**Epitopia Prep**

Filename: epitopia\_prep.py

Epitopia Prep v2.1 Readme

Author: Colin Welsh

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Purpose: Adds SEQRES header to .pdb files that are required by Epitopia. Renumbers

protein sequence if necessary.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Requirements: Python 3.4.3 or later

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

To use: Place epitopia\_prep.py into the directory containing your .pdb file you wish to submit to Epitopia and run the file via the terminal.

To run from the terminal, open the terminal and enter:

cd \*directory of .pdb file here\*

example: Desktop/Research\ Program/PLA2R\_models/

Then enter:

python3 epitopia\_prep.py

The program will run, prompting you to enter the filename of the .pdb file to be converted. NOTE: You must include ".pdb" when entering the filename. The script will then output a file with the name epitopia\_\*filename\*.pdb.

'''

\*\*\*\* Title: Epitopia Prep v2.1

\*\*\*\* Author: Colin Welsh

\*\*\*\* Date: 12 June 2017

\*\*\*\* Purpose: Converts a minimized file from Chimera into a file that is compatible

\*\*\*\* for use with epitopia.

'''

import os

### Parameters: infileName, the name of the .pdb file to be converted

###

### Returns: outfileName, the name of the temp file with no repeating sequence nums

###

### Purpose: changes the loc numbers of the amino acids if there is overlap

def pre\_check(infileName):

infile = open(infileName, 'r')

outfileName = "tempfile\_" + infileName

outfile = open(outfileName, 'w')

prevNum = 'ABC'

cnt = 0

for line in infile:

if line.startswith('ATOM'):

num = line[23:26]

if num != prevNum:

cnt += 1

if cnt < 10:

fixedNum = " " + str(cnt)

elif cnt >= 10 and cnt < 100:

fixedNum = " " + str(cnt)

else:

fixedNum = str(cnt)

prevNum = num

outfile.write(line[:23])

outfile.write(fixedNum)

outfile.write(line[26:])

outfile.write('\n')

outfile.close()

return outfileName

### Parameters: infileName, name of the .pdb file to be converted

###

### Returns: aminoList, a dictionary of the amino acids matched with their loc number

### largest, the largest loc number

###

### Purpose: creates a dictionary to be used in creating the file header

def make\_amino\_list(infileName):

infile = open(infileName,'r')

aminoList = {}

largest = 0

for line in infile:

if line.startswith('ATOM'):

loc = line[22:26]

loc = loc.lstrip()

loc = int(loc)

aminoAcid = line[17:20]

if loc > largest:

largest = loc

if loc not in aminoList:

aminoList[loc] = aminoAcid

infile.close()

return aminoList,largest

### Parameters: outfile, the file to be submitted to Epitopia

### aminoList, a dictionary containing all unique AA/loc numbers from original .pdb

### largest, largest loc number in original .pdb

###

### Returns: None

###

### Purpose: writes the file header required by Epitopia

def write\_seqres(outfile,aminoList,largest):

k = 1

if largest % 13 == 0:

numRows = largest // 13

else:

numRows = (largest // 13) + 1

if largest < 10:

spacer = " "

elif largest > 9 and largest < 100:

spacer = ' '

elif largest > 99 and largest < 1000:

spacer = ' '

if largest > 999:

spacer = ' '

for i in range(1,numRows+1,1):

multiplier = (i - 1) \* 13

if i < 10:

header = "SEQRES " + str(i) + " A" + spacer + str(largest) + " "

elif i >= 10 and i < 100:

header = "SEQRES " + str(i) + " A" + spacer + str(largest) + " "

else:

header = "SEQRES " + str(i) + " A" + spacer + str(largest) + " "

outfile.write(header)

for k in range(1,14,1):

loc = k + multiplier

if loc in aminoList:

amino = aminoList[loc]

outfile.write(amino)

outfile.write(' ')

outfile.write('\n')

