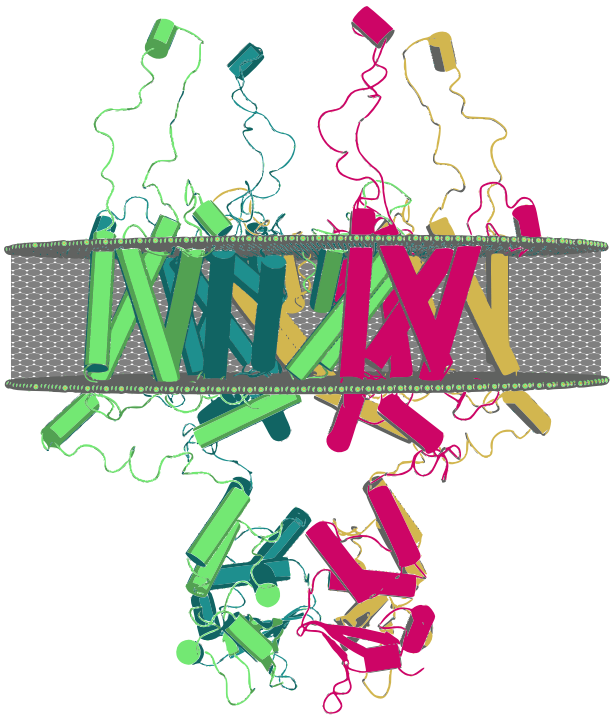
Molecular Modeling of Membrane Proteins

A homology modeling tutorial



Ariela Vergara-Jaque
Nanotechnology Innovation Center of Kansas State
Institute of Computational Comparative Medicine
Department of Anatomy and Physiology
Kansas State University

October 27, 2015

Introduction

Because of the difficulty of expressing, purifying, and crystallizing integral membrane proteins, relatively few structures have been elucidated to date. A protein structure is always of great assistance in the study of protein function, dynamics, interactions with ligands and even in structure-based drug discovery. In the absence of an experimentally determined structure, **comparative** or **homology modeling** can provide a useful 3D "low-resolution" structure, which will contain sufficient information about the spatial arrangement of important residues in the protein in order to guide the design of new experiments.

Specifically, voltage-activated potassium (K_v) channels are integral membrane proteins that allow the potassium ions flux across the cell membrane. These proteins are important drug targets because they play a crucial role in controlling a very wide spectrum of physiological processes. *Shaker* potassium channel from *Drosophila melanogaster* was the first channel cloned and characterized at the molecular level; thus, it has become an important tool for studying human pathologies due to its homology to human K_v channels. Interestingly, despite decades of work in the structure and function of *Shaker*, its 3D structure at atomic resolution remains unknown. Therefore, here we will focus on modeling a truncated version of *Shaker*, whose properties are the most often studied.

Getting Started

This tutorial show all the steps required to model a membrane protein structure using a known experimentally determined structure of a homologous protein as a template.

The following software are required, which are available as online/webserver versions or can be downloaded for free to run on most platforms. We will use the online/webserver versions for most of these software, but three of them (Jalview, Pymol and Modeller) should be installed in your computer.

- **UnitProt**: freely accessible resource of protein sequences and functional information (`http://www.uniprot.org/`).
- **PSIPRED**: Psi-blast based secondary structure prediction server (`http://bioinf.cs.ucl.ac.uk/psipred/`).
- **TOPCONS**: web server for consensus prediction of membrane protein topology (`http://topcons.cbr.su.se/`).
- **Jalview**: free program for multiple sequence alignment editing, visualization and analysis (`http://www.jalview.org/`).
- **Psi-Blast**: position-specific iterative basic local alignment search tool (`http://blast.ncbi.nlm.nih.gov/`).
- **PDB**: repository of information about the 3D structures of proteins (`http://www.rcsb.org/`).
- **PyMOL**: 3D molecular visualization software (`http://www.pymol.org/`).
- **OPM**: orientations of proteins in membranes database (`http://opm.phar.umich.edu/`).
- **DSSP**: database of secondary structure assignments (`http://www.cmbi.ru.nl/dssp.html`).

- **AlignMe**: membrane protein sequence alignment web server (`http://www.bioinfo.mpg.de/AlignMe`).
- **Modeller**: program for comparative protein structure modeling (`https://salilab.org/modeller/`).
- **ProQM**: protein quality predictor (`http://www.bioinfo.ifm.liu.se/ProQM/index.php`).
- **Procheck**: program to check the stereochemical quality of a protein structure (`http://www.ebi.ac.uk/thornton-srv/software/PROCHECK/`).

A '**Tutorial_Modeling**' folder will be provided for this tutorial, including all the required files. Note that if you can not do any step, in the **../Tutorial_Modeling/example_files** folder are the files with the results.

1 Obtain the protein primary sequence

In order to model a protein structure, we first need to obtain its whole sequence from the UniProt website (`http://www.uniprot.org/`), using the most known name of that protein.

1. Type '*Shaker*' in the query box at the top of the UniProt webpage. Make sure the query box state is "UniProtKB". Click on the "Search" button.
2. To cut down on the volume of results, we can restrict the searching on the "Filter by" tab on the left-hand side of the query box. From the "Popular organisms" tab select "Other organisms" by typing '*Drosophila melanogaster*' and clicking "Go".
3. Select the 'KCNAS_DROME' entry name (Potassium voltage-gated channel pro...) by clicking on the hyperlink to 'P08510' to view the UniProt entry.
4. From the search result page, you can download the protein primary sequence by clicking the "Format" tab and selecting "FASTA (canonical)" format.
5. Save the sequence in your working directory (which should be the directory of the '**../Tutorial_Modeling**' folder provided for this tutorial. Save the file as 'Shaker.fasta'.

2 Predict secondary structure segments in the target sequence

For protein sequences of unknown structure, one of the first and more insightful analyses is a prediction of the secondary structure, because it can be used in guiding sequence alignment and the modeling. PSIPRED (`http://bioinf.cs.ucl.ac.uk/psipred/`) is one of the most popular and accurate secondary structure prediction method. The server allows users to submit a protein sequence, perform a prediction and receive the results via email or graphically on the web.

1. Insert the amino acid sequence for *Shaker* in the "Input Sequence" field. You should copy the whole information from the 'Shaker.fasta' file and paste it here. Make sure the selected prediction method is "PSIPRED v3.3 (Predict Secondary Structure)".
2. You must provide a short identifier for your PSIPRED job ("Short identifier for submission"), e.g. 'PsiPred_Shaker', and optionally your email address to have the results. This is recommended, because the results may take a short while to arrive.

3. Press “Predict” to run the job.

4. After a few minutes the results of the prediction will appear on the screen. Select the “Downloads” tab and click the option “Download a zip of everything (may take some minutes to prepare)”. Save the “downloads.zip” file and decompress it. Three files (*.psipass2, *.ps, *.pdf) should be included.

5. Because this process takes a long time, please find all these files in the **../Tutorial_Modeling /PsiPred_Shaker** folder provided for this tutorial. Open the ‘d51a473c-...Shaker.psipass2’ and ‘d51a473c-...Shaker.psipred.pdf’ files.

