

# GPCR-ModSim

## Tutorial I

April, 2016

Mauricio Esguerra, Jessica Sallander, Xabier Bello, David Rodríguez, Hugo Gutiérrez de Terán  
hugo.gutierrez@icm.uu.se

*A step by step tutorial on the use of [open.gpcr-modsim.org](http://open.gpcr-modsim.org) using as example the human Orexin receptor type 2 (uniprot id: [ox2r\\_human](http://www.uniprot.org/uniprot/ox2r_human)).*

### - WORK FLOW

A descriptive flowchart of the whole procedure that this tutorial will follow is shown in Figure 1.

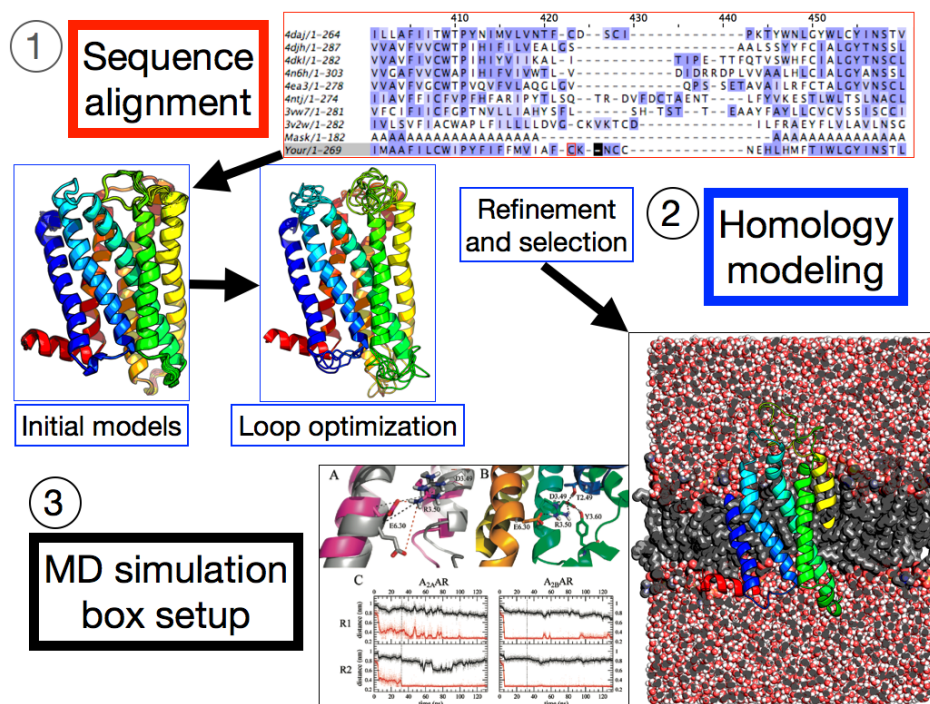


Figure 1: Workflow of one way to interact with open.gpcr-modsim.org. 1) A sequence alignment is made to one of our curated multiple sequence alignments, 2) Initial homology models are created and loops are refined using Modeller [2], 3) A model is selected to continue with a Molecular Dynamics equilibration run in our local cluster using GROMACS [3].

### - TEST CASE

In this tutorial we will show you how to follow a step-by-step recipe to create models of the human Orexin receptor type 2 (uniprot id: [OX2R.HUMAN](http://www.uniprot.org/uniprot/OX2R.HUMAN)). This receptor was crystallized (PDB id: [4s0v](http://www.rcsb.org/pdb/entry/main.do?entry=4s0v)) in complex with suvorexant, a G Protein-Coupled Receptor (GPCR) **antagonist**, also known as a **blocker**.

Membrane proteins live in greasy environments and are traditionally difficult to crystallize. A common strategy to get the crystallization job done is to create so-called fusion proteins. This means that a part or parts of the GPCR, usually the extracellular or intracellular loops, are replaced (fused) with parts of another protein. The most successful fusion protein so far in crystallization attempts has been

T4-lysozyme. Nonetheless, for the case of the Orexin receptor this fusion partner did not succeed in creating crystals, so, the catalytic domain of *Pyrococcus abyssi* glycogen synthase (PGS) was used to replace the third intracellular loop (ICL3) and microcrystallization was achieved [1]. Since the ICL3 is replaced with PGS, then this part is not observed in the crystal structure, nor are other parts which are shortened for crystallization purposes, such as a fragment of the N-terminus tail, the C-terminus tail, as well as parts of the solvent exposed region in the second extracellular loop (ECL2).