### Parameters: outfile, the file to be submitted to epitopia

### infileName, the name of the file that is the original .pdb (or pre-checked file, if

### appropriate)

###

### Returns: none

###

### Purpose: writes the rest of the original .pdb

def write\_bulk(outfile,infileName):

infile = open(infileName,'r')

for line in infile:

if line.startswith('ATOM') or line.startswith('END'):

outfile.write(line)

def main():

infileName = input('Enter file name to be prepped for epitopia (include .pdb): ')

outfileName = "epitopia\_" + infileName

outfile = open(outfileName,'w')

doPreCheck = input('Precheck files? (Y/N): ')

if doPreCheck == 'Y' or doPreCheck == 'y':

infileName = pre\_check(infileName)

aminoList,largest = make\_amino\_list(infileName)

write\_seqres(outfile,aminoList,largest)

write\_bulk(outfile,infileName)

outfile.close()

if doPreCheck == 'Y' or doPreCheck == 'y':

os.remove(infileName)

main()

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ROSIE Prep**

Filename: rosie\_prep.py

ROSIE Preparer v1.1 Readme

Author: Colin Welsh

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Purpose: Removes unnecessary information from .pdb files that result in ROSIE errors and

correctly formats files for use by ROSIE.

Proper file format includes:

\* Only ATOM, TER, and END lines

\* TER lines between chains

\* Appropriate length lines that seem to reduce the probability of ROSIE returning

an error.

NOTE: This script does NOT properly name your chains. You must do that manually via

Chimera.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Requirements: Python 3.4.3 or later

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

To use: Place rosie\_prep.py into the directory containing your .pdb file you wish to run on ROSIE and run the file via the terminal.

To run from the terminal, open the terminal and enter:

cd \*directory of zdock files here\*

example: Desktop/top\_preds/

Then enter:

python3 rosie\_prep.py

The program will run, prompting you to enter the filename of the .pdb file to be converted. NOTE: You must include ".pdb" when entering the filename. The script will then output a file called proteins.pdb that is ready to be used with ROSIE.

NOTE: The script will overwrite any file called "proteins.pdb" that is preexisting in its directory. Make sure that there is no file called "proteins.pdb" where you place the script.

'''

\*\*\* ROSIE File Preparer v1.1

\*\*\* Author: Colin Welsh

\*\*\* Date Created: 14 June 2017

'''

def main():

filename = input("Enter filename to be prepared: ")

infile = open(filename,'r')

outfile = open("proteins.pdb",'w')

for line in infile:

if line.startswith('ATOM'):

outfile.write(line[:55])

outfile.write('\n')

elif line.startswith('TER'):

outfile.write(line[:3])

outfile.write('\n')

elif line.startswith('ENDMDL'):

outfile.write('TER\n')

outfile.write('END')

infile.close()

outfile.close()

main()

**ZDOCK Fix**

Filename: epitopia\_prep.py

zdock\_fix v1.1 Readme

Author: Colin Welsh

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Purpose: Removes unnecessary information from files from ZDock that prevent the file

from being opened in Chimera.

Requirements: Python 3.4.3 or later

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

To use: Place zdock\_fix.py into the directory containing the 10 prediction files from ZDock (complex.1.pdb; complex.2.pdb) and run the file via the terminal.

To do this, open the terminal and enter:

cd \*directory of zdock files here\*

example: Desktop/top\_preds/

Then enter:

python3 zdock\_fix.py

The program will run, first prompting you to enter an antibody name (ex. PLA2R) and then a cap name, the 4 character identifier from PDB (ex. 5IMK). The program will run and output ten new files that are compatible with Chimera. The program will also output the proteins.pdb files required by ROSIE in a separate folder in the same directory.

'''

\*\*\* ZDOCK Fix v1.1

\*\*\* Author: Colin Welsh

\*\*\* Date Created: 19 June 2017

'''

import os

### Purpose: Removes information from each line that prevents .pdb file from being opened in Chimera

### Parameters: filePrefixAntigen, the the name of the antigen;

### filePrefixCap, the name of the cap

### Returns: outNames, a list of the outputted .pdb files that are compatible with Chimera

def zdock\_to\_chimera(filePrefixAntigen, filePrefixCap):

cnt = 1

fileNames = []

outNames = []

for i in range(1,11,1):

name = "complex." + str(i) + ".pdb"

fileNames.append(name)

for obj in fileNames:

infile = open(obj, 'r')

outName = "zdock\_" + filePrefixAntigen + "\_" + filePrefixCap + "\_complex" + str(cnt) + ".pdb"

outNames.append(outName)

outfile = open(outName, 'w')

cnt += 1

for line in infile:

line = line.rstrip('\n')

info = line[0:55]

outfile.write(info)

outfile.write('\n')

infile.close()

outfile.close()

return outNames

### Purpose: Creates the file directory in which proteins.pdb files are placed

### Parameters: dirList, the list of directories to create

### Returns: None

def make\_dir(dirList):

cnt = 0

for obj in dirList:

os.makedirs(dirList[cnt])

cnt += 1

### Purpose: Creates the list of directories to be created for the proteins.pdb files

### Parameters: fileRoot, the location of zdock\_fix.py;

### fileNames, the list of Chimera-compatible .pdb files;

### antiName, the name of the Antigen

### capName, the name of the protein cap;

### Returns: dirList, the list of directories to be created

def make\_dir\_list(fileRoot, fileNames, antiName, capName):

dirList = []

cnt = 0

newFolder = antiName + '\_' + capName + '\_ROSIE\_files'

for obj in fileNames:

newDir = os.path.join(fileRoot,newFolder,fileNames[cnt].rstrip('.pdb'))

dirList.append(newDir)

cnt += 1

return dirList

### Purpose: Converts Chimera-ready .pdb files into the appropriate format for ROSIE

### Parameters: dirList, the list of the directories to place proteins.pdb files in

### fileNames, list of files to be converted into the appropriate ROSIE format

### Returns: None

def convert\_to\_rosie(dirList,fileNames):

cnt = 0

for obj in fileNames:

infile = open(fileNames[cnt],'r')

outName = os.path.join(dirList[cnt],'proteins.pdb')

outfile = open(outName,'w')

prevNum = 0

for line in infile:

line = line.rstrip('\n')

num = line[23:26]

num = int(num.lstrip(' '))

if num < prevNum:

outfile.write('TER\n')

if line.startswith('ATOM'):

if line[21] != 'A' or line[21] != 'C':

outfile.write(line[:21])

outfile.write('C')

outfile.write(line[22:])

outfile.write('\n')

else:

outfile.write(line[:55])

outfile.write('\n')

elif line.startswith('TER'):

outfile.write(line[:2])

outfile.write('\n')

elif line.startswith('ENDMDL'):

outfile.write('TER\n')

prevNum = num

outfile.write('END')

infile.close()

outfile.close()

cnt += 1

def main():

fileRoot = os.getcwd()

print(fileRoot)

antiName = input("Enter antigen name: ")

capName = input("Enter cap name: ")

fileNames = zdock\_to\_chimera(antiName, capName)

dirList = make\_dir\_list(fileRoot, fileNames, antiName, capName)

make\_dir(dirList)

convert\_to\_rosie(dirList,fileNames)

main()

**ZDOCK to Chimera**

Filename: zdock\_to\_chimera.py

ZDock to Chimera v1.0 Readme

Author: Colin Welsh

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Purpose: Removes unnecessary information from files from ZDock that prevent the file

from being opened in Chimera.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Requirements: Python 3.4.3 or later

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

To use: Place zdock\_to\_chimera.py into the directory containing the 10 prediction files from ZDock (complex.1.pdb; complex.2.pdb) and run the file via the terminal.

To do this, open the terminal and enter:

cd \*directory of zdock files here\*

example: Desktop/top\_preds/

Then enter:

python3 zdock\_to\_chimera.py

The program will run, first prompting you to enter an antibody name (ex. PLA2R) and then a cap name, the 4 character identifier from PDB (ex. 5IMK). The program will run and output ten new files that are compatible with Chimera.

'''

\*\*\* ZDock to Chimera Converter v1.0

\*\*\* Author: Colin Welsh

\*\*\* Date Created: 15 February 2017

'''

def main():

filenames = []

filePrefixAntigen = input("Enter Antigen name: ")

filePrefixCap = input("Enter Cap name: ")

for i in range(1,11,1):

name = "complex." + str(i) + ".pdb"

filenames.append(name)

for obj in filenames:

infile = open(obj, 'r')

outName = "zdock\_" + filePrefixAntigen + "\_" + filePrefixCap + "\_complex" + str(cnt) + ".pdb"

outfile = open(outName, 'w')

for line in infile:

line = line.rstrip('\n')

info = line[0:55]

outfile.write(info)

outfile.write('\n')

infile.close()

outfile.close()

main()