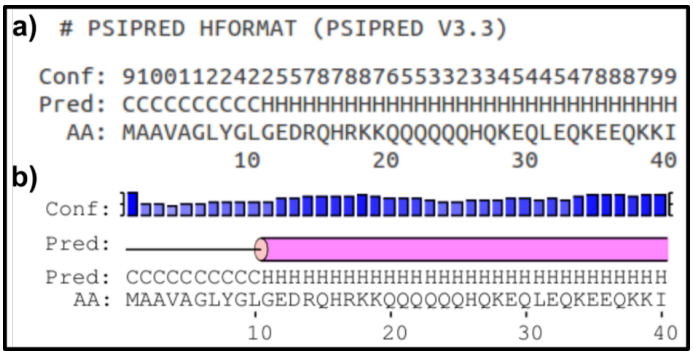


Figure 1: PsiPred secondary structure prediction **(a)** PSIPRED results in plain text format. **(b)** PDF version of the PSIPRED diagram.

As is shown in the *.psipass2 file (Figure 1a as an example), each residue has been assigned to either helix (H), strand (E), or coil (C) secondary structure elements. Along with this assignment a confidence level on the scale from 0=low to 9=high is also reported.

In addition, the *.pdf file (Figure 1b as an example) shows a PSIPRED output produced by PSIPRED View — a Java visualization tool that produces two-dimensional graphical representations of PSIPRED predictions.

6. Use the following script to convert the *.psipass2 file into fasta format (after making sure that you are in the working directory **../Tutorial_Modeling**):

```
./scripts/psipred2fa.pl -s PsiPred_Shaker/*.psipass2
```

7. An output file called ‘psipred.fa’ should have been created. Now, combine this secondary structure prediction with the target sequence using:

```
cat Shaker.fasta psipred.fa > Shaker+ss.fasta
```

3 Predict the transmembrane segments in the target sequence

A fundamental aspect of the structure of membrane proteins is the number of transmembrane segments and their orientation in the membrane. Fortunately, the physicochemical constraints imposed by the lipid environment over the membrane proteins provide a simple method to predict their topology. TOPCONS (`http://topcons.cbr.su.se/`) is a widely used web server for consensus prediction of membrane protein topology. Given the amino acid sequence of a putative alpha-helical transmembrane (TM) protein, TOPCONS predicts the topology of the protein, i.e. a specification of the membrane spanning segments and their in/out orientation relative to the membrane.

1. Insert the FASTA format sequence for *Shaker* in the input field (copy the sequence from the 'Shaker.fasta' file). You can either paste the sequence in the text-area provided, or, alternatively, upload a file with the sequence.

2. You can optionally provide a name for your TOPCONS job ("Job name"), e.g. 'TopCons_Shaker', and your email address to have the results.

3. Press "Submit" to run the job.

4. The results of your prediction will appear on the screen. Click the 'query.result.txt' link and save the result in your working directory (**../Tutorial_Modeling**) using the 'TopCons_Shaker.txt' name.

5. In a text editor, from the TopCons_Shaker.txt' file copy the TOPCONS prediction (just TOPCONS predicted topology) into the 'Shaker+ss.fasta' file with the following format:

```
>sp|P08510|KCNAS_DROME Potassium voltage-gated channel prote
MAAVAGLYGLGEDRQHRKKQQQQQQHQKEQLEQKEEQKKIAERKLQLREQQLQRNSLDGY
>Psipred
CCCCCCCCCCHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHCCCCCCC
>TOPCONS predicted topology:
iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii
```

6. Save the modified file using the 'Shaker+ss+tm.fasta' name, which must contain the target sequence along with its secondary structure and transmembrane predictions.

4 Visualize the target sequence

Jalview is a free program developed for the interactive editing, analysis and visualization of sequence alignments. It can also work with sequence annotation, secondary structure information, phylogenetic trees and 3D molecular structures.

1. We will now use Jalview to visualize the target sequence along with its secondary structure and TM predictions. This software must be installed in your computer; therefore, you can load it using:

```
jalview &
```

2. Choose File → Input Alignment → From File. In the dialog box, ensure that the file format filter is "Fasta (.mfa, fastq, fasta, fa)". Choose the file 'Shaker+ss+tm.fasta' from your working directory.

3. The sequence should appear as one long continuous sequence. You can turn it off by choosing Format → Wrap.

4. You can also turn off the annotations display by choosing Annotations and disabling Show annotations.

5. Show the alignment column position scale by choosing Format and enabling Scale Above.

6. Select a colour scheme in the Colour menu, preferably Zappo.

You should see an alignment like that shown in Figure 2. From these predictions we can quickly notice that *Shaker* is a protein containing six transmembrane segments; therefore, a homologous protein with similar characteristics should be used as template in the homology modeling. For now, you can close the Jalview window.

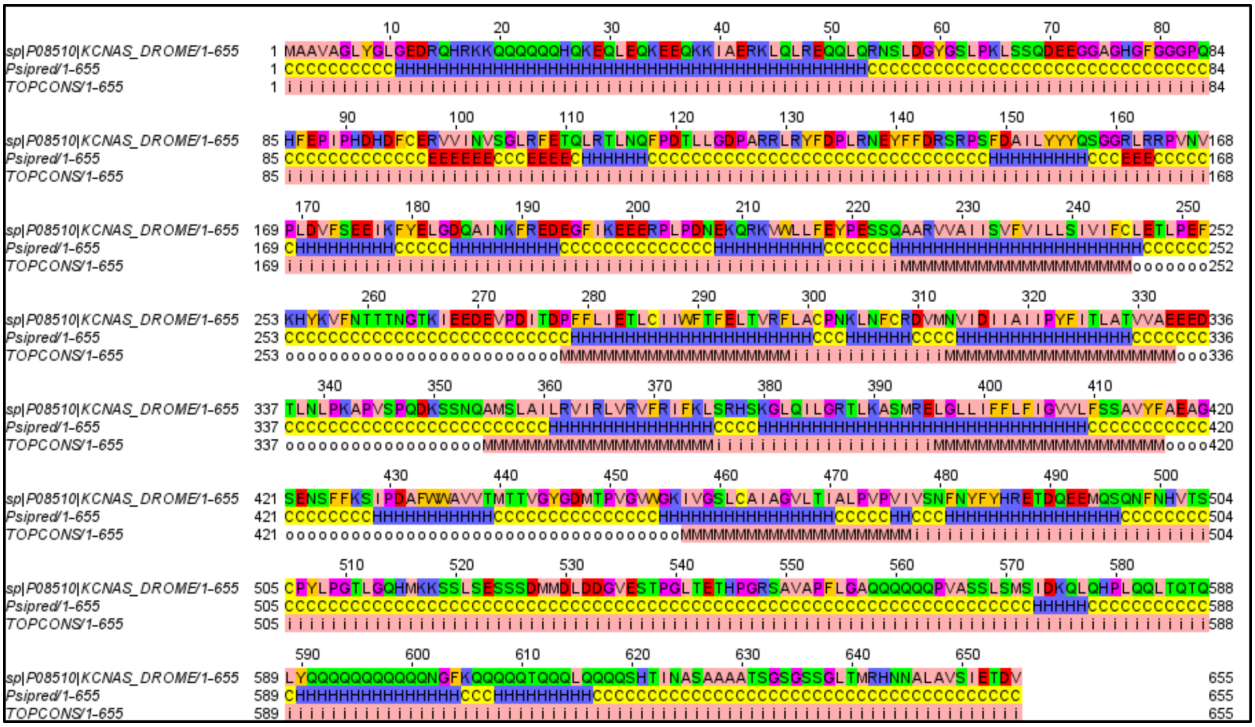


Figure 2: *Shaker* sequence along with its secondary structure (PSIPRED) and transmembrane segments (TOPCONS) predictions. The sequence was rendered using Jalview v2.8.2.

