INTER-PROTEIN CONTACT PREDICTION (iPatch) Package

Qiang LUO

mrqiangluo@gmail.com

2010-03-05

This program is used to predict the inter-protein (domain) contact sites by using the amino acid propensity, pair propensity, triangle propensity of the interface, as well as the structure information for each contacting partner. The main function is iPatch.m. For more information: http://sites.google.com/site/qluochina. This package is free of use, but the authors do not guarantee bug free of this package. Please feel free to contact the author at mrqiangluo@gmail.com with any comments, or suggestions. The primary reference for the present package is "*i-Patch: Inter-Protein Contact Prediction using Local Network Information*" (submitted), which should be cited whenever this toolbox is used.

1 REQUIRMENT of iPatch

Unix with PSA: this program needs to call *PSA* installed on Unix to calculate the solvent accessible surface area for each protein. The PSA program will be installed automatically, if you have installed JOY which can be found from the following link http://tardis.nibio.go.jp/joy/. If you want to make your own executable version of the PSA, the source code of this program can be found in the following link ftp://salilab.org/pub/miscellaneous.

BUG:

1) Some times the calling of PSA from Matlab fails with some unknown reason and the whole Matlab program will be terminated, please simply restart it and run it again.

2) For the NMR PDB file, please make sure only one model of coordinates is left in your input PDB file, since this package is not currently dealing with multiple models. (All the other models that you don't need can be simply deleted from the PDB, and also delete the line of "MODEL *number*".)

2 EASY TO START

There is a Demo.m giving an example of the usage of this package. If you try the demo, it may help you to get quick start. Before you run this package, please check the following things:

1) PSA or JOY can be run from your command line of your working path;
2) Matlab installed in your Unix system;
3) Start Matlab and change the Current Folder to where you store the iPatch package;

Now if you input *Demo* and enter at the command window, this demo can be run and the results will be found in the folder of result/ already included in this package.

2.1 To get the scores,

The following input information need to be provided to the program.

1) ***pdbfile1*** and ***pdbfile2***: two PDB files for two protein families.

2) ***MSAfile***: one multiple sequence alignment (MSA) for the sequence pairs from these two protein families, and it is recommended to prepare this MSA as follows: first, align each protein family separately with Maxalign and Muscle iteratively; second, concatenate each pair of sequences from two protein families into one sequence according to some one2one pairing information; third, two PDB sequences must be included in the MSA with the header specified by '>reference'. The MSA must be in fasta format.

e.g.,

```
>reference
NISDTALTNELIHLLGHSRHDWMNKLQLIKGNLSL-QK-YDR-VFEMIEEMVIDAKHESKL--SNLKTPH
LAFDFLTFNWKTHYMTL--EY--EVLGEI--K----D-LSA------YDQKLAKLMRKLFHLFDQAVSRE
-SENH-LTVSLQTDHPDRQLILYLDFHGAFADPSAFDDIRQNGYEDV-----------DIMRFEITSHEC
LIEIGLD------------NEKILIVDDQSGIRILLNEVFNKEGYQTFQAANGLQALDIVTKERPDLVL
LDMKIPGMDGIEILKRMKVIDENIRVIIMTAYGELDMIQESKELGALTHFAKPFDIDEIRDAVKKYLPL
```

3) ***segments.number*** and ***segments.chain***: specify the two domains in which you are interested by residue sequence number and chain ID. This package only deal with two proteins/domains at a time, but the domain can be non continuous. E.g., segments.number{1} = {1, 10; 60, 100}; segments.number{2} = {20, 48}; segments.chain = {'A', 'A'}.

4) ***location*** ---- It has two cell data with the location vectors for two domains. E.g.,

location{1} = [1,10; 50, 100]; location{2} = [11, 49];

The first domain is located in the regions specified by location{1}, and the second domain is in the region specified by location{2}. **Please make sure that the first domain is inside the region defined by location{1} on the reference sequence, and the second domain is located in the region given by location{2}. Please also make sure the *location* covers the whole reference sequence.** It might have more residues than its corresponding domain now, but it is OK and the program will automatically exclude the extra residues by using the definition of domain according to the ***segments***.

