Exercise SBI 11-13.

Statistic Potentials

The analysis of a database of protein structures shows that certain residues end to be in close proximity than others. This frequency can be interpreted as probability and by the inverse of the Boltzmann law we can calculate energies. This potential force field is named statistic potential or knowledge based potential:

$$\mathbf{P=(1/z)(e^{(-E/kT)})}$$
$$\mathbf{E= -KTlnP + KTlnZ}$$

Recent advances on high-throughput technologies have produced a vast amount of protein sequences, while the number of high-resolution structures has seen a limited increase. This has impelled the production of many strategies to built protein structures from its sequence, generating a considerable amount of alternative models. The selection of the closest model to the native conformation has thus become crucial for structure prediction. Several methods have been developed to score protein models by energies, knowledge-based potentials and combination of both. Here we present one of these methods, named PROSA. We will use PROSA scores to:

1) Check the quality of one protein structure (only valid for soluble globular proteins)

2) Check by threading who's the best conformation between several models of different folds, using a Z-score function Z-score= $(E - \mu)/\sigma$ (see theory)

Practice 5.1 : Prosa

TUTORIAL

Copy the file exercise_5 in your directory of work:

Also copy and open the manual of Prosa2003.

```
> evince Manual_prosa2003.pdf
```

The files required to execute the sessions of the manual are found in the subdirectory PROSA of practice_5. Go to this subdirectory

Follow the tutorial of Prosa to execute 8 sessions by running the program prosa2003. Check that all aliases are running (do source of bashrc_aules), and run prosa

```
> prosa
```

The program reads PDB structures with the command “read pdb”. To calculate the energy profile per residue it uses the command “analyse energy”

COMB energy is the result of a linear combination of PAIR and SURF energies (see theory for pair-wise and solvation).

The option “draw” will be used to visualize a specific energy term (with boolean1) or neglected it (with boolean 0).

Session 4b is special and it’s used to calculate the Z-score. Results are stored in a file named z-result. Open the file once created and check the Z-scores of the combined, pair and surface energies. In order to know if one structure is correct you will have to compare the Z-score with the normal set of Z-scores from PDB. The set of z-scores of PDB is plotted in the manual of prosa as a function of the length of the molecules. Check the length of the protein-problem and compare if the Z-score is much higher than most structures with the same length.

Session 1: Learn to read PDB files and to calculate and visualize energies	Session 2: Learn to compare different energies and recognize the relevance of the resolution.
read pdb pdb2aat.ent obj1 analyse energy obj1 plot winsize obj1 50 plot draw * obj1 1 plot color comb obj1 cyan color surf obj1 magenta plot read pdb pdb1spa.ent obj2 analyse energy obj2 plot color * obj2 red winsize * 50 plot draw * * 0 draw pair * 1 plot export plot myplot	read pdb pdb2aat.ent 2aat read pdb pdb3aat.ent 3aat read pdb pdb1aaw.ent 1aaw read pdb pdb1spa.ent 1spa analyse energy * draw * * 0 draw pair * 1 color * 3aat cyan color * 1aaw red color * 1spa white winsize * 50 plot export plot refine

Session 4a:
Learn to shift the graph in order to compare the energy of sequences of different length

```
read pdb pdb2gn5.ent gn
read pdb pdb1bgh.ent bgh
analyse energy *
color * bgh red
winsize * 10
draw * * 0
draw pair * 1
graph title 2gn5 1bgh

shift bgh 1

plot
export plot session4a
```

Session 4b: Learn to calculate Z-scores with Prosa. Results are not in a profile plot but in a text file named z-result.slp. This is used as threading. The selection of the best fold is obtained by comparing the Z-scores (COMB, PAIR and SURF) named “zp”.

```
read pdb pdb2gn5.ent gn
read pdb pdb1bgh.ent bgh

init zscore
combine type sdev

zscore gn z-result

sorted_depth = 0

zscore bgh z-result
```