5 Search for structures that can be used as homology modeling template

Protein structure homology modeling relies on the evolutionary relationship between a target protein and a known structure protein that can be used as template. A critical step in homology modeling is the identification of a 3D template chosen by virtue of having the highest sequence identity with the target sequence, if indeed any are available. Blast and Psi-Blast are statistically driven search methods for detecting sequence similarities between a query sequence and a target database. Specifically, Psi-Blast is one of the most popular and powerful methods to search suitable templates of an unknown structure using the Protein Data Bank (PDB) database.

1. Go to the Blast web page (`http://blast.ncbi.nlm.nih.gov/`). Click on "protein blast" tab.

2. Insert the FASTA format sequence for *Shaker* in the enter query sequence box or upload a file with the sequence. The *Shaker* sequence is in the 'Shaker.fasta' file.

3. Provides a "Job Title" – e.g. 'Psipred_Shaker'.

4. Select as database the "Protein Data Bank proteins(pdb)" option.

5. Use "PSI-BLAST (Position-Specific Iterated BLAST)" as search algorithm.

6. Press "BLAST" to run the job.

The results will appear on the screen like is shown in Figure 3. In this case, the first hit suggested as template is 'Chain B, Mammalian Shaker Kv1.2 Potassium Channel- Beta Subunit Complex', which PDB ID is 2A79. Interestingly, the first hit provided by Psi-Blast search might not always be the most appropriate template. Because although such template can have a high identity with the target sequence, it might cover only one part of the structure of the protein. In addition, the resolution or quality of the template is also a factor to consider. In this tutorial, we will use the second hit "*Shaker* Potassium Channel Kv1.2 from *Rattus norvegicus*" (3LUT and chain B) as a template. The Kv1.2 sequence is ∼68% identical (Ident) with our target sequence, and ∼69% of *Shaker* residues align (Query cover) with Kv1.2. Basically, we have selected this template because is a full-length Shaker potassium channel Kv1.2 structure corresponding to a refinement of the crystal structure PDB ID 2A79.

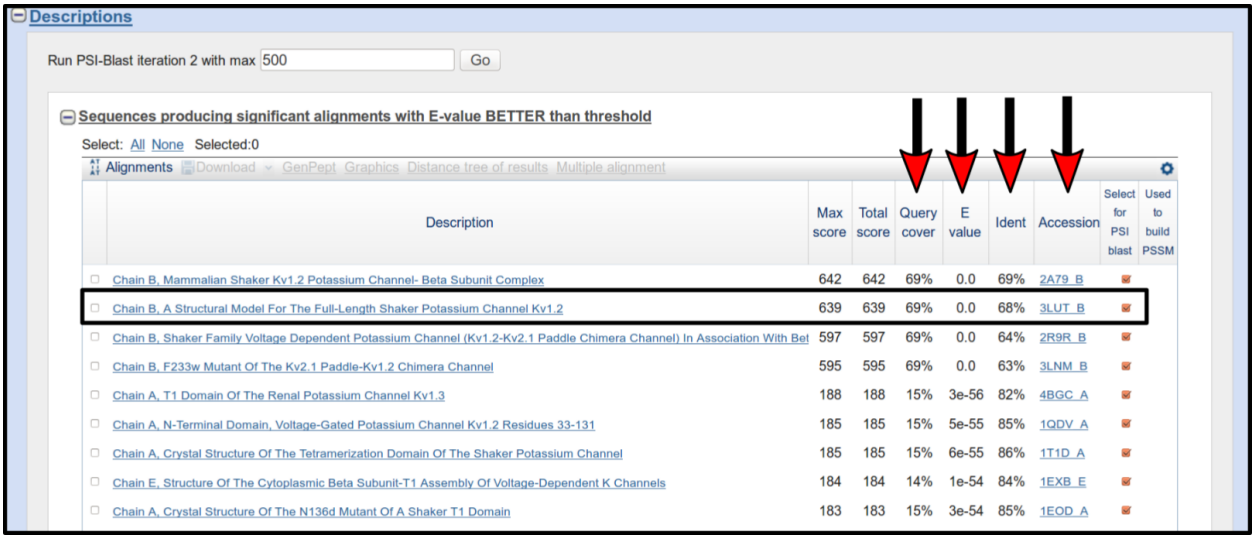


Figure 3: Results from templates search using PSI-BLAST. The most significant parameters are: "Query cover", percent of query sequence that is aligned; "E Value", number of matches with same score expected by chance (E values greater than 0.01 indicate poor hits); "Ident", sequence identity to the query; and "Accession", PDB accession code.

6 Obtain the atomic coordinates of the template protein structure

In order to use a protein structure as a template in the homology modeling, we first need to download its atomic coordinates.

1. At the Protein Data Bank web page (`http://www.rcsb.org/`) enter the PDB code '3LUT', and take a look at the results. What type of protein is it and what is its function? (If the function is related to that of the target protein, then this suggests that it may be good template structure). Was it structure solved by X-ray crystallography or NMR spectroscopy? What is its resolution? Do the template and the target protein belong to the same organism?

2. Download the coordinates of your template protein in the "Download Files" tab by clicking "PDB File (text)". Save the file in your working directory (**../Tutorial_Modeling**) with the '3LUT.pdb' name.

3. While you are at it, also download the fasta formatted sequence of the protein used in the crystallization. Click the "Display Files" tab and select "FASTA sequence". Save the file in your working directory (**../Tutorial_Modeling**) using the '3LUT.fasta' name.

7 Visualize the structure of the template protein

PyMOL is 3D molecular visualization software used to create, view, and analyze molecular-based data. You will now run the PyMol structure visualization software to inspect the structure of the template protein. This software must be installed in your computer.

1. Type the following command into your Unix prompt (after making sure that you are in the working directory):

   ```
   pymol 3LUT.pdb &
   ```

2. The PyMol visualization and command windows will now open. You can find more information and help on using PyMol at `http://pymol.sourceforge.net/newman/user/toc.html`. Please take some time to read the sections on "Manipulating the View" and "Getting Started with Commands", because these will help you with the later steps.

3. The initial view contains one monomer of the *Shaker* Potassium Channel Kv1.2 structure (PDB ID 3LUT) shown as lines. Change the view so that all of the structure is shown using cartoon, button 'S' on top right of viewer window (close to the '3LUT' tab) → cartoon . Colored according to the secondary structure, button 'C' → by ss.

