A_purva : User Manual

1) Introduction

A_purva is a Contact Map Overlap maximization (CMO) solver. Given two protein structures represented by two contact maps, A_purva computes the amino-acid alignment that maximizes the number of common contacts.
More information about the solver can be found in the following articles:

R. Andonov, N. Malod-Dognin and N. Yanev, "Maximum Contact Map Overlap Revisited", Journal of Computational Biology, vol 18(1), p.27-41, 2011.

R. Andonov, N. Yanev and N. Malod-Dognin, "An Efficient Lagrangian Relaxation for the Contact Map Overlap Problem", K.A. Crandall and J. Lagergren (Eds.) : WABI 2008, LNBI 5251, pp. 162-173, 2008".

2) Installation / Compilation

Unzip the archive "A_purva 1.x" in the folder of your choice. Then compile A_purva by typing "make". Note that A_purva run faster if the correct architecture is set in the Makefile (ex : -march=core2 for Intel core2 processors). You can safely remove temporary object files by typing "make clean".

3) Command line options

PDB file specific options:

Command	Mandatory	Description	Default value
--pdb1 <filename>	yes	First .pdb file (used as row)	
--pdb2 <filename>	yes	Second .pdb file (used as col)	
--chain1 <chain Id>	yes	Id of the chain that will be processed in the first pdb file	
--chain2 <chain Id>	yes	Id of the chain that will be processed in the second pdb file	
--dthr <value>	no	Distance threshold for generating contacts, in angstrom	7.5
--vmd <0|1>	no	If set to 1, generate visualisation file for VMD*	0

* VMD: Visual Molecular Dynamics, http://www.ks.uiuc.edu/Research/vmd/

Contact map specific options:

Command	Mandatory	Description	Default value
--cm1 <filename>	yes	First contact map file (used as row)	
--cm2 <filename>	yes	Second contact map file (used as col)	

General options:

Command	Mandatory	Description	Default value
--solventfilter	no	Amino-acid alignment filter based on solvent accessibility, as derived from contact density 0 → Do not use solvent accessibility filter 1 → Use solvent accessibility filter	0
--ssefilter <0|1|2>	no	Amino-acid alignment filters based on secondary structure elements (SSE) 0 → No SSE filter 1 → Matchings between alpha helices and beta-stands are prohibited 2 → Also prohibits matchings between secondary structure elements and loops	0

--hierarchical <0|1>	no	Hierarchical approach: first align the secondary structure elements, and second extend the alignment to the amino-acids 0 → Do not use the hierarchical approach 1 → Use the hierarchical approach, which	0
--sseonly <0|1>	no	If set to 1, A_purva only computes secondary structure alignment	0
--display <0|1|2>	no	Set the display level of A_purva 0 → Display only the comparison scores 1 → Comparison scores + statistics 2 → Comparison score + statistics + alignment	0
--nodelimit <value>	no	Maximum number of branch and bound nodes to be explored (0 → root node only)	10000000 nodes
--iterlimit <value>	no	Maximum number of iteration per Branch and Bound node	4000 iterations
--timelimit <value>	no	Maximum execution time (in seconds). Note that the time limit is checked after each iteration, and thus it is not very accurate.	1800 sec.

Examples:
./A_purva --cm1 ./1amkA.hcm --cm2 ./1aw2A.hcm --display 2
A_purva will align the two contact maps "1amkA.hcm" and "1aw2A.hcm", and will display the corresponding alignment.

./A_purva --pdb1 1amk.pdb --chain1 A --pdb2 1aw2.pdb --chain2 A --vmd 1
A_purva will align chain A of 1amk.pdb with chain A of 1aw2.pdb, and will generate two files for VMD that highlight the matching between the two protein chains.

4) PDB file restrictions

A_purva can process .PDB files (.pdb or .ent), with the following restrictions:
- 1) One chain per PDB file
- 2) The PDB files should not contain more than one model
- 3) The HETATM are not processed

5) Contact map format

A_purva can process contact maps, i.e. files describing the closeness relations between the amino-acids in each protein. A contact map file is divided in three sections, separated by an empty line.

- The first section describes the contacts between the amino-acids. The first line contains an integer value X, which is the number of residues. Note that residues are labelled starting from 1. The second line contains an integer value Y, which is the number of contacts. The next Y lines each contain two integer values i and j, i<j, the tail and head residues of each contact.

- The second section describes the secondary structure assignment of the protein. It start with a line containing an integer value S, which is the number of secondary structure elements (SSE). It is then followed by S lines, each containing two integers (i and j) and a symbol t, where i is the first amino-acid of the SSE, j is the last amino-acid of the SSE, and t is the type of the SSE (either H for an alpha-helix, or b for a beta-strand).

- The last section describes the contacts between the secondary structure elements. It starts with a line containing an integer value C, which is the number of contacts between the SSE. It is then followed by C lines, each containing three integer values i, j and k, corresponding to the contact between SSE i and the SSE j, i<j, and having a weight k, which is the number of amino-acid contacts between the two SSEs.

Examples of contact map files are given in the CM.zip archive. They correspond to the Skolnick set, and were

generated using the following parameters:

- Contact distance threshold between alpha carbons is 7.5A.
- Contacts between consecutive amino-acids (i and i+1) are not taken into account.
- Secondary structure assignments were computed with Kaksi

http://migale.jouy.inra.fr/outils/mig/kaksi/