2.2 To print the result

1) The score results will be printed in the following files in ./result/:
APro*jobname*.txt, PPro*jobname*.txt, TProjobname.txt, EBMcBASC*jobname*.txt, IPatch*jobname*.txt
File format:
position score
For pairwise scores:
position1 position2 score
The *position* here is the positions on the MSA after deleting the columns with more 50% (default) gaps. The alignment after the deletion of gaps can be found ./result/ *jobname*_nogap.fasta.
2) There are two types of prediction results you can choose to print out in files like this:
File format:
Number of predictions on the first protein
RankN ResSeqNumber ChainID Score absEntropy
…..…
Number of predictions on the second protein
RankN ResSeqNumber ChainID Score absEntropy
…..…

i. top N predictions given by user specified score:
The prediction results will be print in ./result/result_topN*jobname*.txt
1)***scoreID***: You can chose which score you want to use to give the prediction by set the scoreID, and the default score is IPatch.
2) ***topmode***: You can also choose the mode to select top scores by set topmode: 0, topN scores for all; 1, topN scores on each protein. The default is 0;
3) ***topN***: the number of top scores you choose. The default is 40.
ii. the predictions that have the score above some score cutoff
The prediction results will be print in ./result/result_scorecutoff*jobname*.txt
1) ***cutoff***: what ever reasonable score cutoff you choose by looking at the score distribution
2) ***scoreID***: as before

3 MORE DETAILS ABOUT iPatch FUNCTION

3.1 INPUT of IPatch function

1) ***pdbfile*** ---- two pdbfiles for two proteins. e.g., {'yourprotein1.pdb', 'yourprotein2.pdb'}

2) ***MSAfile*** ---- this is the multiply sequence alignment for the paired protein sequences in FASTA formart. In the MSA, the reference sequence must be specified by the header '>reference'.

3) ***segments*.number** ---- the residue sequence numbers

.chain ---- the chain ID of the boundaries for the interested segments on the reference sequence.Currently, only two segments are allowed, and noncontinuous segments are also allowed.

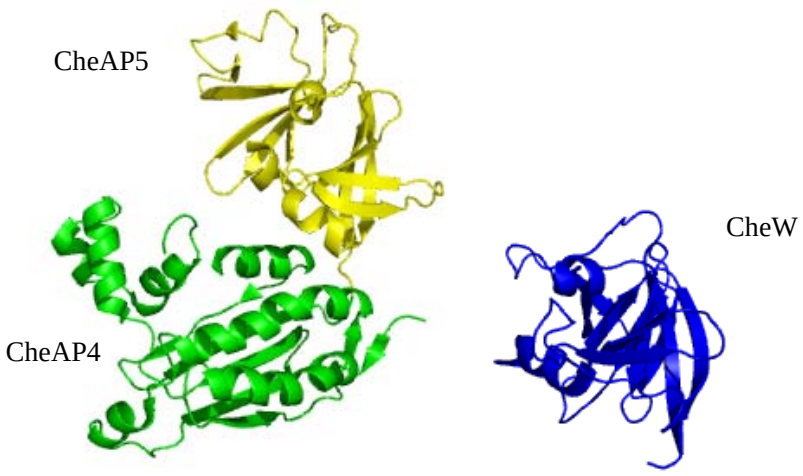
4)***location*** ---- It has two cell data with the location vectors for two domains. E.g.,
location{1} = [1,10; 50, 100]; location{2} = [11, 49];

5) ***jobname*** ---- the name of your complex, it will be in the names of many cache files

6) ***parameters*** ---- (the default settings are showing)

parameters*.*surfaceoption = 0; % the surface information for the domain interested is given by running the psa on the domain; if set it to 1, the surface information for the domain interested is given by running the psa on the whole protein on which this domain is located;

e.g., when you are interested in the interface between CheAP4 domain and CheAP5 domain, the *surfaceoption* needs to be set to 0 so as to calculate the surface for each domain; when you are interested in the interface between CheAP5 and CheW, the *surfaceoption* needs to be set to 1 so the surface will be calculated for CheAP4 and CheAP5 as a whole molecule and those sites on CheAP5 involved in the interface between CheAP4 will be buried and wont have much effect on its bonding to CheW, if the conformation of CheAP4 and CheAP5 wont change a lot when CheAP5 binds to CheW.



parameters.gapcutoff = 0.5; % columns with gap percentage greater than this cutoff wont be used in the score calculation

parameters.discutoff = 4.5; % the intra domain space neighbors are defined by this cutoff, i.e., if the distances between residues that are measured as the shortest distance between any nonhydrogen atoms are less than this cutoff, these two residues are intra domain neighbors

parameters.parallelized = 1; % 1, if you can run the Matlabpool, most of the algorithms can be parallelized; 0, otherwise.

