**Protocol Capture for RosettaGPCRPocketSize**

All files and examples are given at: <https://github.com/FabianLiessmann/RosettaGPCRPocketSize>

Rosetta Documentation main site: <https://www.rosettacommons.org/docs/latest/Home>

* RosettaCM Documentation
* [Hybridize mover documentation](https://www.rosettacommons.org/docs/latest/scripting_documentation/RosettaScripts/Movers/movers_pages/HybridizeMover)
* [RosettaScripts documentation](https://www.rosettacommons.org/docs/latest/scripting_documentation/RosettaScripts/RosettaScripts)
* [Constraint documentation](https://www.rosettacommons.org/docs/latest/rosetta_basics/file_types/constraint-file)

Additional Material:

* [High-Resolution Comparative Modeling with RosettaCM](https://www.sciencedirect.com/science/article/pii/S0969212613002979) (doi 10.1016/j.str.2013.08.005)
  + Original paper describing method
* [Protocols for Molecular Modeling with Rosetta3 and RosettaScripts](https://pubs.acs.org/doi/10.1021/acs.biochem.6b00444)
  + Tutorial paper giving overview of Rosetta, primary methods, and associated tutorials
* [Original RosettaGPCR paper](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7652349/) Improving homology modeling from low-sequence identity templates in Rosetta: A case study in GPCRs (doi 10.1371/journal.pcbi.1007597)
  + Paper describing the method and input, preparing the templates
  + The full preparation is available at [www.rosettagpcr.org](http://www.rosettagpcr.org/) and [www.github.com/benderb1/rosettagpcr](http://www.github.com/benderb1/rosettagpcr).
* [MeilerLab Tutorials (comparative modeling and more)](http://meilerlab.org/index.php/rosetta-tutorials)
* [Constraint Tutorial](https://new.rosettacommons.org/demos/latest/tutorials/Constraints_Tutorial/Constraints)

Notes: Paths are relative, Rosetta: all source code necessary to run Rosetta -- /PATH/TO/ROSETTA/. Two examples for RosettaGPCRPocketSize are provided (ADRB1\_example and CXCR4\_example). Four directories for a step-wise approach are given, this is not necessary but can facilitate the execution of RosettaGPCR-PocketSize and getting used to the pipeline. In the end, for this method you generate a Ballosteros-Weinstein numbering (BW) file, which correlates your residue numbers to the BW numbering, build receptor individual randomized constraints, execute RosettaGPCR with RosettaCM.xml scripts as well as following evaluation script.

All scripts are written as jupyter notebook (\*ipynb). If Anaconda3 is installed, it should be in the base environment, else use conda install -c anaconda jupyter and start the script in the respective conda environment with jupyter notebook \*.ipynb. Another option is to install jupyter notebook with pip install notebook and again jupyter notebook \*.ipynb. All needed inputs are marked with several lines of ############# in the script.

Necessary inputs:

* RosettaGPCR preparation
* Fasta sequence (either from RosettaGPCR or self-selected)
* BW file e.g. gpcrdb.com (or generated in Step1 with BW\_assignment.ipynb, assignment.list and all\_gpcrdb\_num.data)
* copy\_cst\_and\_xml.ipynb for generating constraint files
* Calculate\_volume\_and\_filtering.ipynb for the volume calculation and final filtering

Step 0: Prepare the input files as described in RosettaGPCR

* Prepare templates, alignment, setup for RosettaCM (see setup\_RosettaCM\_updated.py) and span file
* Copy the target.fasta into “Step1\_Generate\_BW\_assignment”

Step 1: Generate a BW file assigning the residue numbers to BW numbering

* Select a target sequence and copy it in the “Step1\_Generate\_BW\_assignment”
* The target sequence should include a < {receptor name} line
* assignment.list includes the UniProt and IUPHAR name of all class A GPCRs, check if your receptor name is included in the list. The second column is used for assigning the receptor
* all\_gpcrdb\_num.data lists all human class A GPCRs with their BW numbering and respective residues. The script is optimized for human sequence, but will work for other organisms as well. In doubt, add your target BW from gcprdb (<https://gpcrdb.org/alignment/targetselection>)
* Run BW\_assignment.ipynb step-wise and produce your receptor BW file

Step 2: Generate several constraint file and update the RosettaCM script

* Copy your BW file and change into “Step2\_Generate\_CST”
* Check if selected\_pairs.list is in the same directory. This file includes the distance pairs and the selected distances. Six distances equal one tetrahedron
* Update your number of desired constraint files and run Make\_cst\_file.ipynb. Several hundred to a few thousand homology models are best for RosettaCM, change your constraint file number accordingly (either constraint file number ≜ desired homology model number or a multiple of the constraint file number ≜ desired homology model number, can be adjusted with the nstruct number in flags.options)
* A bonus block will prepare a copy\_cst\_and\_xml.sh script.
* Before running the bash script, check your rosetta\_cm.xml. Have you used the setup\_RosettaCM\_updated.py and does your xml script include ConstraintSetMover and FastRelax? Compare to example\_rosetta\_cm.xml

Step 3: Run RosettaCM/RosettaGPCR accordingly

* Check that all inputs are present and run homology modeling
* In the rosetta\_cm directory there should be:
  + rosetta\_cm{number\_of\_constraint\_file}.xml (Several xml files according to the number of constraint files)
  + {name\_of\_target}.cst (Several constraint files)
  + flags.options
  + span.txt
  + disulf.txt
  + threaded template pdbs
  + output/directory
* If everything is present, run Rosetta\_scripts and include a -parser:protocol rosetta\_cm{number}.xml flag

Step 4: Evaluate the pdbs, their volume and score

* Copy all pdb files and the BW file to Step4\_Filtering and run Calculate\_volume\_and\_filtering.ipynb
* For all pdb the score in the pdb is extracted and the volume calculated (first, a \*.pdb.xyz file with the cartesian coordinates is generated and afterwards, a \*.pdb.xyz.vol file with the respective volume. This number of files is for debugging if there is an error with the volume)
* The script will merge score and the volume, calculate the median volume and select the pdbs within a specific bandwidth and the best 5 models

Further notes/tips:

* Check the results of the run in PyMOL, Chimera or any other visualization tool
* Doublecheck with gpcrdb.com your BW numbering
* If your receptor has a known smaller or bigger binding pocket, change the distances in selected\_pairs.list. A overview with all inactive determined class A GPCRs and their respective pocket distances is given in S1\_all\_inactive\_distances.ods. Family related or specific receptor distances can be selected
* If you are unsatisfied with the results, change the parameter selection in Step2 (increase or decrease the weight, distance bonus and or constant function)