WIP requirements and how-to for making proper membrane protein structure files

See <https://opm.phar.umich.edu> for a great database of protein structures that include planar membrane annotations within the files.

Requirements for membrane proteins

To properly place a membrane protein in/on a membrane, CTS has two requirements: the TMD must have a known centroid coordinate (either an annotated submodel or fixed at 0,0,0) and the structure must be oriented with the external domain in the positive Z direction (towards the viewer in most software, including ChimeraX and PyMol). OPM database structures conform to this standard (the TMD centroid may be off 0,0,0 by a negligible margin of 0-2 angstroms) but the annotation must be discarded or marked for handling by CTS.

DIY with ChimeraX (using OPM structure)

OPM .pdbs are already aligned and centered, the annotation only needs to be deleted or named for handling by CTS.

1: delete the annotation

Select the annotation ‘residues’ of the dummy atoms:

select :DUM

Delete the selected atoms:

delete sel

The structure can be saved, as it is already aligned and centered as needed. You must NOT save relative to any model, as that will shift the coordinates written to the output file. You can split the remaining model into submodels for use as a complex with the split command as below.

2. name the annotation

Select the annotation as above, but instead of deleting the atoms instead split them into a submodel:

split #1 atoms sel

rename the new submodel (might be different from 1.2) something that includes ‘origin’

rename #1.2 origin\_TMD

This structure must be saved as a .cif file because .pdb do not save submodel names. The ‘origin’ submodel will be used to define the center of the whole model, then be deleted. It is suggested to not save it relative to a submodel to not shift coordinates for similarity to the original. This will also improve performance if there is no membrane/if the model is renamed to be standard non-membrane.

DIY without OPM structure

Select/split the TMD thing

Define centroid the model/selection

Move the model such that the centroid becomes 0,0,0

First, a model must be split with the split command – example to select OPM plane annotations:

split #1 atoms :DUM

For a new structure, replace :DUM with ‘sel’ after selecting manually or with the select command the atoms that will define your centroid. The split command can also split by chains or any other arbitrary selection of atoms/residues for generating a multi-model structure that CTS can use as a complex class. For quick and dirty manual models, you can use ctrl-shift to box select a rectangular TMD.

Alignment

Centering

Option 1: OPM annotated .cif file

OPM .pdb structures are (\*as far as I can find) already oriented along Z and have the TMD centered at or near 0,0,0.

Model generation now supports placement of membrane proteins in or on generated vesicles through the classes 'memplex' (functioning as a complex) and 'membrane'. Membrane proteins must be oriented with the external domain in the positive Z direction, and must be oriented around 0,0,0 as the transmembrane centroid.

To conform to these requirements using ChimeraX, you can use define to create a plane or axis from the TMD, and then align to orient it along the Z axis, using turn y 108 if it is reversed. If this does not center the TMD at 0,0,0 you can use the measure command to determine the offset and the move command to shift all models until the TMD centroid is at 0,0,0. Do not save relative to any model, that will shift coordinates.

If you use an OPM structure or otherwise create a model annotation for the TMD, you can instead rename that TMD-defining model (such as the OPM planes) to something that includes 'origin' and save it as a .cif (as .pdb do not retain model names). CTS will detect the model name and use it to define the centroid. OPM models seem to already be aligned to this standard (and with the TMD centroid very close to 0,0,0) and so should be easier.

Note that old chimera cannot save .cif files, and chimeraX does not load model names - it can only save them, so it will overwrite any model names if used to open and save the file again.