4. If there is more than one protein chain visible, limit the view to just one chain by typing the following commands in the PyMol command prompt:

   ```
   hide line
   select chainB, chain B
   hide everything, not chainB
   center chainB
   ```

5. Identify the characteristic six transmembrane helices of one monomer of the potassium channels. Remembering that these are membrane-embedded, identify which parts of the protein that you expect to be surrounded by lipid, and which would be exposed to the aqueous solution either side of the membrane.

6. Now, take a look at the ligand bound to the protein. In this case we know that 3LUT has potassium ions and water. Type the following commands in the PyMol command prompt:

   ```
   select notprotein, resname K+HOH
   show sphere, notprotein
   ```

7. Take a picture of the '3LUT' template protein. First, change the background color by clicking Display → Background → White. Then, move the protein in the way you want and click File → Save Image As → PNG.

8 Create the template coordinate file for modeling

You will now create a version of the coordinate file that only includes the part of the protein that you need to use as a template.

1. We will select the chain B of the protein, from residue 150 to 421, and three potassium ions. To do this, try typing the following in the PyMol prompt:

   ```
   select template, chain B and resid 150-421 or (name K and resid 500+502+504)
   ```

2. A new group, that you have called "template", will appear on the right. Choose File → Save Molecule, select the "template" entry, and click on OK and Save. This file should be saved in your working directory (**../Tutorial_Modeling**).

In this tutorial, we first will model one monomer of the protein, and then, we will replicate the modeling to build the tetrameric *Shaker* potassium channel structure. To do this, we also need to create a tetrameric template using the biological assembly of 3LUT. The PDB coordinates file for a given structure does not always represent the biologically relevant assembly. Therefore, a good tool to download biological assemblies might be the Orientations of Proteins in Membranes (OPM) database (`http://opm.phar.umich.edu/`). It should be noted, this database is not precisely to download biological assemblies of proteins, but in this case will be useful. OPM provides a collection of transmembrane proteins from the Protein Data Bank whose spatial arrangements in the lipid bilayer have been calculated theoretically and compared with experimental data.

3. Go to the Orientations of Proteins in Membranes (OPM) database. On the right at the top of the page web, type '3LUT' and click "Search OPM".

4. The results will appear on the screen. Click the "Download Coordinates" tab under the figure. Save the file in your working directory (**../Tutorial_Modeling**), change the name to '3lut_opm.pdb'.

5. Open and visualize the file '3lut_opm.pdb' in PyMol by typing the following in the PyMol prompt (after making sure that you are in the working directory (**../Tutorial_Modeling**):

   ```
   hide everything, *
   load 3lut_opm.pdb
   show cartoon, 3lut_opm
   hide lines, 3lut_opm and not resname DUM
   select template_tetramer, 3lut_opm and chain K+I+L+J and resid 150-421
   ```

6. A new group, that you have called "template_tetramer", will appear on the right. Save this selection by choosing File → Save Molecule, select the "template_tetramer" entry. Click on OK and Save. This file should be saved in your working directory (**../Tutorial_Modeling**).

7. We will check our files and also do some important modifications. First, we will align both templates and then, we will change the chains name. New files will be generated. Type the following in the PyMol prompt:

```
delete all
load template.pdb
load template_tetramer.pdb
align template and name CA, template_tetramer and name CA
show cartoon, *
hide line, *
show spheres, template and resname K
alter (template_tetramer and chain J) , chain = 'A'
alter (template_tetramer and chain I) , chain = 'B'
alter (template_tetramer and chain K) , chain = 'C'
alter (template_tetramer and chain L) , chain = 'D'
save template_align.pdb, template
save template_tetramer_align.pdb, template_tetramer
```

8. Now, we have templates ready to use (like is shown in Figure 5). If you wish to view the template PDB structure again, save your session by clicking File → Save Session As, select a name (e.g. 'templates.pse') and click Save. You can then reload this session file in PyMol. For now, you can close the PyMol window.

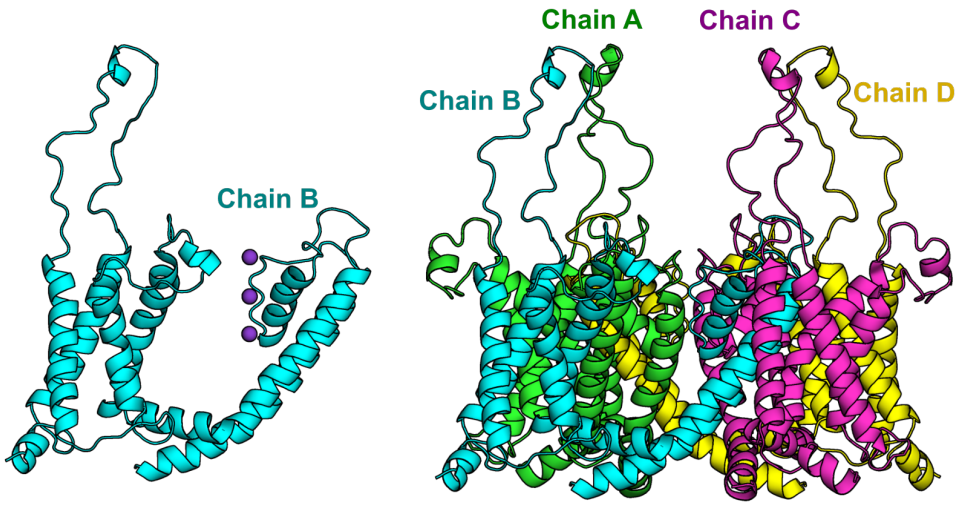


Figure 4: Structures to be used as template in the *Shaker* modeling. At left is shown the monomer template, including potassium ions. At right is shown the tetramer template (chains A, B , C and D). Chain B in the monomer has exactly the same position than Chain B in the tetramer structure.

9 Obtain the precise fasta sequence of the template file

For building a homology model we will actually need to make the alignment between the target sequence and the full-length wild-type template protein, which is not necessarily the same as the sequence of the residues in the PDB file. For example, there are often insertions, deletions or mutations in structures generated using X-ray crystallography due to the use of specific constructs that crystallized properly. Thus, the sequence of the protein used for crystallization may be different from the wild-type sequence or even from the sequence of the residues in the template PDB that you

downloaded from the Protein Data Bank (3LUT.fasta). Therefore, it is suggested to use the full-length wild-type sequence of the template to build the initial alignment with the target sequence; then, if necessary, you can modify the alignment with the residues that are actually included in the PDB file (you will see an example in section (15)).

1. Open the '3LUT.fasta' file downloaded from PDB web page, using a text editor (e.g. vi or gedit). Delete the sequence corresponding to chain A, and save a new file using the name '3LUT_chainB.fasta'.

2. Go to the UniProt web page (`http://www.uniprot.org/blast/`) to search the full-length wild-type sequence of the *Shaker* Potassium Channel Kv1.2 from *Rattus norvegicus*.