Following this tutorial you will generate models for the sequence which starts and ends just like the corresponding Orexin receptor crystal structure, but you are going to include regions which are missing in the crystal structure. The sequence that you will use in this tutorial is the following:

```
PKEYEWVLIAGYIIIVFVVALIGNVLVCVAVWKNHHMRTVTNYFIVNLSLADVLVTITCLPATLVVDITETWFFGQSLCKVIPYLQT  
VSVSVSVLTLSLSCIALDRWYAICHPLMFKSTAKRARNISIVIIWIVSCIIMIPQAIVMECSTVFPGLANKTTLFTVCDERWGGEIYPK  
MYHICFFLVITYMAPLCLMVLAYLQIFRKLWCRQKQIRARRKTARMLMVLLVFAICYLPISILNLVKRVFGMFAHTEDRETVYAWF  
TFSHWLVIYANSAANPIIYNFLSGKFREEFKAAFSC
```

For the case of the Orexin 2 receptor the crystal structure is already known, and we would like to model this receptor as if the structure was unknown (by not using it as a template). Hence, this tutorial serves as an exercise on how to model GPCR's of unknown crystal structures, and you are able to compare your obtained models to the experimental crystal structure.

## - PROCEED!

In the following Figures you will see snapshots of the website which you can follow along with a step-by-step guide which tells you what to do, and most importantly, why you're doing it.

### 1. ENTER YOUR DATA OR CREATE AN ACCOUNT:

The latest version of [open.gpcr-modsim.org](https://open.gpcr-modsim.org) doesn't require a password. You can simply click on the navigation tab "**Model a GPCR**" and enter your GPCR sequence or a 3D structure in PDB format to do your homology or Molecular Dynamics (MD) modeling. Nonetheless, we recommend for users to create an account, as their results in the passwordless interface will only be stored for four days, and their results in the registered version will be stored for a month or more if our disk-space allows. To submit a sequence just click in the navigation toolbar "Model a GPCR".

### 2. RETRIEVE OR INPUT A SEQUENCE:

The server allows you to input protein sequence information in the **FASTA sequence** box either as a UniprotID (in this case, you should type: **O43614**) or using the [fasta](#) format. If you were to type **O43614** in the *FASTA sequence* box the webserver will connect in the background to the uniprot database ([www.uniprot.org](https://www.uniprot.org)) and download the fasta sequence for you. In this tutorial we will use the fasta format to input our sequence, the header line (starting with an optional > sign) is used to give a name to the sequence that is sent to the alignment program in our cluster, so at a later stage you will see this name in the alignment between your sequence and the MSA profile. The **Name** box is used to give the project a name, in our example we are naming it **orexin2model1** (see Figure 2).

- Tip 1: If no header line is given along with the fasta sequence you enter, then it will be assigned the header "Your sequence" automatically.
- Tip 2: In this example we have extracted the fasta sequence corresponding to regions observed in the crystal structure of *ox2r\_human* as indicated in its corresponding PDB entry as shown in Figure 2. This will make it easy to compare between the obtained models, and the experimental crystal structure.

You can now proceed to select the template set (*inactive, active, or active-like*) you want to use for performing the initial alignment from our curated multiple sequence alignments (MSA) by clicking on

Start a new project

Name

FASTA sequence or Uniprot ID [\[help\]](#)

```
>orexin2model1
PKEYEWVLIAGYIIVFVVALIGNVLVCVAWKNHHMRTVTNYFIVNLS
LADVLVTITCLPATLVVDITETWFFGQSLCKVIPYLQTVSVSVSLTSL
CIALDRWYAICHPLMFKSTAKRARNISIVIVSCIMIPQAIVMECSTV
PGLANKTTLFTVCDERWGGEIYPKMYHICFFLVTYMAPLCLMVLA
YLQIFRKLWCRQKQIRARRKTARMLMVLLVFAICYLPISILNVLKRIV
FGMFAHTEDRETVYAWFTFSHWLVYANSAANPIIYNFLSGKFREFEK
AAFSC
```

[Click here for a sample sequence](#)

Group

[News](#) [Project](#) [Team](#) [Publications](#) [Links](#) [Q&A](#) [Feedback](#) [About](#)

Figure 2: For this example we are using the sequence corresponding to the crystal structure **4s0v** and we are adding missing residues **LMFK** in the second intracellular loop (ICL2), **PGLAN** in the second extracellular loop (ECL2), and **TE** residues in the third extracellular loop (ECL3) as shown in the highlighted boxes. Note that this sequence is incomplete when compared to the full sequence as can be seen in the uniprot database (<http://www.uniprot.org/uniprot/043614.fasta>).

the **Group** area right after the **FASTA sequence** box at the lower part of the window, and then click on the **Submit** button. We are selecting the **inactive** set in this example.