parameters.randomized = 1; % 1, if you want to get the scores for the random matching pairs as backgroud score; 0, otherwise.

parameters.reference = 0; % 1, if you want to include your reference sequence in the score calculation; 0, otherwise.

parameters.cleancache = 1; % clean all the cache files; if set to be 0, the cache files will be used, if any.

parameters.modifiedres = 1; % if 1, change MSE to MET, CSE to CSC; if 0, those modified residues will be ignored.

parameters.manualalignment = 0; % if 0, the alignment between the PDB sequence and the reference sequence will be generated by using the local alignment algorithm swalign(); if 1, the user specified alignment [*jobname*, myAlignment.mat] will be used for the residue number assignment.

3.2 IMPORTANT NOTES of IPatch:

1) MSA

recommend to do the multiple alignment for each protein family separately by muscle-maxalign-muscle concatenate the sequences of the contact pairs by using any pairing information you have and make sure only one-to-one match, and then delete the redundancy by cd-hit .

2) pdbfile

the PDB file will be read into a structure by using pdbread (Matlab function), and only the coordinates in the ATOM section will be used which means those modified residues in the PDB file will be ignored . There is also an option for you to change MSE to MET, CSE to CSC, if necessary. The default option is yes.

3) Distance

the distance between residues are calculated as the smallest all atom distance, which is implemented in the function: compute_atom_distances.m

if you need to calculate the distance only between side chain atom, please comment the lines 26 ~ 28, and uncomment the lines 14 ~ 25.

5) CACHE

there are many CACHE codes you could uncomment them, if you are running this for the same proteins. If you want to clean all the cache files, please set the *parameters.cleancache* to 1.

6) parallel

if Matlabpool is available, parameters.parallelized = 1; (default)

7) alignmentforresseqassign.txt

For the assignment of the coordinates from PDB residues to MSA positions, the local alignment has been used for each protein between the reference sequence in MSA and the sequence in its PDB file, since there usually some residues are missing in the PDB file. Please check this file to make sure this assignment is all right. If there is anything wrong with this alignment, you could make your own alignment by manipulating the **Alignment.mat**, and saving it as [***jobname***, **myAlignment.mat**]. At the same time, set the ***parameters. Manualalignment* = 1** in the **Demo.m**.

e.g., >> load('Alignment.mat');

The cell variable *allAln* will be loaded in Workspace and you can revise the alignment by changing the values of this variable directly. For example,

>>allAln{2}(1,200) = 'S'; allAln{2}(2,200) = '|';

>>allAln{2}(2,209) =' '; allAln{2}(1,209)='-';

The 200th position in the PDB sequence in this alignment for the second protein will be replaced with 'S', and the 209th position in the PDB sequence will be set to '-'; and then, the corresponding positions in the second line of this alignment will be changed to '|' and ' ', respectively. The meaning of the symbols in the second line can be find in the help file of the function swalign() in Matlab. Usually, the '|' means there is a match in the pairwise alignment; otherwise, the ' ' or ':' will be used as mark of mismatch.

>>save([jobname, 'myAlignment.mat'], 'allAln');

Before you re-start the program, please open the Demo.m to set the *parameters. Manualalignment* = 1.

3.3 Output of IPatch function

result.pairwisescores{1} pairPPro

result.pairwisescores{2} pairTPro

result.pairwisescores{3} pairMcBASC

result.sitescores{1} APro

result.sitescores{2} PPro

result.sitescores{3} TPro

result.sitescores{4} EBMcBASC

result.sitescores{5} Comb

result.protein a structure data format with many fields

If you choosed the parameters.randomized to be 1, then there will be another scores structure as result.random with the same fields listed above.

Please note that the scores in the result data structure are those scores for the sites in the MSA after deleting columns with more than 50% gaps (default). This cutoff can be reset in the parameters.gapcutoff for IPatch. The MSA after gap deletion can be found in ./result/ *jobname*_nogap.fasta, and ./processing/*jobname*_nogap.fasta, and it can also be found in protein data structure result.protein.aln.nogap.seq.

