Alignments

In this Übung you get the opportunity to perform database searches with protein sequences. You will use the sequence alignment program BLAST from the internet services of NCBI. It is fast enough to perform heuristic alignments of a query sequence to all sequences in a database within seconds. You will also try out the powerful molecule viewing and manipulation program UCSF Chimera.

Remember: Whenever you use free internet services, do not overload them with jobs as someone has to pay for all this.

Please prepare some directories and files that you will need during the exercise.

1) The shell's configuration is stored in a hidden textfile (shell is the program that is run inside the konsole program and provides the command prompt). Please determine which shell is set as standard for your account by entering:
   `echo $SHELL`
   If the output was `/bin/tcsh` then your configuration is stored in a file named `.tcshrc`.
   1. Touch it with the command:
      `touch ~/.tcshrc`
   2. Open it in your favourite text editor (e.g. `vim`, `emacs`, `joe`, `pico`, `xedit`, `kedit`, `kate`, `kwrite`, ...). Do NOT use any word-processing program like OpenOffice, StarOffice or the like. Append the following line at the end of the file (in vim press i before writing and Esc after writing):
      `setenv PATH /usr/local/zbh/bin:$PATH`

   Otherwise, if the output was `/bin/bash` then your configuration is stored in a file named `.bashrc`.
   1. Touch it with the command:
      `touch ~/.bashrc`
   2. Open it in your favourite text editor (e.g. `vim`, `emacs`, `joe`, `pico`, `xedit`, `kedit`, `kate`, `kwrite`, ...). Do NOT use any word-processing program like OpenOffice, StarOffice or the like. Append the following line at the end of the file (in vim press i before writing and Esc after writing):
      `export PATH=/usr/local/zbh/bin:$PATH`
2) Save the file and close your editor.

Editor	Save	Exit
`vim` - Vi IMproved	Esc `:w` Enter	Esc `:q` Enter
`emacs` - GNU project Emacs	Ctrl-`x` Ctrl-`s`	Ctrl-`x` Ctrl-`c`
`joe` - Joe's Own Editor	Ctrl-`k` `d`	Ctrl-`k` `q`
`pico` - simple text editor in the style of the Pine Composer	Ctrl-`o`	Ctrl-`x`
graphical editors	*File->Save*	*File->Quit*

   The next time you start a terminal/shell the commands in `/usr/local/zbh/bin` can be executed directly.
3) Logout completely.
4) Login again.
5) Create a directory named `alignment` in your home directory and change into it.
6) Create a subdirectory `queries` and a subdirectory `results`.
7) Copy the files
   `protein1.fasta`

`protein2.fasta`
`protein1.pdb` and
`protein2.pdb` from
`/home/schenk/teaching/SS_2009/Bioinf/alignments/<Übungsgruppe>`
to the new subdirectory `queries`. Be sure to replace `<Übungsgruppe>` (including the <>) with either `first` or `second`, depending on the group you are doing this Übung.

The files you copied with the suffix `.fasta` are text files containing protein sequences in the FASTA format. Have a look at their contents. You should be able to recognise a comment line starting with > and the amino acid sequence. Unfortunately, the comment is not very informative yet. Lets find out about the function and the name of the proteins these sequences fold to via a BLAST-search on the protein data bank (PDB), which contains all known protein structures. Here is how you can find out to which proteins these sequences fold:

1) Open the following URL in a browser window (e.g. `firefox`, `netscape` or `konqueror`):
   `http://www.ncbi.nlm.nih.gov/blast/`
2) Choose the program *protein blast (blastp)*, as you will query with a protein sequence.
3) BLAST offers two possibilities to upload your sequence query. You can either directly paste your sequence into the text field of the HTML-form or you may upload the text file containing your query (e.g. `protein1.fasta`). Optionally, you may choose a sound title. Leave all other fields empty.
4) Choose the database *Protein Data Bank proteins(pdb)* and select the *blastp* algorithm.
5) If you feel brave you can play around with the algorithm parameters and see how they effect your results (optional). Otherwise, just leave them as they are. The default values should do fine.
6) Hit *BLAST* and wait until the results return. (The browser window will be reloaded automatically after a few seconds.)
7) You should save your results locally in your newly created `results` directory for later comparison.

The BLAST-search returns a list of alignments of your query sequence and the best scoring hits in the database.

- Note down the names of the best ten hits (the pdb id and chain number is enough) together with their scores, e-values and sequence identity of the alignments.
- To what protein class do they belong? Which hit is probably your protein?
- Open the fasta file in a text editor and add the name and class of the putative protein to the comment. Do not break the comment line. The fasta format specifications allow only a single comment line. Save and close the file.