### 3. CHECK YOUR ALIGNMENT AND SELECT A TEMPLATE:

The server now aligns your sequence with the curated multiple sequence alignment (MSA) which you selected from *active*, *active-like*, or *inactive*. This alignment can be inspected and manually edited through the Jalview applet (See Figure 3), or it can also be downloaded and edited with standalone sequence alignment programs such as Seaview, ClustalX or Aliview, to mention a few of the many programs available. When you check your alignment for this exercise you will see that in the region corresponding to the second extracellular loop a conserved cysteine (CYS 161) which forms a disulphide bridge with the beginning of TM3 (CYS 78), is misaligned. For this example we are deleting 17 gaps before the region containing CYS 161 and adding the same number of gaps after it. More detailed information is given in the **tips** of the **"GENERATE HOMOLOGY BASED MODELS"** section.

- Tip 1: You can manually modify the alignment using the jalview applet (see Figure 3) following the instructions shown online:
  - Click the applet and press F2 to enter the edit mode (a black rectangle should appear).
  - Space inserts a gap, del or backspace removes one.
  - Alt + cursor key moves the whole sequence.

- Remember to upload ("Save" button below) the edited sequence.

• [View/edit alignment.](#)  
(If unable to edit the alignment online, please [download](#) it, edit locally, and [upload](#) when ready).

### Edit your alignment

	250	260	270	280	290	300
4n6h/1-303	--AVTRPRD--	CAVVC--	MLQFP--	PSWYWD--		VTK
4ea3/1-278	--GSAQVED--	EEIECL--	VEIPT--	PQDYWC--		VFA
4ntj/1-274				KSEFGLVWH		EIV
3vw7/1-281	--KEQTIQVPC--	GITTC--	HDVLSSETLLE	GYA--		YF
3v2w/1-282			LSSCSTVLP	Y--HK		HYI
4yay/1-288	--NVFF--I--	ITVC--	AFHYE--	TLPI		GLC
4buo/1-304	--LQNL--SGDGT	PCGLVC--	TPIVD--	TATLK		VVI
4z35/1-290			IENC SNM	APL--Y--SD		SYL
Mask/1-182						A
Your/1-293	--ECSTVFPCLANKTT	LFTVC--	DERWC--		GEIYKMY	

Sequence 24 ID: Your Residue: CYS (161)

- **Note:** jalview has been tested to work in firefox, internet explorer, and safari using java 8.
- Click the above applet and press F2 to enter the edit mode (a black rectangle should appear).
- Space inserts a gap, del or backspace removes one.
- Alt + cursor key moves the whole sequence.
- Remember to upload ("Save" button below) the edited sequence.

If the applet is being blocked, click this.

Save

Saving OK!

Figure 3: The jalview applet allows you to edit your sequence alignment. You will need a recent version of Java 8 and the Firefox, Safari, or Internet Explorer web browser. Remember always to click on the **Save** button after you have edited your alignment and wait for a confirmation which should say "Saving OK". Notice how the alignment has been edited to have CYS 161 in the same column as other cysteines in the alignment.

After inspecting your MSA you can click on the right side of the **Sequence Identity Graph:** list item. This will display a barplot of the identified sequence identity between your sequence and that of the five most homologous templates (see Figure 4) ranked by transmembrane sequence identity.

Two tables below the barplot (see Figure 5) give a detailed breakdown of the identity of the query sequence to the selected template set in total identities per topological region, that is, the 7TM's, 3ECL's, 3ICL's, the amino and carboxy terminus, and also homology percentages according to: i) the whole query sequence, ii) the TM regions, and iii) the loop regions. This allows the user to identify the regions of major influence to total sequence identity.

Without any additional information, we recommend that you use the identity score for transmembrane regions **TM % id** as a major indicator of the suitability of a template (See the table in Figure 5). Selection of the top template based on this criteria is done automatically, in this case we see that 4dkl scores as the top template. In cases of low sequence identity in general it's good to use our general knowledge of GPCR's. In this case we know that the Orexin receptor belongs to the  $\beta$ -branch of the Rhodopsin family, so, using a template from the same branch in the Rhodopsin family makes sense. In our templates we have only one member of the *beta*-branch of the Rhodopsin family, that is, the **4buo** templates, so we choose this template as the one to create our models even if it doesn't score in the