4 OTHER TOOLS

1) get the top N predictions

toppredictions.m with parameters

topmode: 0 -- topN score across two proteins

1 -- topN score on each protein

topN: the number of top predictions

scoreID: the ID of the score you choose to give the final prediction, can be

1 -- APro; 2 -- PPro; 3 -- TPro; 4 -- EBMcBASC; 5 -- Comb;

2) get the predictions by setting score cutoff

toppredictions_byscorecutoff.m with parameters

cutoff: what ever reasonable score cutoff you choose by looking at the score distribution

scoreID: as before

There is also one line for the view the prediction results in Pymol.

3) The sequence file and feature file for jalview (http://www.jalview.org/) to mark the predictions on reference sequence, as long as the other seuqences you choose from your MSA by specifying their header names in seq_header. Then you can use Jalview to open your alnfordisplay.fa, and load the feature file predict_feature_reference.feature.

4) Find the real inter-chain contact sites in a big complex from a PDB file

findInterSegmentContact.m with the following I/O data:

input: working path – the path of your PDB file for the complex

complex name – the name of your PDB file for the complex

segment definition – the residue ranges and chain ID's of many segments on the complex.

Each segment is defined as follows:

segmentdefinition.number{i} = [1, 101]; (the start resSeq number and the end resSeq number)

chain{i} = 'A'; (the chain ID of this segment)

output: inter segment contact pairs and contact sites on each segment by residue number, and the data structure of the output is as follows:

output.interfaceASAChange = interfaceASAChange; % interface asa change

output.allcontactpairs = allcontactpairs; % pairs of res number and chain id

output.allcontactsites = allcontactsites; % sites of res number and chain id

output.interdomaincontact = interdomaincontact; % location pairs

output.interdomaincontactSites = interdomaincontactSites; % location sites

output.allcontactsitesASAChange = allcontactsitesASAChange; % asa change for each contact site

output.joypsa = joypsa; % all the asa information for all residues

5 Supplementary Materials

The following supplementary data files can be found on the website

http://sites.google.com/site/qluochina/Home/systemsbiology/ipcop-matlab-package

1) The the MSA and PDB files for Blind Test data set and the list of proteins

blindtestdata.zip and Table S4—blindtestdata.xls

2) The MSA and PDB files for Fitting data set and the list of proteins

fittingdata.zip and Table S3—fitingdata.xls

3) The Matlab code to run i-Patch on the Blind Test data set

demoForTest.m

4) The Matlab code to run i-Patch on the Fitting data set

demoForFit.m

6 Algorithms

The facts that any type of amino acids can be contact sites and any type of pairs of amino acids can be contact pairs on the interfaces suggest that the intra-molecule neighborhood information of a site on the protein surface is as important as this site itself. In this paper, the propensities of different types of amino acids on the interface against the amino acids on the whole surface are used as a measurement of the degree to which a site on the surface is a contact site, different types of pairs of amino acids are introduced to measure the degree of complementarity, and the propensities of triangles of amino acids on the interfaces have also been established to consider the neighborhood information of the contact sites. In addition of those propensities as the profiles of the contact sites, contact pairs, and contact triangles, the profiles of the intra-molecule neighbors of contact sites and contact pairs have also been calculated from our database. Based on these profiles, the amino acid propensity score, pair propensity score, and triangle propensity score have been proposed to predict the inter-protein contact sites. The results on both our fitting dataset

with 35 domain-domain interfaces and blind test dataset with 31 domain-domain interfaces show that these propensity scores clearly outperform the other 5 correlated mutation scores. By using logistic model, we also tried to combine the propensity scores with the McBASC score which is the best score in the 5 correlated mutation scores on our fitting dataset but not on the blind test dataset. The combine score (Comb) takes into account both the profiles of the contact sites and the covariance of the amino acids substitutions of them. The logistic model is fitted on the fitting dataset, and the test results on the blind test dataset show that Comb can achieve a precision of more than 50% at a recall of 20%.