3. Enter the protein sequence from the '3LUT_chainB.fasta' file into the form field. Click the "Run BLAST" button.

4. Click on the entry "P63142" (Potassium voltage-gated channel subfamily A member 2 (Rattus norvegicus)). Go to the "Format" tab, select FASTA (canonical) and save the file in your working directory (**../Tutorial_Modeling**) with the 'Kv12.fasta' name.

10 Identify secondary structure and transmembrane segments in the template structure

We need to know the secondary structure assignment and transmembrane segments of the template protein, which will be used to guide the sequence alignment. The DSSP program (`http://www.cmbi.ru.nl/dssp.html`) works by calculating the most likely secondary structure assignment given the 3D structure of a protein. DSSP extracts out the residue names, and at the same time, the secondary structure.

1. Go to the DSSP server. Select the 'template_align.pdb' file from your working directory (**../Tutorial_Modeling**) and press "Calculate DSSP". Save the result in your working directory with the 'PDB2DSSP' name.

2. Use the following script to convert the PDB2DSSP file into fasta format:

   ```
   ./scripts/dssp_2_fa.pl PDB2DSSP template_align+ss.fasta
   ```

3. Now we are going to identify the transmembrane segments in the template protein. Go again to the Orientations of Proteins in Membranes (OPM) database (`http://opm.phar.umich.edu/`). Type '3LUT' and click "Search OPM". The results will appear on the screen.

4. See the results on the bottom. The six transmembrane segments of the Kv1.2 channel are detailed (e.g. Tilt: 18° - Segments: 1(164-182), 2(222-241), 3(254-270), 4(294-310), 5(328-346), 6(386-406)). From the information given, select the first transmembrane subunit (first line) and create a text file with the following format:

```
TM pdb-s pdb-e
1 B 164 182
2 B 222 241
3 B 254 270
4 B 294 310
5 B 328 346
6 B 386 406
```

5. Each line contains the residue range for one of the transmembrane segments, and a chain name (which is "B" in our template). Save the file as 'template_tm.txt'.

6. Now run the following script to create a transmembrane segments assignment file into fasta format (after making sure that you are in the working directory (**../Tutorial_Modeling**):

   ```
   ./scripts/tmpred2fa.pl -f fulltm -l T -s Kv12.fasta -t template_tm.txt
   ```

7. A file called 'fulltm.fa' should have been created, which contains the full-length wild-type sequence of the template protein as well as a representation of the transmembrane segments in fasta format. Remember if you can not do any step, these files are in the **../Tutorial_Modeling/example_files** folder provided for this tutorial.

11 Create an alignment between the target and template

In order to build a homology model using the template structure that we have just obtained, we will need an alignment between the sequence of the template protein and the sequence of the target protein. One way to do this is to generate a pairwise sequence alignment using the AlignMe server (`http://www.bioinfo.mpg.de/AlignMe`). AlignMe provides a user-friendly interface for a set of sequence alignment approaches specifically tuned to membrane proteins.

1. Go to the AlignMe web page and select the "Sequence to Sequence Alignment" tab. From the working directory (**../Tutorial_Modeling**), upload the sequence files for the target ('Shaker.fasta') and template ('Kv12.fasta') protein into the two sequence boxes.

2. Then, in α-helical membrane proteins section, click on "Most accurate alignments for very closely related proteins (>45% identity)". It is because we saw in the "Search for structures that can be used as homology modeling template" section that *Shaker* and Kv1.2 have ∼68% of identity (see Figure 3).

3. Enter an email address to receive the results and click on "Submit". Because this process takes a long time, please find the AlignMe results in the **../Tutorial_Modeling/AlignMe** folder provided for this tutorial.

We will now analyze the alignment in more detail, and check where the secondary structure and transmembrane segments are in the template, or are predicted to be in the target sequence.

4. To do this we will need the AlingMe alignment in fasta format, so run the following script to convert the 'AlignMe_aligned_sequences_of_19954_1445729542.aln' file into fasta format (after making sure that you are in the working directory (**../Tutorial_Modeling**):

   ```
   ./scripts/clustal2fa.pl AlignMe/*.aln align_shaker_kv12.fasta
   ```

Remember if you can not do any step, these files are in the **../Tutorial_Modeling/example_files** folder provided for this tutorial.

12 Combine the alignment with the secondary structure and transmembrane segments definitions

In sections (2), (3) and (10), you calculated the secondary structure (SS) and transmembrane segments (TM) definitions for the template and target protein. So, to combine the TM and SS definitions together with the AlignMe alignment, we need to do a few steps. Be careful that for each step, all the inputs and outputs are in the working directory (**../Tutorial_Modeling**).

1. From your working directory, run the following command:

   ```
   cat fulltm.fa template_align+ss.fasta align_shaker_kv12.fasta Shaker+ss+tm.fasta
   > align+ss+tm.fasta
   ```

2. Now, we will use Jalview to edit the sequences. Load Jalview using:

   ```
   jalview &
   ```

3. Choose File → Input Alignment → From File. In the dialog box, ensure that the file format filter is "Fasta (.mfa, fastq, fasta, fa)". Choose the file 'align+ss+tm.fasta' from your working directory (**../Tutorial_Modeling**).

4. Now, to combine the TM and SS definitions with the AlignMe alignment, you can select and move the sequences by holding [SHIFT] and the left-click of the mouse. Use Figure 5 as a reference to move the sequences. The point here is build an alignment with the secondary structure and transmembrane segments definitions matching with the full-length target and template sequence, allowing us to compare them easily. You can find more information and help on using Jalview at `http://www.jalview.org/tutorial/TheJalviewTutorial_screen.pdf`.

5. Because this step takes a long time, please find the 'align+ss+tm_jalview.fasta' file with the whole alignment in the **../Tutorial Modeling/** folder provided for this tutorial.

6. Load this file into Jalview (File → Input Alignment → From File), and study the alignment. Note the difference (if any) between the template sequences. Compare the TM predictions (M) with the known TM regions (T). Are they in general the same regions? Similarly, compare the predicted secondary structure (Psipred) with the known secondary structure (2nd_struct). Now, check the pairwise alignment between template and target: are there any gaps in transmembrane or secondary structure regions?

7. If you think that these results suggest changes to the pairwise target-template sequence alignment, you should do those changes. In this case, due to the high identity (∼68%) between the target and template proteins not changes are suggested.