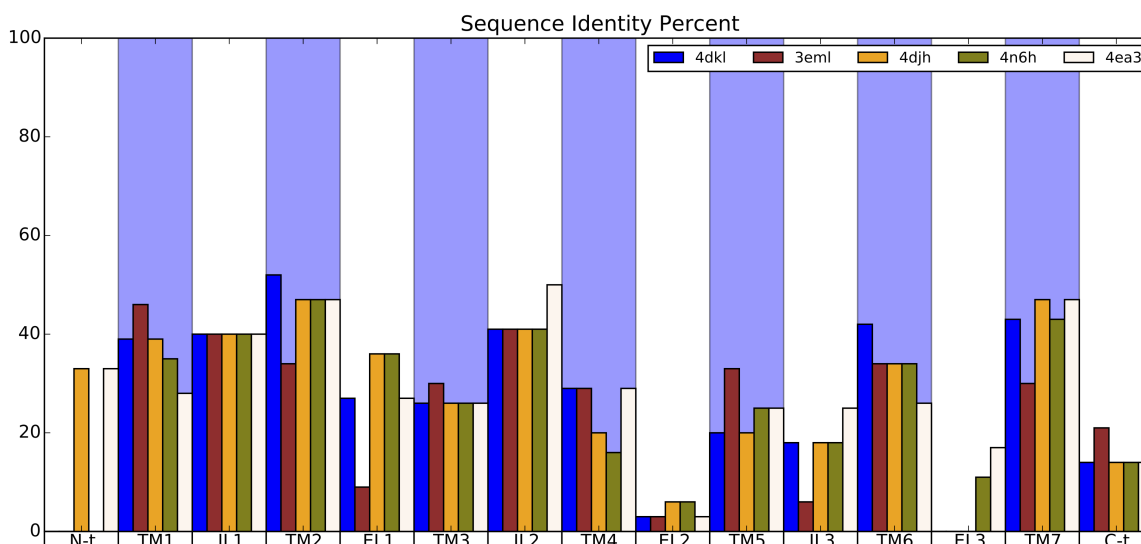


Figure 4: The barplot shows the sequence identity of the query sequence to that of the best five templates in the alignment group selected by the user, that is, the **inactive** group. Sequence identity is split into the known topological regions of GPCR's.

top five templates. We can see in the detailed per region sequence identity numbers in Figure 5 that in general all templates score poorly for this example.

After you have checked your alignment and refined it if necessary, then you can take the next step and click on **Model**. You will be shown a new window where you can select the template or templates you want to use to construct your single template or multiple template model. By default all topological regions in the top ranked model are selected (**TM1-H8**), but you can change your selection. You could for example select the tenth model if you have an experimental clue to base your selection on, or if you have a preference based on your refined sequence alignment. In our case we pick the **4buo** template. The **4buo** template corresponds to the Rat Neurotensin receptor which belongs to the  $\beta$ -branch of the Rhodopsin family. This is the only receptor in our templates which belongs to the same branch as the Orexin receptor, our target. To select it we write **TM1-H8** in the box and make sure to delete those fields from the default selection, that is, delete the **TM1-H8** from the **4dki** box.

#### 4. GENERATE HOMOLGY-BASED MODELS:

Up to 50 homology models can be generated on the basis of the pairwise alignment with the selected template or templates. This step makes use of the Modeller [2] software. You can select the number of desired models (we recommend 10) and input any additional disulfide bridges in your query sequence. In this case we are going to request a disulfide bridge between residues 78 and 161, that is, a cysteine at the beginning of TM3 and a cysteine near the end of ECL2. Click on **Submit** and wait for a few minutes.

- Tip 1: Note that the conserved disulfide bridge between TM3 and ECL2 should be maintained so the two cystein residues involved must be well aligned with the template. In this example the alignment between the cysteine in the ECL2 which forms the disulfide bridge and all other cysteines in the MSA is missed. So we modify our alignment by deleting 17 gaps before the sequence ECSTVFPGLANKTTLFTVC~~DERWG~~ and insert seventeen gaps after to keep the rest of the alignment unmodified. We also notice that there are five gaps after this conserved cysteine in the MSA, so we also introduce five gaps after CYS 161 and delete five gaps (slashes or minus signs in the alignment) after the DERWG sequence (see Figure 3). If any other disulfide bridges occur on the template and there is homology in the alignment with the corresponding cysteine residues, the bridge will also be modeled without any further information from the user.