Algorithm descriptions

APro	$S_i^{\text{APro}} = \frac{1}{\lvert N(i)\rvert}\sum_{i_t \in N(i)} \frac{1}{M}\sum_{j=1}^{M} w_{\text{intra}}(a_{ji_t} \mid a_{ji}) \frac{1}{M}\sum_{j=1}^{M} AApro(a_{ji_t})$.
PPro	$S_i^{\text{PPro}} = \frac{1}{\lvert N(i)\rvert}\sum_{i_t \in N(i)} \frac{1}{M}\sum_{j=1}^{M} w_{\text{intra}}(a_{ji_t} \mid a_{ji}) S_{i_t}^{P}$, where $S_i^{P} = \frac{1}{\lvert\{k \in SurfaceB\}\rvert}\sum_{k \in SurfaceB} \frac{1}{M}\sum_{j} Pairpro(a_{ji}, a_{jk})$.
TPro	$S_i^{\text{TPro}} = \frac{1}{\lvert N(i)\rvert}\sum_{i_t \in N(i)} \frac{1}{M}\sum_{j} w_{\text{intra}}(a_{ji_t} \mid a_{ji}) S_{i_t}^{T}$, where $S_i^{T} = \sum_{k \in SurfaceB} \frac{\sum_{t \in N(i)\cup N(k)} \frac{1}{M}\sum_{j} w_{\text{pair}}(a_{jt} \mid a_{ji}, a_{jk}) Trianglepro(a_{ji}, a_{jk}, a_{jt})}{\lvert N(i)\cup N(k)\rvert}}{\lvert\{k \in SurfaceB\}\rvert}$.
McBASC	$S_i^{\text{McBASC}} = \max_{k \in SurfaceB}(r_{ik})$, where s_{ipl} is substitution score given by substation matrix Blosum62 for the amino acid substitution between sequences p and l in the MSA at site i, $r_{ik} = \frac{1}{M^2}\frac{\sum_{pl}(s_{ipl} - \langle s_i\rangle)(s_{kpl} - \langle s_k\rangle)}{\sigma_i \sigma_k}$, $\langle s_i\rangle$ is the average value of the substitution scores $\{s_{ipl}\}$, and $\sigma_i = \frac{1}{M^2}\sum(s_{ipl} - \langle s_i\rangle)^2$.

For each site $i \in SurfaceA$, the scores S_i are calculated. M is the number of sequences, a_{ji} the residue in sequence j at site i, $N(i) = \{i_t : d_{i,i_t} < 4.5\}$ is the set of all surface intra-domain neighbours of site i, and $\lvert N(i)\rvert$ is the number of sites in

set $N(i)$. The distance between site i and itself is zero, so $i \in N(i)$. The APro, PPro, TPro and EBMcBASC (Table 4) scores are combined by the model:

$$\text{logit(Comb)} = \beta_0 + \beta_1 \text{APro} + \beta_2 \text{PPro} + \beta_3 \text{TPro} + \beta_4 \text{EBMcBASC},$$

where $\beta_i, i = 0,1,2,3,4$ are the model coefficients that are given by fitting the logistic model on the fitting dataset.

