b gradient=user, nbins=100, user\_rgb = (0.21569,0.49412,0.72157, 1,1,1, 0.89412,0.10196,0.1098), sat=1, minimum=0, maximum=11

**#highlight pockets and caviFes by depth-dependent shadowing**

util.ray\_shadows('occlusion2')

**#Saving figure in png format**

png 4mfw\_all, height = 4 cm, width = 4 cm, dpi=300, ray = 1