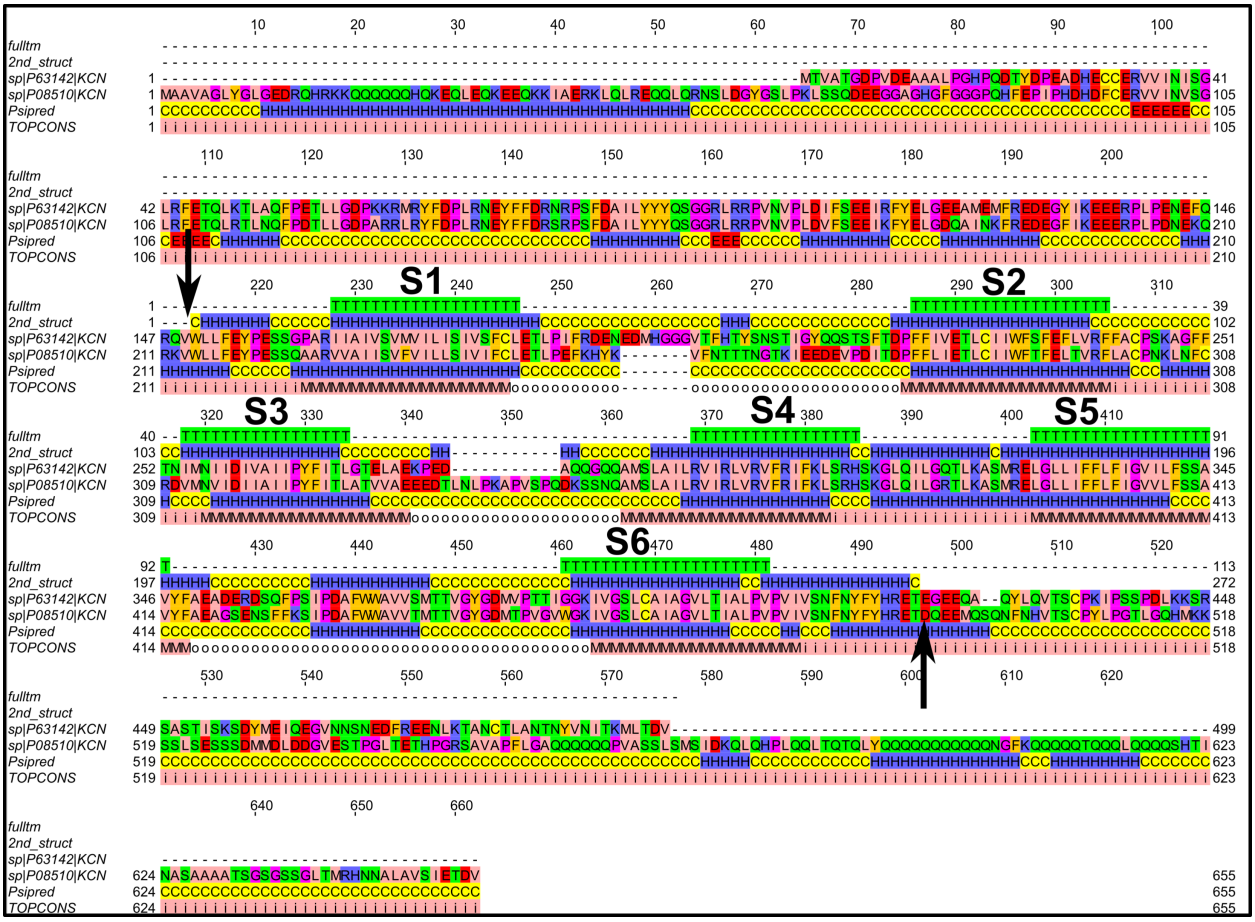


Figure 5: Pairwise sequence alignment generated with the AlignMe server. Secondary structure and transmembrane segments definitions are shown to verify the alignment. Black arrows indicate the region of the alignment that we will be used for the modeling. Six transmembrane segments are identified both in target and template sequence.

13 Prepare the input alignment file for homology modeling

In section (11), we built a pairwise sequence alignment with the full-length wild-type sequences of the template and target protein. This is your target-template alignment, but we need to edit it slightly for model building. For example, we do not really want to model any long N- and C-terminal domains of *Shaker* (residues 1 to 213 and 490 to 655), because we are interested in studying the trasmembrane region of the protein. To remove those residues from the target-template alignment do the following:

1. From your working directory (**../Tutorial_Modeling**), load the 'align_shaker_kv12.fasta' file into Jalview (File → Input Alignment → From File).

2. We will cut the sequence alignment; first, from position 497 to 662 (C-terminal domain). To do this scroll along to find position 496 (see the black arrows in Figure 5 for reference), and click on the white space above it. A red square will appear. Then choose Edit → Remove right to delete all the C-terminal residues.

3. Do the same to delete the N-terminal domain, from position 1 to 213 in the alignment. Find position 214 (see the black arrows in Figure 5 for reference), and click on the white space above

it. A red square will appear. Then choose Edit → Remove left to delete all the N-terminal residues.

4. Now we will change the name of the sequence to something simpler, so right-click on the sequence title of each sequence, and then again on the name in the list, until you have the option to Edit Name/Description. Change the name from 'sp|P63142|KCN' to 'template_align/150-421' and from 'sp|P08510|KCN' to 'Shaker/214-489'. Note that the start and end residues need to be given.

5. Once you have the sequence names formatted correctly, and the termini deleted, save this pairwise alignment in PIR format, which is compatible with the homology modeling program Modeller. Choose File → Save as, ensure that the file format filter is "PIR (.pir)". Save the modified alignment in your working directory (**../Tutorial_Modeling**) with the 'shaker_on_kv12.pir' name.

6. From your working directory (**../Tutorial_Modeling**), open the 'shaker_on_kv12.pir' file using a text editor (e.g. vi or gedit). Modify this file according to the following example, which is required to work with Modeller. The format of the sequence alignment file is very important. You can find further instructions at `www.salilab.org/modeller/manual`.

```
>P1;template_align
structure:template_align:150:B:504:B::::
WLLFEYPESSGPARIIAIVSVMVILISIVSFCLETLPIFRDENEDMHGGGVTFHTYSNSTIGYQQSTSFTDP
FFIVETLCIIWFSFEFLVRFFACPSKAGFFTNIMNIIDIVAIIPYFITLGTELAEKPED----------AQ
QGQQAMSLAILRVIRLVRVFRIFKLSRHSKGLQILGQTLKASMRELGLLIFFLFIGVILFSSAVYFAEADER
DSQFPSIPDAFWWAVVSMTTVGYGDMVPTTIGGKIVGSLCAIAGVLTIALPVPVIVSNFNYFYHRET/...*
>P1;Shaker
sequence:Shaker:214:B::B::::
WLLFEYPESSQAARVVAIISVFVILLSIVIFCLETLPEFKHYK-------VFNTTTNGTKIEEDEVPDITDP
FFLIETLCIIWFTFELTVRFLACPNKLNFCRDVMNVIDIIAIIPYFITLATVVAEEEDTLNLPKAPVSPQDK
SSNQAMSLAILRVIRLVRVFRIFKLSRHSKGLQILGRTLKASMRELGLLIFFLFIGVVLFSSAVYFAEAGSE
NSFFKSIPDAFWWAVVTMTTVGYGDMTPVGVWGKIVGSLCAIAGVLTIALPVPVIVSNFNYFYHRET/...*
```

7. The most important changes are highlighted in blue. Note that the start and end number of the residues and the chains name of the template structure need to be given (you can see this information in the 'template_align.pdb' file). A line to detail the target sequence also is required. Please do not forget to include the start residue number and chain name of the model. The '/...' characters at the end of each sequence means that three potassium ions will be included in the final model. The '*' symbol at end of the template and target sequence also is necessary. Save the modified file in your working directory (**../Tutorial_Modeling**) with the 'alignment.pir' name.