Session 5:
Learn to use CB and CA potentials

```
pair potential $PROSA_BASE/pII3.0.pair-cb pcb
surface potential $PROSA_BASE/pII3.0.surf-cb scb
pair potential $PROSA_BASE/pII3.0.pair-ca pca
surface potential $PROSA_BASE/pII3.0.surf-ca sca
read pdb pdb2aat.ent aat
read pdb pdb1spa.ent spa
use potential aat pca sca pcb scb
use potential spa pca sca pcb scb
analyse energy *
color * aat yellow
color * spa red
winsize * 50
plot
diff aat spa diff
plot
draw * * 0
draw pair * 1
plot
export plot session5
```

Session 6:
Learn to use only CA potentials

```
pair potential $PROSA_BASE/pII3.0.pair-ca pca
surface potential $PROSA_BASE/pII3.0.surf-ca sca
read pdb pdb2phy.ent phy
read pdb pdb2lzh.ent lzh
use potential phy pca sca
use potential lzh pca sca
analyse energy *
color * phy yellow
color * lzh red
winsize * 10
plot
export plot session6
```

Session 7: Learn to analyse mutant structures without generating new models	Session 8: Learn to compare the stability of different regions of a protein using the mutability.
read pdb pdb1ubi.ent wt mutate sequence wt 46 P mutant1 mutate sequence wt 5 E mutant2 mutate sequence mutant2 32 P mutant3 mutate sequence mutant3 47 L mutant4 list objects analyse energy * color * mutant1 red hide * show wt show mutant1 plot diff wt mutant1 diff1 hide * show diff1 plot diff wt mutant4 diff4 color * diff4 blue show diff4 plot init zscore delete diff1 delete diff4 zscore * exit	init zscore read pdb pdb1aqk.ent,H, Fd analyse mutability Fd 69 analyse mutability Fd 98-106,205-210 Fd_xmpl randomise sequence Fd 153,180,182,184 Fd_epi1 randomise sequence Fd 100,104-106 Fd_epi2 exit

Answer the following questions:

1) Use your models from previous practice 4 and obtain the Z-score and the profile of energies per residue. Check the Z-score to decide if the structure of the model is correct
2) Compare the energies of your models (per residue) with the energy of the templates. Select your best model.
3) Has your model some regions wrongly modelled even though the Z-score is correct?
4) Identify the wrongly modelled regions in a superposition between the model and the original template.
5) Can you identify the region wrongly modelled using *in-silico* mutations? How would you do it?
6) Try to answer the following question, even if this is not the case of your model: After checking the energy profile of a model, this shows a pick produced in the COMBINED energy. However, the pick is mainly produced by the SURFACE energy while the PAIR energy is correct. Do you think the model is incorrect? What's the most likely reason for this effect?

Practice 5.2: Secondary structure prediction and comparison.

Programs:

Dssp, to calculate the REAL secondary structure of a protein. It requires a PDB file.

Psi-pred, to PREDICT the secondary structure of a protein. It requires the sequence in FASTA format.

1.1-Psi-pred

The program uses a neural network to predict the secondary structure. First it uses psi-blast to increase the information of sequences and then it runs the prediction. We will use a FASTA format file, named “target.fa”, from practice2 and practice4, for which a model was obtained.

Command:

```
➢ psipred  target.fa
```

Results are in files: target.ss2 & target.horiz

Where H means helix, C means coil and E means strand.

Then we run a script to transform the output “target.ss2” in an alignment with format PIR:

```
➢ psipred.pl target.ss2 > psipred.pir
```

1.2-DSSP

The program DSSP runs with the command: dssp <input> <output>
in our case:

```
➢ dssp model.pdb model.dssp

  (optional /mnt/NFS_UPF/soft/dssp/dssp model.pdb model.dssp )
```

Again, this result shows the real secondary structure in a format difficult to compare with the sequence. Therefore we run another script to transform this in PIR format

```
aliss.pl model.dssp > dssp.pir
```

(optional aliss_old.pl model.dssp > dssp.pir)

By merging both files, dssp.pir and psipred.pir we obtain a file in PIR format with the alignment of sequences and secondary structures, predicted and real.

```
cat psipred.pir>compare.pir
cat dssp.pir >>compare.pir
```

Then we transform this file (in format PIR) into clustal format:

```
aconvert –in p –out c <compare.pir>compare.aln
```

And now it’s easy to compare the sequence and its secondary structure prediction with the real secondary structure of the model. In the case of DSSP, helix is H, beta-strand is E and the rest can be identified with different symbols, all of them treated as coil.