	N-t	TM1	IL1	TM2	EL1	TM3	IL2	TM4	EL2	TM5	IL3	TM6	EL3	TM7	C-t
1u19 - opsd bovin	0	4	3	7	2	5	2	3	2	7	0	7	1	5	3
2rh1 - adrb2 human	0	6	5	7	3	9	5	7	2	7	3	7	1	8	4
2vt4 - adrb1 melga	0	8	3	7	2	11	5	5	2	5	2	7	3	13	5
2z73 - opsd todpa	1	5	3	6	3	8	2	6	1	5	0	5	0	5	5
3eml - aa2ar human	0	13	4	8	1	9	5	7	1	8	1	9	0	7	3
3odu - cxcr4 human	0	8	2	10	3	12	3	4	3	4	4	6	3	6	0
3oe0 - cxcr4 human	0	8	1	10	3	12	3	4	3	4	2	6	3	6	0
3pbl - drd3 human	0	8	4	8	3	8	2	3	1	7	3	9	0	10	5
3rze - hrh1 human	0	6	3	8	2	6	3	5	1	5	2	6	0	10	3
3uon - acm2 human	1	7	5	5	2	9	3	7	1	4	2	6	0	9	3
3v2w - slpr1 human	0	4	1	9	0	8	0	6	0	3	1	9	0	8	2
3vw7 - par1 human	0	8	0	7	3	7	3	3	3	4	1	6	3	3	1
4buo - ntr1 rat	0	7	0	7	4	3	6	7	3	2	4	6	0	9	4
4daj - acm3 rat	0	8	4	5	3	8	4	7	1	4	0	8	0	8	4
4djh - oprk human	1	11	4	11	4	8	5	5	2	5	3	9	0	11	2
4dkl - oprm mouse	0	11	4	12	3	8	5	7	1	5	3	11	0	10	2
4ea3 - oprx human	1	8	4	11	3	8	6	7	1	6	4	7	3	11	2
4mbs - ccr5 human	0	8	1	5	3	4	3	6	2	6	2	5	0	7	3
4n6h - oprd human	0	10	4	11	4	8	5	4	2	6	3	9	2	10	2
4ntj - p2y12 human	1	7	0	6	1	7	1	4	0	3	1	5	0	9	2
4yay - agtr1 human	0	9	2	10	4	9	5	6	3	3	1	6	2	7	3
4z35 - lpar1 human	0	7	1	6	0	9	0	6	0	2	0	8	1	10	1

	Total id	Total % id	TM % id	Loop % id
4dkl - oprm mouse	82	27.99	35.96	16.33
3eml - aa2ar human	76	25.94	34.27	12.24
4djh - oprk human	81	27.65	33.71	18.37
4n6h - oprd human	80	27.30	32.58	20.41
4ea3 - oprx human	82	27.99	32.58	21.43
2vt4 - adrb1 melga	78	26.62	31.46	17.35
3pbl - drd3 human	71	24.23	29.78	13.27
2rh1 - adrb2 human	74	25.26	28.65	19.39
3oe0 - cxcr4 human	65	22.18	28.09	15.31
3odu - cxcr4 human	68	23.21	28.09	18.37
4yay - agtr1 human	70	23.89	28.09	17.35
4daj - acm3 rat	64	21.84	26.97	12.24
4z35 - lpar1 human	51	17.41	26.97	2.04
3v2w - slpr1 human	51	17.41	26.40	2.04
3uon - acm2 human	64	21.84	26.40	13.27
3rze - hrh1 human	60	20.48	25.84	11.22
4buo - ntr1 rat	62	21.16	23.03	17.35
4ntj - p2y12 human	47	16.04	23.03	3.06
4mbs - ccr5 human	55	18.77	23.03	11.22
2z73 - opsd todpa	55	18.77	22.47	9.18
3vw7 - par1 human	52	17.75	21.35	13.27
1u19 - opsd bovin	51	17.41	21.35	10.20

Figure 5: The tables included with the barplot of sequence identity percent show the total number of sequence matches between the query sequence and the available templates split by topological region. They also show the total number of sequence matches and the corresponding percentages of sequence id. for the whole sequence, the TM regions, and the loop regions. Highlighted in a red box is the **4buo** template which we select in this tutorial.

- Tip 2: Now you are ready to analyze your models. They appear sorted by their most favorable value of the scoring function (DOPE-HR). In addition, a Ramachandran plot is provided as a PDF file, so you can check the stereochemistry quality (click on **Ramachandran test in pdf** on the corresponding model). There is no magical rule of thumb to select the most appropriate model, but a combination of these two values is usually a good idea. In this case, the best model according to the DOPE-HR has also good stereochemical quality, so we will select this one for the next step (click on the box with the highest value after the name models\_-).
- Tip 3: Visual inspection is an important check in this step. A JSmol applet is provided to visualize each model. Just click on **View** for the corresponding model and a new window will pop-up and display the model in your web browser. Figure 7 shows what this looks like. In addition, if you want to download and compare all your models, a tar file is provided, on top of the list of the models (**Prepare a .tar with this**). It includes a Pymol script (align.pml) which allows you to superimpose all models in a Pymol session. This is useful for example to check the orientation of selected sidechains, such as those in the binding site within the different models.

## 5. REFINE LOOPS (OPTIONAL):

Performing loop refinement is only advised for loops with up to 12 residues. We include an option to do this using the LoopModel routine as implemented in Modeller. Importantly, this refinement starts from

**Number of Models:**

**Apply Lennard-Jones restraints:** ☐ [note](#)

**Select top templates:**

3rze	TM1, EL3 etc.
4mbs	TM1, EL3 etc.
4ntj	TM1, EL3 etc.
4buo	TM1-H8
2z73	TM1, EL3 etc.