14 Prepare all input files for homology modeling

MODELLER is a known program for homology or comparative modeling of protein three-dimensional structures. The user only need to provide an alignment of a sequence to be modeled with known related structures and MODELLER automatically calculates a model containing all non-hydrogen atoms.

1. In order to build the *Shaker* homology model, in your working directory (**../Tutorial_Modeling**) create a new folder with the **../modeling/** name. You will need three input files:

   - The template structure with PDB-format coordinates ('template_align.pdb')
   - The pairwise sequence alignment between the target and the template in PIR format ('alignment.pir').
   - A command file containing the instructions for the modeling ('do_model.py'). This file is included in the **../Tutorial_Modeling/** folder provided for this tutorial.

2. Please view the 'do_model.py' file in a text editor, and check that the input parameters are all consistent with your files. The 'do_model.py' file should have the following format:

```
################################
# define the modeling parameters
class mymodel(automodel):
    def special_restraints(self, aln):
        self.rename_segments (segment_ids=('B'),renumber_residues=(214))

        rsr = self.restraints
        at = self.atoms
        # Restraints for secondary structure
         # (includes all helices defined in original pdb; modifiable)
        #rsr.add(secondary_structure.alpha(self.residue_range('44:A', '54:A')))

        # Restrains for subtrates


    def user_after_single_model(self):
        self.rename_segments (segment_ids=('B'),renumber_residues=(214))

a = mymodel(env,
              alnfile  = 'alignment.pir', # alignment filename
              knowns   = ('template_align'), # codes of templates
              sequence = ('Shaker'))  # code of the target

a.starting_model= 1  # index of the first model
a.ending_model  = 3  # index of the last model (determines how many models to calculate)
a.deviation = 5.0    # deviation in models
#a.md_level = refine.very_slow
a.make()
```

3. The most important parameters are highlighted in blue. Looking in this file: the start residue number and chain name of the model, the alignment filename, code of template and target sequence in the 'alignment.pir' file, and the number of models that will be generated.

4. Copy all three input files ('template_align.pdb', alignment.pir' and 'do_model.py') into the directory **../modeling/**.

15 Build the homology model for one monomer

To run the Modeller program (which should be installed in your computer with the mod9.15 name), change to the **../modeling/** directory.

1. Use the following Unix command:

   ```
   mod9.15 do_model.py
   ```

2. Do you have some error? You should have a message "Sequence difference between alignment and pdb :". Open the 'do_model.log' file and check the error. Here there is an example of what you have to check:

```
Alignment   FHTYSNSTIGYQQSTSFTDPFFIVETLCIIWFSFEFLVRFFACPSKAGFFTNIMNIIDIVA
      PDB   FHTYSQSTIGYQQSTSFTDPFFIVETLCIIWFSFEFLVRFFACPSKAGFFTNIMNIIDIVA
    Match   ***** **********************************************************
```

3. As it was explained in section (9), the sequence of the protein used for crystallization may be different from the wild-type sequence. This is a clear example. Therefore, change the asparagine (N) residue in the 'alignment.pir' file (in the template protein sequence) to glutamine (Q). For example:

```
>P1;template_align
structure:template_align:150:B:504:B::::
WLLFEYPESSGPARIIAIVSVMVILISIVSFCLETLPIFRDENEDMHGGGVTFHTYSQSTIGYQQSTSFTDP
FFIVETLCIIWFSFEFLVRFFACPSKAGFFTNIMNIIDIVAIIPYFITLGTELAEKPED----------AQ
QGQQAMSLAILRVIRLVRVFRIFKLSRHSKGLQILGQTLKASMRELGLLIFFLFIGVILFSSAVYFAEADER
DSQFPSIPDAFWWAVVSMTTVGYGDMVPTTIGGKIVGSLCAIAGVLTIALPVPVIVSNFNYFYHRET/...*
```

4. Run Modeller again. This will probably take a few minutes to calculate. There are several output files (descriptions of these files can be obtained from `hwww.salilab.org/modeller/manual`). We are interested in the atomistic models of *Shaker* with the .pdb extension; but first, check the end of the log file 'do_model.log': which of the models has the lowest molpdf and DOPE scores? Here there is an example of what you have to check:

```
>> Summary of successfully produced models:
Filename                          molpdf
----------------------------------------
Shaker.B99990001.pdb          1794.45776
Shaker.B99990002.pdb          1509.32629
Shaker.B99990003.pdb          1634.86328
```

16 Build the homology model for the tetramer

Now, to build the *Shaker* homology model as a tetramer, in your working directory (**../Tutorial_Modeling**) create a new directory **../modeling_tetramer/**. For this modeling, you will also need three input files.

1. In a text editor, open the 'alignment.pir' file (where you changed asparagine to glutamine). This file should be in the **../Tutorial_Modeling/modeling/** directory.

