by D.M. at EML Research
domantas.motiejunas@eml-r.villa-bosch.de
V0.01 2005.12.16
V0.02 2006.01.30

- Added scoring part. Now centers of protein 1 and protein 2 are read from fort.55 not calculated from the coordinates since in trimmed docking case the centers have to be calculated from trimmed structure to be exact.

V0.03 2006.02.07

- Adapted everything to new fort.55 format again now with 17 columns stdDeviation of energy at the last column is considered
- Now programs give error and exits if it can't write or can't find data files 'clust.dat', 'score.dat'
- There was a problem when there were two clusters with the same sizes, representative recovery and printing out of clusters in analysis mode was mixing the places of these cluster, now comparison of cluster size method is improved and this should not be a problem any more
- Added stdDeviation for rp scores
- fixed nan output when occurrences were 0

WHAT FOR

This program can be used to process SDA docking results in fort.55 file and do some scoring

1. Clustering
   Hierarchical clustering of number of docked complexed. It starts by assigning all docked proteins to separate clusters and proceeds by merging the closest clusters until two clusters are left. Average linkage distance calculation is used to evaluate distance between clusters. (That is distance between centers of clusters).
2. Scoring
   Electrostatic energy, occurrences are extracted from fort.55 file, backbone rmsd to reference structure is calculated*, Residue Propensities are calculated for docked proteins, all this info is stored for further analysis.
   * NOTE currently I just take protein 1 and protein 2 which are used in docking and in case were X-ray structure is available these starting unbound should be superimposed onto X-ray structure of the complex to get relevant rmsd. To be more precise one should use the the bound structure of protein 2 to evaluate the quality of docking by calculating the rmsd. (on the other hand this is maybe not so important after the stage of rigid-body protein docking).
3. Analysis
   a. from stored clustering results the formation of the clusters at each cycle can be monitored, this can be used to identify clustering cycles where the most distinct clusters where formed, and what is the distance between closest clusters at particular clustering cycle, based on this information user can decide which clustering cycle to use for further analysis.
   b. information about clusters at particular clustering cycle can be printed including all scoring data
4. Recover
   Recovers representatives of clusters to pdb files, at particular clustering cycle. Representative of the cluster is a structure in the middle of the cluster.

YOU NEED

fort.55	file	main SDA output file
p1.pdb	pdb	file of the first protein used in docking
p2.pdb	pdb	file of the second protein used in docking

HOW

The program can be run in four modes: clustering, scoring, analysis, recover (representatives).
In most cases you also need to indicate files necessary for the program.
See options below:

FILE options - how to specify which files to use.

-f55 <string>	fort.55 file
-p1 <string>	protein 1 file
-p2 <string>	protein 2 file
-clo [string]	file to store clustering data, DEFAULT 'clust.dat'
-cli [string]	file to read clustering data from, DEFAULT 'clust.dat'

CLUSTERING options - how to do clustering

-cl	switch to run in clustering mode
-h	print help for clustering mode
-n [int]	number of docked structures to cluster from fort.55, DEFAULT is 500

NOTES
If not specified by '-n' option, by default 500 structures are taken from fort.55 file, or if there are less then 500 structures, all are taken for clustering

EXAMPLE
./program -cl -n 300 -f55 fort.55 -p1 p1.pdb -p2 p2.pdb

300 docked proteins from fort.55 will be clustered and clustering results will be stored in 'clust.dat' file (default). fort.55, p1.pdb and p2.pdb files are loaded from current directory.

SCORING options – hot to do scoring

-sc	switch to scoring mode
-h	print help for scoring mode
-rpin <string>	option to specify residue propensity file, any number of RP files can be provide each preceded by this
-cdist [float]	contact distance between residues for residue propensity calculation, DEFAULT is 5.0 A
-rpdef [int]	definition for residue propensities calculation, DEFAULT is 1 Available options: 1 residues are in contact if any of their atoms are within -cdist distance 2 the same like 1 but backbone atoms are excluded (therefore GLY residue is not relevent) 3 only CB atoms (CA for GLY) are used to check if they are within -cdist distance 4 like 1 but specified residues are excluded from RP calculations 5 like 2 but specified residues are excluded from RP calculations 6 when trimmed residues are specified only backbone + CB atoms are used for them int RP calculations
-add [float]	this probably should not be changed DEFAULT is 1.0

NOTES:
Clustering should be done first, then scoring looks at the output file from clustering to see hoe many proteins were used for clustering and uses the same number of proteins for scoring. By default after scoring data is stored in binary score.dat file, and it is read in analysis mode.