Finally, if there is a region where the secondary structure of the model is different than the predicted secondary structure, we still have to check the quality of the prediction. The file target.horiz contains this information in the rows identified as “Conf”. The information is encoded in numbers from 0 to 9, where high numbers indicate reliability.

Confirm that the region wrongly modelled coincides with a region where the prediction of secondary structure doesn’t correspond with the structure of the model and that the reliability of the prediction is acceptable.

Once a region with wrong secondary structure has been found you can modify the model with a modification of the alignment between the sequence of the target and the template. The following is just an example of how this modification can be done:

Step 1) Confirm that the region is wrongly modelled according to the statistical potentials and that the prediction of secondary structure is different than the structure of the model. Superpose the model and the original template and check if the model has a broken secondary structure.

Step 2) As an example, I will suppose we have a model with a broken alpha-helix. Let this be the alignment between the sequence of target and the template, indicating the predicted secondary structure and the structure of the original template

```
TARGET        XXXXXXXXXXXXXXXXXXXXXX
PREDICT_SS    -HHHHHHHHHHHHHCCCCCCEE
TEMPLATE      YYYYYYYY-----YYYYYYYYY
DSSP_TMPL     HHHHHHHH-----HHHHHCCEE
```

Step 3) Modify the alignment such that the secondary structure of the original template is nor broken. In the case of the example the modified alignment should be:

```
TARGET        XXXXXXXXXXXXXXXXXXXXXX
PREDICT_SS    -HHHHHHHHHHHHHCCCCCCEE
TEMPLATE      YYYYYYYYYYYYY-----YYYY
DSSP_TMPL     HHHHHHHHHHHHH-----CCEE
```

Although there is no full coincidence between the predicted secondary structure and the structure we wish for our model, the important is to maintain the secondary structure.

It has to be noted that I moved the gap to the right side (towards the C-tail), but the gap could also be moved to the left (towards the N-tail side):

```
TARGET        XXXXXXXXXXXXXXXXXXXXXX
PREDICT_SS    -HHHHHHHHHHHHHCCCCCCEE
TEMPLATE      -----YYYYYYYYYYYYYYYYY
DSSP_TMPL     -----HHHHHHHHHHHHHCCEE
```

Or also partially moved to the right side and to the left (towards the N-tail side):

```
TARGET        XXXXXXXXXXXXXXXXXXXXXX
PREDICT_SS    -HHHHHHHHHHHHHCCCCCCEE
TEMPLATE      --YYYYYYYYYYYYY---YYYY
DSSP_TMPL     --HHHHHHHHHHHHH---CCEE
```

Step 4) Transform the alignment in the right format to run the program MODELLER and create a new model, or several models, with the modification. Run modeller, generate the models and superpose them with the template and the original model. Confirm that the modification of the alignment has produced your expected model (remember: the gap to left implies a loop region on the N-tail side, the gap to the right implies a loop modelled in the C-tail side)

Step 5) Optimize the model.

Optimization can be obtained with energy minimization or simulated annealing.

Step 6) Compare with Prosa2003 the different models. Check the improvement and, if necessary, change again the alignment.

Not always is possible to obtain a correct model. Occasionally it's even more feasible to leave a gap in a loop region and check in the database other structures that could fit in the unstructured loop region. Often this approach requires to check also the natural allocation of gaps by checking the different structures of the family and testing the normal modes of motion o loops around the core scaffold of the fold. Nevertheless, this is out of subject for the exercise.

Answer the following questions:

1) Compare the Z-scores of the different models (as in practice 5.1) and it's modifications. Select the best model according to Z-scores.
2) Compare the energy profiles among different models (with/without modifications) and select the best model. Is this selection the same as in question 1? Try to reason your choice.
3) Use the mutability analysis and the use of several mutant-forms to select the best model and interpret the results.
4) Once you have a model with a gap (loop) and without broken secondary structures, try to find other structural models that could fit and match the sequence of the gap and fill the structure. If you have found some putative structures try to fit the extremes N-tail and C-tail with the model. If you have found a putative loop template: propose a new modification of the alignment to add this new template.