Figures

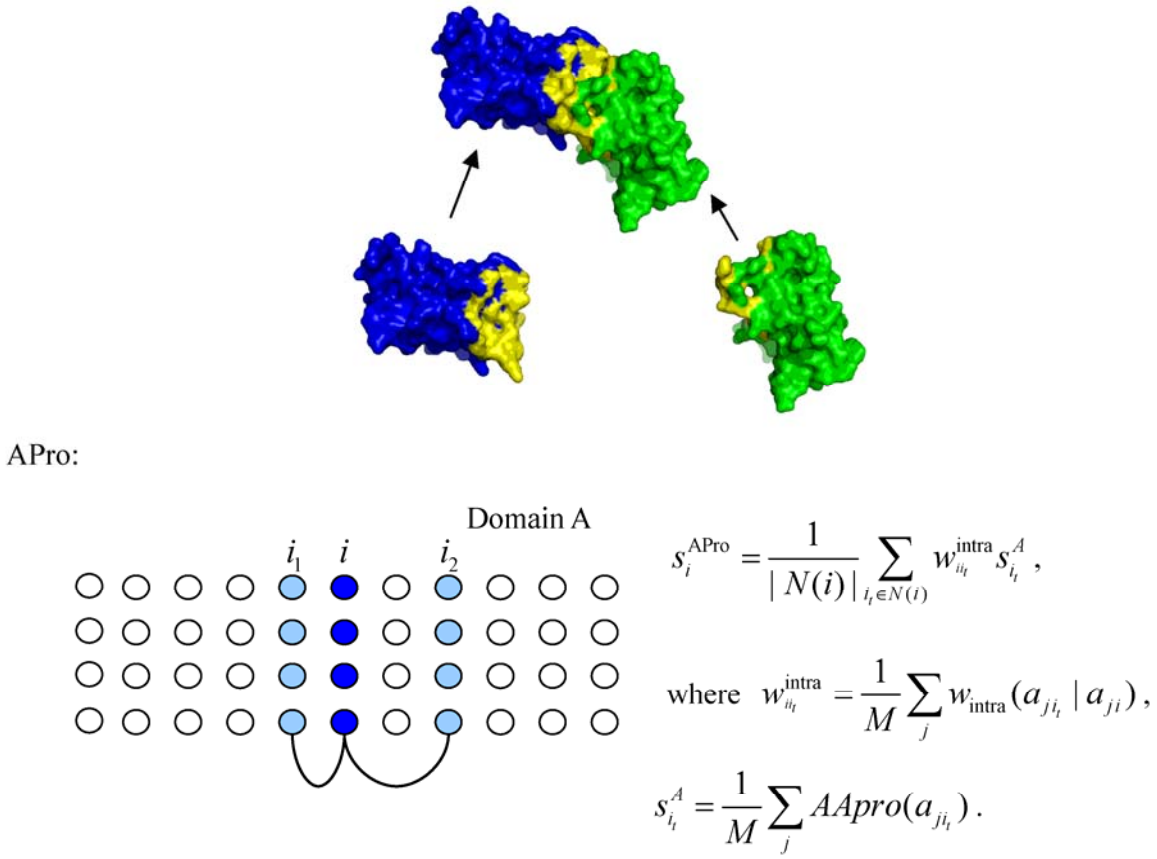


Figure 1. Calculation of the amino acid propensity score (APro). Intra-domain weights are calculated using residues in the same domain as residue *i* which are within 4.5Å of residue *i*. For full details see Table 4.