**S-positions in sequence:** 27, 58, 78, 98, 108, 132, 144, 161, 177, 188, 203, 228, 293

**add S-bridges:**

**S-bridge:**

Figure 6: The top template is selected by default but you can choose another template or a combination of regions from various templates. You can also select the number of homology models you want to produce. In this example we select the **4buo** templates since it's the only one which belongs to the same  $\beta$ -branch the Orexin receptor belongs to.

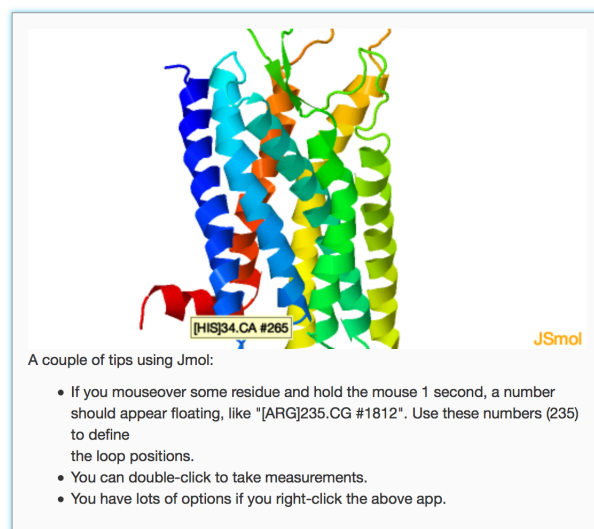


Figure 7: The JSmol plugin allows visualization of homology-modeled receptors. Notice how by placing the mouse pointer in top of a residue it will pop-up a tag with the corresponding residue identity.

the initial configuration of the sequence modeled in the previous step (i.e., by homology to the considered template or templates). Select the loops that you want to refine by activating the corresponding check boxes (See Figure 8). **PLEASE NOTE** that the initial guess of the start and end of loop regions based on the structural alignment of the templates is poor. You will need to redefine these regions, and doing so with the aid of the JSmol applet is a good idea. Just place the mouse cursor on top of the starting residue for the loop that you want to model, wait for a second or two, and the corresponding residue number will appear. You can repeat this procedure to get the remaining residue numbers in the loop or loops you want to model. You can alternatively download the pdb file and get the residue numbers for the loop region to be modeled using Pymol. Once you have have defined the residue numbers comprising the loops that you want to model click on the **Submit Query** button and proceed.

- Tip 1: The speed of this process is directly proportional to the number of loops and their length. Take this into account and select only the regions needing further refinement. The slower pace of loop modeling is the reason why only 5 parallel runs are allowed.
- Tip 2: It is generally possible to refine the 1st and 3rd extracellular loops as they are shorter than 12 residues. The 2nd extracellular loop is usually too long to be considered in this step, but if we choose only part of it after the conserved cysteine, then we can also “partially ” refine ECL2. We are going to select the region starting three residues after the conserved Cys 161, that is, the region starting with Arg 164 and ending in Tyr 170, for a total of seven residues.
- Tip 3: If you do not want to refine any loop, just select your model and proceed to click in **Run Dynamic**. You will select the structure (i.e. without loop refinement) to perform an MD simulation.

View the PDB structure ×

In the following boxes please provide the beginning and ending residue numbers of the loops you want to model. To find these numbers take a look at your pdb model using JSmol in the previous window. You can also download your model and open it with a molecular visualization program such as PyMOL to get the correct residue numbers.

Number of Models		5	
ICL1 From		to	
ECL1 From	68	to	75
ICL2 From		to	
ECL2 From	164	to	170
ICL3 From		to	
ECL3 From	243	to	251

Remember to tick the loops you want to refine!

☐ ICL1  
☒ ECL1  
☐ ICL2  
☒ ECL2  
☐ ICL3  
☒ ECL3

Figure 8: Selecting loops for loop-modeling. We model the three extracellular loops. For the second extracellular loop we choose to model only a part after the conserved cysteine 161 residue. Notice also that we are selecting to produce 5 models.

In Figure 8 the selection of the three extracellular loops is shown. You can use these definitions for this exercise. The selection of the best model, if you have refined the loops, can be done on the basis of the same criteria explained before for the homology-based models (DOPE-HR, stereochemistry quality, mutagenesis data). If you did choose to refine the loops then you can select the best model according



to the DOPE-HR score. If, alternatively, you have not refined any loop, you will proceed to the next (**Run MD**) step in the same way as before.

## 6. MEMBRANE INSERTION AND MD SIMULATIONS:

Once you have selected your loop-refined model, you can proceed to equilibrate the structure in a phospholipid bilayer. This procedure is quite simple for the user: the only thing you have to do is to click in the **Run MD** button and confirm the submission by clicking **Submit Query**. Then you have to wait ... for about 24h. A new window telling you the status of the process will appear. You can click on the process number and it will show you a progress bar telling you which part of the dynamic run is being computed in the cluster.