Practice 5.3: Threading

Copy and uncompress the folder exercise_5 in your working space. For testing the threading programs we will use the same sequence of target.fa, as in practices 2 and 4. For further additional tests and questions of this practice, copy the folder practice5 in your working space (see practice 5.1) and use the folder THREADING.

1) Program THREADER

The program THREADER uses a neural network similar to PSIPRED and aligns the predicted secondary structure with the set of known structures. We can use PSIPRED independently of THREADER or run THREADER's prediction of secondary structure. Then THREADER scores the best structural matches using a statistic potential.

Help is obtained with the command

```
threader -H
```

Putative structures are stored in the folder:
/mnt/NFS_UPF/soft/THREADER3.4/tdb

The list of sequences to thread our query is defined in the file **psichain.lst** that exist in the $THREAD_DIR folder (environment variable of THREADER program). If we want to use a reduced set we can create our own list with the codes of the files in /mnt/NFS_UPF/soft/THREADER3.4/tdb and add at the end of the command the list. For example select some codes from psichain.lst and generate the file **mylist.lst**. We will use the same target as for modelling, which we know is a serine-proteinase. Therefore we select the chains that are also proteinases in the database of threader. Run the following search:

```
grep PROTEINASE $THREAD_DIR/tdb/* | fgrep -v INHIBITOR | cut –d “ “ -f 2
```

This shows a list of codes of PROTEINASES that are not INHIBITORS. Store the names in mylist.lst for the next use.

To run the comparison with secondary structure we have to obtain first the secondary structure with psipred, the run the threading with the file “horiz” generated with psipred:

```
psipred  target.fa
```

Then, we run threader using a specific list from mylist.lst. We select also options –v, -j and –pm to obtain all possible tests (-j), extract all verbose information (-v), and store all tread alignments in MODELLER format (-pm)

- ➢ threader –pm –v –j target.horiz results.ssout mylist.lst >& results.sslog

To evaluate the result we use the expert application:

- ➢ $THREADER_DIR/texp/texp -s $THREADER_DIR/texp/weights.dat results.ssout | sort –n –r > results.texp

Then we obtain the best threading score and the alignment between the target and the template can be found in the file results.sslog with PIR format, prepared to run MODELLER.

We can directly run the threading without using the previous secondary structure prediction, by running the command:

- ➢ threader –pm –v –j target.fa results.out mylist.lst >& threader.log

Results of the combined energies of pairwise residue-residue interactions and solvation are in column 6, and z-scores of this energy in column 7. In column 8 are the z-scores of column 7 where the worst results are given with a low negative value -9.99. The best scores do not necessarily mean a correct model, unless a limit threshold is reached.

Generate the model of target.fa using the best candidates selected from results.out. WARNING: use the alignment given by the results of threader, found in the file threader.log. The file is already in the format accepted by modeller (PIR). If you have any doubts transform the alignment to CUSTALW format and back again to PIR format.

2) Programs ALPHAFOLD, PHYRE, iTASSER, FUGUE and MODLINK

Run website programs ALPHAFOLD, PHYRE, FUGUE and MODLINK with the sequence target.fa. Pay attention to the thresholds that define the quality of the results, some have to be positive scores while others are negative or based in small p-values. Obtain the results and compare the models with Prosa: PHYRE, iTASSER and AlphaFold produce PDB files.

These servers take some time to run. We have upload in Campus Global the results of iTASSER and Phyre. The result of AlphaFold is alreadyy in Uniprot if you check for P11018.

Adresses:

AlphaFold: https://alphafoldserver.com/
iTasser: https://zhanggroup.org/I-TASSER/about.html
Phyre: http://www.sbg.bio.ic.ac.uk/phyre2/html/page.cgi?id=index

Answer the following questions:

1) Compare the Z-scores of the different models of target.fa obtained by homology modelling, AlphaFold, iTasser, THREADER, PHYRE. Select the best model according to Z-scores.
2) Compare each model with the corresponding template structure using the energy profile per residue.
3) Model the structure of target.fa with the first five choices of the programs THREADER, PHYRE. Compare and understand the models when the scores are acceptable according to the threshold quality of each program. What's your opinion on these methods?.