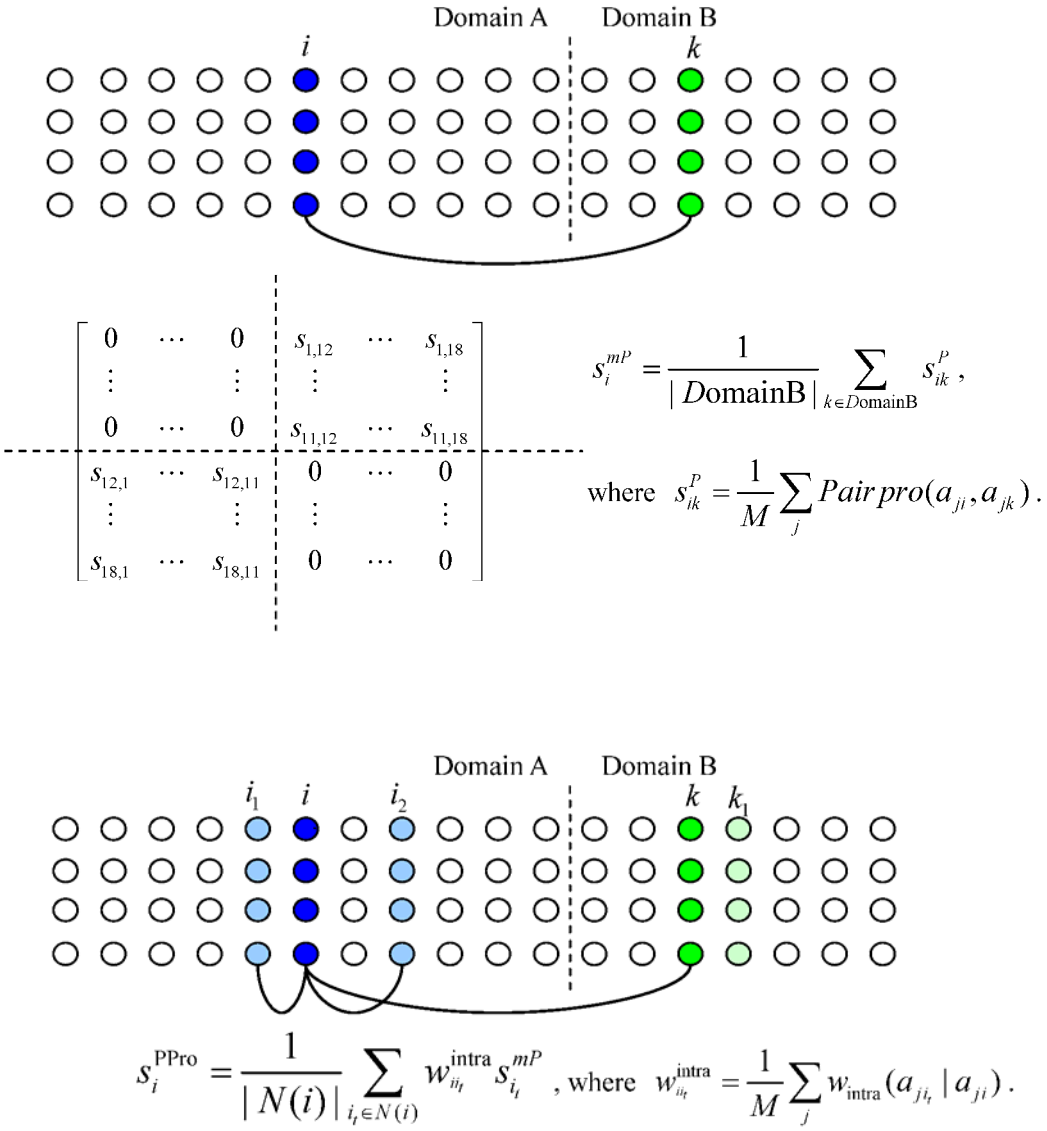
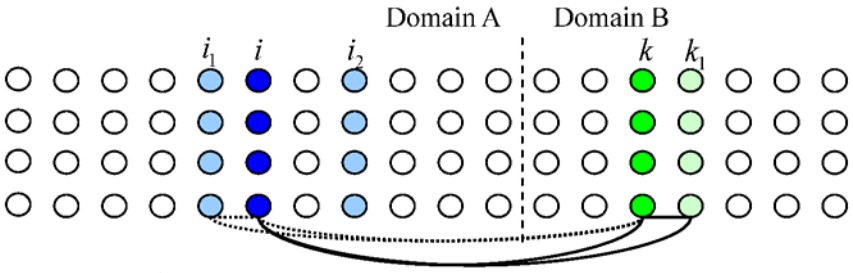
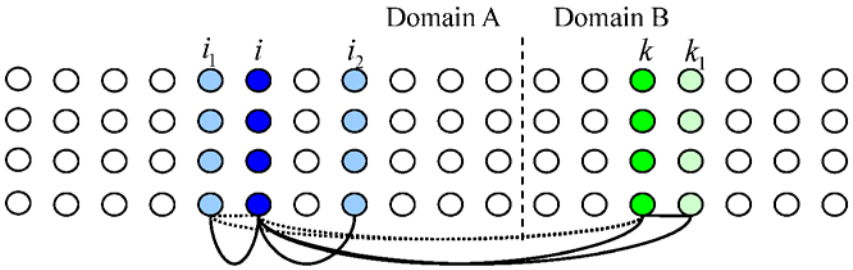


Figure 2. Calculation of the pair propensity score (PPro). All pairs of residues (*i*, *k*), where *i* is from domain A and *k* is from domain B, are used to calculate s_i^{mP}. The intra-domain neighbours of each site *i* on domain A are then used to calculate a weight for site *i*.



$$s_{ik}^{mT} = \frac{1}{|\,\mathrm{DomainB}\,|} \sum_{k \in \mathrm{DomainB}} s_{ik}^{T},$$

$$\text{where} \quad s_{ik}^{T} = \frac{\sum_{t \in N(i) \cup N(k)} \frac{1}{M} \sum_{j} w_{\mathrm{pair}}(a_{jt} \mid a_{ji}, a_{jk}) \mathit{Triangle pro}(a_{ji}, a_{jk}, a_{jt})}{|\,N(i) \cup N(k)\,|}.$$



$$s_i^{\mathrm{TPro}} = \frac{1}{|\,N(i)\,|} \sum_{i_t \in N(i)} w_{ii_t}^{\mathrm{intra}} s_{i_t}^{mT}.$$

Figure 3. Calculation of the triangle propensity score (TPro). A pair of residues (*i*,*k*), where *i* is from domain A and k is from domain B, are selected. A third site is then selected which is within 4.5Å of either site *i* or *k*, to make a triangle. The intra-domain neighbours of site *i* are then used to calculate a weight for site *i*.