When the job is finished or if there are any problems with the submission you will be e-mailed about the status of the process and a link in the e-mail message will point to the status page of the molecular dynamics (MD) run, something like:

You can check this dynamic at <http://open.gpcr-modsim.org/dynamic/350>

If the job has been carried through to completion you will find a **Download results** link where you can click to download a compressed file slightly smaller than 30 Megabytes called MD\_output.tgz. This compressed file contains the results of the molecular dynamics (MD) run performed in GROMACS. The procedure that was performed at this stage is shown in a diagram in Figure 9.

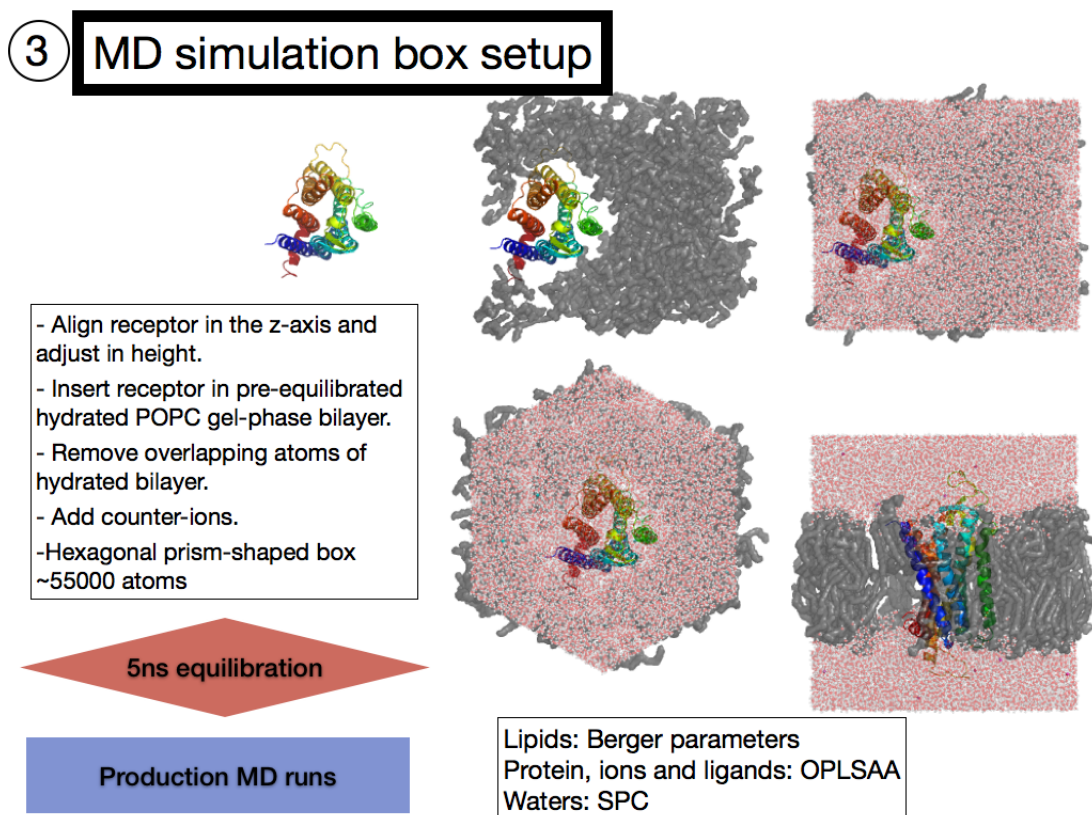


Figure 9: Steps performed for membrane insertion, water-box solvation and molecular dynamics modeling with GROMACS.

In the MD protocol the receptor is aligned with the TM regions parallel to the Z-axis, and then inserted in a pre-equilibrated POPC (Palmitoyl-Oleoyl-Phosphatidyl-Choline) membrane. The OPLS forcefield is selected for the protein atoms, while the Berger parameters are used to represent the membrane. The Berger parameters have been adapted in order to make them compatible with the

OPLS forcefield, and the solvent molecules are represented with the SPC water model. An hexagonal prism-shaped box is built, and simulated under Periodic Boundary Conditions (PBC). More information on the full protocol of the molecular dynamics run performed at the open.gpcr-modsim.org server can be found in our publication <http://www.ncbi.nlm.nih.gov/pubmed/21480628>. The equilibration simulates 5 ns of molecular dynamics (this takes around 24h in the computational biology cluster which is performing the calculations at Uppsala University) and includes the following stages:

STAGE	RESTRAINED ATOMS	FORCE CONSTANT <i>kJ/(mol · nm<sup>2</sup>)</i>	TIME ns
Minimization	-	-	(Max. 500 steps)
Eq1	Protein Heavy Atoms	1000	0.5
Eq2	Protein Heavy Atoms	800	0.5
Eq3	Protein Heavy Atoms	600	0.5
Eq4	Protein Heavy Atoms	400	0.5
Eq5	Protein Heavy Atoms	200	0.5
Eq6	Ballesteros-Weinstein Pair Atoms	200	2.5

Once the MD simulation is complete you will see an additional window inside of the models window and in the list item called **Task status**: details of when the jobs started and finished and whether it was succesful or not. Hopefully you will get a message telling you “status Finished”, meaning that the molecular dynamics run was succesfully completed.