2. Copy three times more each target and template sequence in this file, not including the potassium ions. You should generate a file as following:

```
>P1;template_tetramer_align
structure:template_tetramer_align:150:B:421:D::::
WLLFEYPESSGPARIIAIVSVMVILISIVSFCLETLPIFRDENEDMHGGGVTFHTYSQSTIGYQQSTSFTDP
FFIVETLCIIWFSFEFLVRFFACPSKAGFFTNIMNIIDIVAIIPYFITLGTELAEKPED-----------AQ
QGQQAMSLAILRVIRLVRVFRIFKLSRHSKGLQILGQTLKASMRELGLLIFFLFIGVILFSSAVYFAEADER
DSQFPSIPDAFWWAVVSMTTVGYGDMVPTTIGGKIVGSLCAIAGVLTIALPVPVIVSNFNYFYHRET/
WLLFEYPESSGPARIIAIVSVMVILISIVSFCLETLPIFRDENEDMHGGGVTFHTYSQSTIGYQQSTSFTDP
FFIVETLCIIWFSFEFLVRFFACPSKAGFFTNIMNIIDIVAIIPYFITLGTELAEKPED-----------AQ
QGQQAMSLAILRVIRLVRVFRIFKLSRHSKGLQILGQTLKASMRELGLLIFFLFIGVILFSSAVYFAEADER
DSQFPSIPDAFWWAVVSMTTVGYGDMVPTTIGGKIVGSLCAIAGVLTIALPVPVIVSNFNYFYHRET/
WLLFEYPESSGPARIIAIVSVMVILISIVSFCLETLPIFRDENEDMHGGGVTFHTYSQSTIGYQQSTSFTDP
FFIVETLCIIWFSFEFLVRFFACPSKAGFFTNIMNIIDIVAIIPYFITLGTELAEKPED-----------AQ
QGQQAMSLAILRVIRLVRVFRIFKLSRHSKGLQILGQTLKASMRELGLLIFFLFIGVILFSSAVYFAEADER
DSQFPSIPDAFWWAVVSMTTVGYGDMVPTTIGGKIVGSLCAIAGVLTIALPVPVIVSNFNYFYHRET/
WLLFEYPESSGPARIIAIVSVMVILISIVSFCLETLPIFRDENEDMHGGGVTFHTYSQSTIGYQQSTSFTDP
FFIVETLCIIWFSFEFLVRFFACPSKAGFFTNIMNIIDIVAIIPYFITLGTELAEKPED-----------AQ
QGQQAMSLAILRVIRLVRVFRIFKLSRHSKGLQILGQTLKASMRELGLLIFFLFIGVILFSSAVYFAEADER
DSQFPSIPDAFWWAVVSMTTVGYGDMVPTTIGGKIVGSLCAIAGVLTIALPVPVIVSNFNYFYHRET*
>P1;Shaker_tetramer
sequence:Shaker_tetramer:214:A::D::::
WLLFEYPESSQAARVVAIISVFVILLSIVIFCLETLPEFKHYK-------VFNTTTNGTKIEEDEVPDITDP
FFLIETLCIIWFTFELTVRFLACPNKLNFCRDVMNVIDIIAIIPYFITLATVVAEEEDTLNLPKAPVSPQDK
SSNQAMSLAILRVIRLVRVFRIFKLSRHSKGLQILGRTLKASMRELGLLIFFLFIGVVLFSSAVYFAEAGSE
NSFFKSIPDAFWWAVVTMTTVGYGDMTPVGVWGKIVGSLCAIAGVLTIALPVPVIVSNFNYFYHRET/
WLLFEYPESSQAARVVAIISVFVILLSIVIFCLETLPEFKHYK-------VFNTTTNGTKIEEDEVPDITDP
FFLIETLCIIWFTFELTVRFLACPNKLNFCRDVMNVIDIIAIIPYFITLATVVAEEEDTLNLPKAPVSPQDK
SSNQAMSLAILRVIRLVRVFRIFKLSRHSKGLQILGRTLKASMRELGLLIFFLFIGVVLFSSAVYFAEAGSE
NSFFKSIPDAFWWAVVTMTTVGYGDMTPVGVWGKIVGSLCAIAGVLTIALPVPVIVSNFNYFYHRET/
WLLFEYPESSQAARVVAIISVFVILLSIVIFCLETLPEFKHYK-------VFNTTTNGTKIEEDEVPDITDP
FFLIETLCIIWFTFELTVRFLACPNKLNFCRDVMNVIDIIAIIPYFITLATVVAEEEDTLNLPKAPVSPQDK
SSNQAMSLAILRVIRLVRVFRIFKLSRHSKGLQILGRTLKASMRELGLLIFFLFIGVVLFSSAVYFAEAGSE
NSFFKSIPDAFWWAVVTMTTVGYGDMTPVGVWGKIVGSLCAIAGVLTIALPVPVIVSNFNYFYHRET/
WLLFEYPESSQAARVVAIISVFVILLSIVIFCLETLPEFKHYK-------VFNTTTNGTKIEEDEVPDITDP
FFLIETLCIIWFTFELTVRFLACPNKLNFCRDVMNVIDIIAIIPYFITLATVVAEEEDTLNLPKAPVSPQDK
SSNQAMSLAILRVIRLVRVFRIFKLSRHSKGLQILGRTLKASMRELGLLIFFLFIGVVLFSSAVYFAEAGSE
NSFFKSIPDAFWWAVVTMTTVGYGDMTPVGVWGKIVGSLCAIAGVLTIALPVPVIVSNFNYFYHRET*
```

3. The most important changes are highlighted in blue. Please do not forget the '*' symbol at end of template and target sequence, and '/' to separate each monomer (we have four repeated sequences, because our protein is a tetramer). Save the new alignment file in your working directory (**../Tutorial_Modeling**) using the 'alignment_tetramer.pir' name.

4. A command file containing the instructions for the tetramer modeling ('do_model_tetramer.py') is included in the **../Tutorial_Modeling/** folder provided for this tutorial.

5. Please view the 'do_model_tetramer.py' file in a text editor, and check that the file names are all consistent. This file is very similar to the 'do_model.py' file that we analyzed previously.

6. Copy the 'alignment_tetramer.pir', 'template_tetramer_align.pdb' and 'do_model_tetramer.py' files into the directory **../modeling_tetramer/**.

7. From the **../modeling_tetramer/** directory, run Modeller using the following Unix command:

```
mod9.15 do_model_tetramer.py
```

17 Visualize the homology models of *Shaker*

Finally, we will check the models in Pymol. Change to your working directory **../Tutorial_Modeling/** and load Pymol:

```
pymol &
```

1. Now, open the models (monomer and tetramer) and the templates PDB file by typing the following commands in the PyMol command prompt:

```
load template_align.pdb
load template_tetramer_align.pdb
load modeling/Shaker.B99990002.pdb
load modeling_tetramer/Shaker_tetramer.B99990001.pdb
show cartoon, *
hide line, *
align Shaker.B99990002, template_align
align Shaker_tetramer.B99990001, template_tetramer_align
```

2. All structures are now aligned. How do the models of **Shaker** differ from the structure of the templates? What do you think about the quality of the models?

18 Check the quality of the models

In order to increase the chances that the sampling creates a model that satisfies all of the input restraints from the template, 2000 iterations of model building are suggested. You can define the number of models in the 'a.ending_model = ' line of the *.py input file of Modeller. Now, the models can be evaluated using MolPDF and ProQM scores, as well as Procheck analysis.

- The MolPDF score describes how well the model satisfies the input restraints, including those created from the template, the alignment, and any additional applied restraints, and is therefore an arbitrary value dependent on those features; the model with the smallest MolPDF score best satisfies all the restraints. This value is obtained from the Modeller *.log output file (e.g do_model.log).

- The ProQM score measures the degree to which a set of coordinates is consistent with a number of required features, including TM segments predicted using TOPCONS, the distance to the membrane center for residues in α-helical segments predicted with ZPRED, secondary structure elements predicted by PSIPRED, and others. The ProQM score is assigned as a 21-residue window-average to each residue, and is then summed to give a total score per model, with values ranging from 0 to 1, where values of 0.7 are typical of membrane protein structures solved by X-ray crystallography. You can calculate the ProQM score of your model on the following link: `http://www.bioinfo.ifm.liu.se/ProQM/index.php`.

- Procheck assesses the degree to which the features of the model are consistent with those of known protein structures in terms of bond distances, angles, dihedrals and overlapping atoms. Procheck can be used to identify the fraction of backbone groups that lie outside the favored regions of the Ramachandran plot. You can download the program from: `http://www.ebi.ac.uk/thornton-srv/software/PROCHECK/`.

Summary

This tutorial has hopefully given a real-life example of the basic steps required for building a homology model: analyzing the target sequence, identifying the template, computing an alignment, checking the alignment, building and analyzing the model. More advanced modeling would involve careful scoring of the model, optimization of the alignment between the template and target, perhaps guided by other experimental information, as well as addition of constraints according to experimental information, refinement of side-chain conformations and of loop regions. Ligands may need to be built in. It may also be possible to build models using more than one template, particularly if different regions of the sequence match better with different templates. In addition, the difficulty of the modeling (especially alignment) increases as the identity decreases between the target and available templates, so one can start doing iterative improvements to the alignments to see how the model changes.