EXAMPLE:
./program -sc -f55 fort.55 -p1 p1.pdb -p2 p2.pdb -cdist 6.0 -rpdef 6 -rpin rpFile1 -rpin rpFile2

does scoring of with contact distance 6.0, contact definition 6 and using two files of residue propensities

---

ANALYSIS options - how to analyze clustering results

-an	switch to analysis mode
-h	print help for analysis mode
-p	switch to printing mode
-cy [int]	print information on clustering in last specified number of clusters. DEFAULT is all cycles
-clcn <int>	print information on clusters at specified clustering cycle
-clcv <float>	print information on clusters at specified distance value. Distance value here is distance between closest clusters at particular clustering cycle
-clsz [int]	minimum size of cluster to print with -clcn and -clcv. DEFAULT is 5
-n [int]	restricts number of the first biggest clusters to show. DAFAULT is all
-v	visualization mode (not really useful)
-clcn <int>	approximate visualization of cluster sizes at clustering cycle number
-clcv <float>	approximate visualization of cluster sizes at clustering cycle where distance value

NOTE:
clustering data stored in 'clust.dat' by default is used if you store your clustering data in another file, specify it with -clin filename option. You may also need to specify fort.55 protein 1 and protein 2 files. Also see examples below!!

EXAMPLE 1
./program -an -p -cy

This prints some information about all clustering cycles. Columns mean:
1 - number of clusters
2 - number of cycle
3 - distance between closest clusters in this cycle
4 - % of distance increase from previous cycle. % is here from whole distance changed during the clustering.
5 -- size of the smallest cluster
6 - size of the largest clustere
7 - average size of clusters at this cycle

EXAMPLE 2
./program -an -p -clcn 100 -f55 fort.55 -p1 p1.pdb -p2 p2.pdb

This prints some information about clusters at cycle 100.
1 - number of cluster (sorted by size)
2 - cluster size
3 - cluster size with added occurrence from fort.55
4 - number of protein (number of fort.55 line) which is representative for this cluster
5 - energy of representive
6 - average energy of the cluster
7 - average distribution of energies in the cluster
8 - average rmsd of docked protein towards protein2 in this cluster
9 - RP score
10 - RP score stdDeviation
...

EXAMPLE 3
./program -an -v -clcn 100

This shows approximate sizes of the clusters as bars from '=' characters

---

RECOVER options - how to recover cluster representatives as pdb files

-re	switch to recover mode
-h	help for recover mode

-clcn <int>	recover cluster representatives from cycle number
-clcv <float>	recover cluster representatives from cycle at distance value
-n [int]	recover cluster representatives from first number of clusters DEFAULT 10
-clsz [int]	recover cluster representatives only if cluster size is bigger then this option DEFAULT 5

EXAMPLE

./program -re -clcn 100 -f55 fort.55 -p1 p1.pdb -p2 p2.pdb

This will recover representatives of clusters at cycle 100. Files will have names like cl<cluster number>.pdb

AUXILARY

Some scripts in ./aux folder:
makeGraphCycles.py - make a graph in png with gnuplot from -an -p -cy command output.

EXAMPLE

./program -an -p -cy > anpcy.out
./makeGraphCycles.py mygraph anpcy.out anpcy.png

makeGraphClusters.py - makes a grapht with gnuplot of cluster with and without occurrence
added form fort.55, this is from -an -p -clcn or -an -p -clcv
output.

EXAMPLE

./program -an -p -clcn 100 -f55 fort.tt -p1 p1.pdb -p2 p2.pdb > anpclcn_100.out
./makeGraphClusters.py mygraph anpclcn_100.out anpclcn_100.png