When you have done a BLAST-search for both protein sequences, respectively, compare the two hit lists.

- Do they share any proteins?
- Why are they nevertheless in the same class?

In order to further analyse their relation we will do a pairwise alignment of the two sequences with the program Chimera. Chimera's main purpose is viewing and analysing large biomolecules, such as proteins and nucleic acids. It provides a user friendly interface where you can view, rotate, translate, zoom and even manipulate a three-dimensional representation of the molecule. It is able load several molecules at once. One functionality out of many is to compare two protein structures by superimposing one molecule onto another based on a sequence alignment. We can therefore use this program not only to compare our two sequences but to relate also their corresponding structures to each other at the same time.

1) Run the program Chimera by entering
   `chimera &`
   at the command prompt. Do not forget the ampersand & at the end of the command. It will

ask Linux to execute `chimera` in the background, that means you get the prompt back and may enter further commands.

2) Open the pdb-file `protein1.pdb` from your `queries` directory via *File->Open*.
3) A representation of the molecule is rendered in the *main* window. You may control the view with the mouse: pressing the left mouse button over the molecule and moving the mouse carefully will rotate, the middle button will translate and the right button will zoom the molecule.
4) Chimera has several display styles. The default is a full atom representation. You can change them in the *Actions* menu. Change the display style to ribbon only: *Actions->Ribbon->show* and *Actions->Atoms/Bonds->hide*.
5) You should be able to realise that the molecule is a dimer. For simplicity, we will constrain our analysis to one chain. In order to remove the unwanted chains, select chain A (*Select->Chain->A*), invert the selection (*Select->Invert (selected models)*) and delete the atoms (*Actions->Atoms/Bonds->delete*).
6) Save the remaining chain A in a new pdb-file with the name `protein1A.pdb` (*File->Save PDB..*).
7) In order to highlight the secondary structure content lets colour the ribbon accordingly. First open the *Model Panel* (you will also need it later), *Favorites->Model Panel*. Mark the line with the name *protein1.pdb* by clicking it. Now click on the button *color by SS...*. A new window will pop up where you can choose the colours for helices, strands and coils. Clicking OK will close this window and the structure will be rendered in your colours. Can you find parts where the polymer forms helices or where it forms sheets?
8) Now lets have a look at the other protein. Repeat steps 2) to 4) with the pdb-file `protein2.pdb`. If the second protein has more than one chain (not a monomer), repeat steps 5) and 6) with the second protein. When saving the pdb save only the model for protein2.
9) Highlight the secondary structure of the second protein with different colours than the first.
10) By now, you should see two chains coming from two different proteins. Can you see any structural similarity, yet?
11) The similarities are easier to see if the two structures are superimposed. Chimera can do this automatically for you via a sequence alignment. In the *Model Panel* mark both entries and click on *match...*. A new window pops up. We want to match the chain A of the second protein with the chain A of the first protein. Therefore,
    1. tick *Specific chain(s) in reference structure with specific chain(s) in match structure*.
    2. Mark chain A of protein1 as the *reference chain* and select chain A of protein2 as the *chain to match*.
    3. Untick *Include secondary structure score*.
    4. Select the *Needleman-Wunsch* algorithm with the substitution matrix *BLOSUM-62*. Leave the gap penalties at 11 to open a gap and 1 to extend it.
    5. Tick *Show alignment(s) in Multalign Viewer*.
    6. Tick *Iterate by pruning long atom pairs* with a threshold of *5.0 angstroms*.
    7. Click OK.

The *Multalign Viewer* shows the sequence alignment. The matches that Chimera used to compute a transformation matrix are highlighted. The superimposed structures in the *main* window should enable you now to easily see the structural similarities.

1) Note down the RMSD (root mean squared difference) displayed at the bottom of the *main* window (if it vanished, lock in the *Reply Log*: *Favorites->Reply Log*).
2) In the *Multalign Viewer* window go to *Tools->Percent identity...* and click OK. Note down the percentual sequence identity displayed at the bottom of the *Multalign Viewer* window (also in the *Reply Log*).

The structural similarity is quite high (low RMSD) although the sequence identity is relatively low and that is the reason why BLAST did not find a relation between the two proteins. What can you learn from this observation? Think of the diversity of sequence space versus structure space.

Finally, you may want to save the sequence alignment and a picture of the superimposed structures

for later use and your records in the `results` directory.