If you want to see all of your results simply click on the **Projects** tab in the webpage header and all your projects will be displayed. Navigation through the projects is easy, just click on the name that defines your job. In our case, this was **orexin2model1**. Click there, and all stages will appear. Click on the **associated Dynamics** link and you will see a box which on click will guide you to your dynamic run or runs you’ve submitted. The **Download results** link points to a compressed file (extension .tgz) comprised of all dynamics results and some automated analysis performed with our customized PyMemDyn molecular dynamics recipe.

So, what does that compressed file contain? After decompressing the file you will have a folder with several files related to your simulation:

#### INPUT:

- popc.itp # Topology of the lipids
- ffoplsaa\_mod.itp # Modified OPLSAA-FF, to account for lipid modifications
- ffoplsaabon\_mod.itp # Modified OPLSAA-FF(bonded), to account for lipid modifications
- ffoplsaanb\_mod.itp # Modified OPLSAA-FF(non-bonded), to account for lipid modifications
- topol.tpr # Input for the first equilibration stage
- topol.top # Topology of the system
- protein.itp # Topology of the protein
- index.ndx # Index file with appropriate groups for GROMACS
- prod\_example.mdp # Example of a parameter file to configure a production phase (see TIPS)

#### STRUCTURES:

- hexagon.pdb # Initial structure of the system, with the receptor centered in the box
- confout.gro # Final structure of the system (see TIPS)

#### TRAJECTORY FILES:

#### REPORTS:

- traj\_EQ.xtc      # Trajectory of the whole system in .xtc format: 1 snapshot/50 ps
- ener\_EQ.edr      # Energy file of the trajectory
- tot\_ener.xvg, tot\_ener.log      # System total energy plot and log
- temp.xvg, temp.log      # System temperature plot and log
- pressure.xvg, pressure.log      # System pressure plot and log
- volume.xvg, volume.log      # System pressure plot and log

## LOGS:

In the **logs** subfolder, you will find the log files of mdrun:

- md\_eq\_force\_constant.log      # log of stages with restrained heavy atoms of the receptor
- md\_eqBW.log      # log of the stage with distance - restraints on the receptor

## \* TIPS \*

- If you want to configure a .tpr input file for production phase, you can use the template “prod.mdp” file by changing the number of steps (nsteps), and thus the simulation time you want to run. After that, you just have to type:

```
grompp -f prod.mdp -c confout.gro -p topol.top -n index.ndx -o topol_prod.tpr
```

- If you want to create a PDB file of your system after equilibration, with the receptor centered in the box, type:

```
echo 1 0 | trjconv -pbc mol -center -ur compact -f confout.gro -o confout.pdb
```

## 7. RESULTS:

The present example is a real test of the performance of the current protocol. This example was carried out before the Orexin receptor structure was included into our set of inactive templates. Thus comparing the result with the real structure one finds that the method performs well given the difficulty of the problem where there is little transmembrane and loop identity with the set of available templates.

Model	RMSD C $\alpha$ trace
Homology model (to 4s0v.pdb)	2.24

## REFERENCES

- [1] Jie Yin, Juan Carlos Mobarec, Peter Kolb and Daniel M. Rosenbaum 2015, Crystal structure of the human OX<sub>2</sub> orexin receptor bound to the insomnia drug suvorexant. *Nature*, **519**, 247-250.
- [2] Benjamin Webb and Andrej Sali, Comparative protein structure modeling using MODELLER. *Current Protocols in Bioinformatics*, **47**, 1-32.
- [3] Sander Pronk, Szilard Pall, Roland Schulz, Per Larsson, Par Bjelkmar, Rossen Apostolov, Michael R. Shirts, Jeremy C. Smith, Peter M. Kasson, David van der Spoel, Berk Hess and Erik Lindahl, GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit. *Bioinformatics*, **29**, 845-854.
- [4] David Rodríguez, Angel Piñeiro, and H. Gutiérrez-de-Terán. 2011, Molecular dynamics simulations reveal insights into key structural elements of adenosine receptors. *Biochemistry*, **50**, 4194-4208.