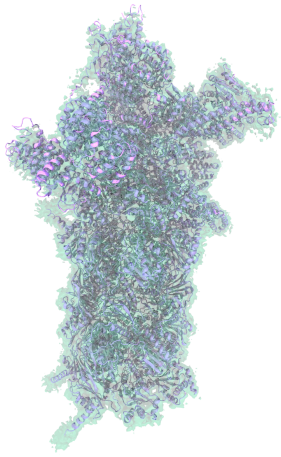
University of Illinois at Urbana-Champaign
Beckman Institute for Advanced Science and Technology
Theoretical and Computational Biophysics Group
Computational Biophysics Workshop

Interactive Model Building with ModelMaker



ModelMaker Developers:
Till Rudack
Maximillian Scheurer
Ryan McGreevy

Tutorial by Maximilian Scheurer, Till Rudack, Marc Siggel, Ryan McGreevy, João Ribeiro, and Justin R. Porter

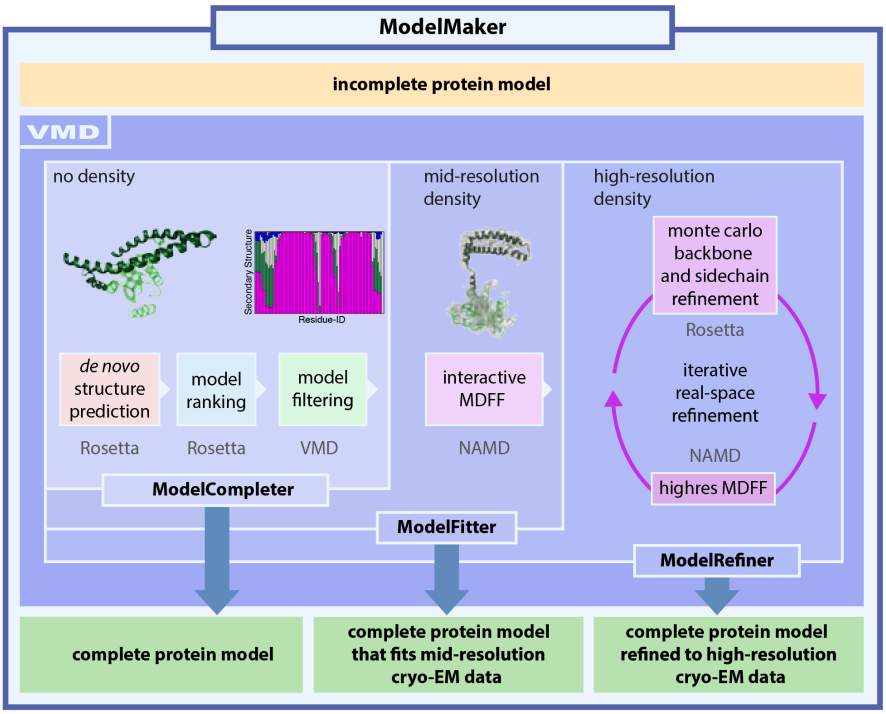
A current version of this tutorial is available at
http://www.ks.uiuc.edu/Training/Tutorials/

Contents

1 Introduction

Hybrid structure analysis strategies, which combine structural data from sources such as X-ray crystallography and cryo-electron microscopy (cryo-EM) with computational modeling, have become successful means for resolving structural models of macromolecular complexes found in living cells. The computational modeling tools employed in these strategies usually aim to automate the whole process of structure analysis in order to avoid human bias, yet the experience of structural biologists may actually be a desirable factor in structure refinement. Here, we present a tool, named ModelMaker, to interactively build complete models guided by incomplete structural data from experiments, automated structure prediction, and user expertise. With ModelMaker, incomplete models are completed by generating ensembles of models of the missing segments with *de novo* structure prediction in Rosetta. Then, a single complete model is obtained by ensemble filtering through sorting, clustering, and secondary structure analysis. This model is further refined in real space to fit mid- or high-resolution cryo-EM densities through a combination of molecular dynamics flexible fitting (MDFF) with monte carlo based backbone and sidechain rotamer search algorithms in an iterative manner. ModelMaker is found to be of particular value for modeling the missing highly flexible or multi-conformational domains of large macromolecular complexes with sparse density as well as refining models to high-resolution densities. Furthermore, ModelMaker can be employed to complete missing segments of structures without any density information to obtain complete structures to initiate molecular dynamics simulations. ModelMaker is a tool that takes advantage of the popular and user-friedly molecular visualizaion software VMD that provides an easy-to-use environment to run the usually very complex computational modeling tools.

In this tutorial the application of ModelMaker is demonstrated by completing the C-terminus (3) and a structurally unresolved protein insertion (4) of the X-ray structure of the proteasomal subunit Rpn11 in yeast (chain B of PDB ID 4OCM) and fit it to the mid-resolution (7.7 Å) cryo-EM density of Rpn11 derived from the cryo-EM density of the yeast 26S proteasome (EMD-2594) (5). The quality of the model generated by ModelMaker is proven by comparison to the structure of Rpn11 (chain G of PDB-ID 3JCK) derived from a high-resolution cryo-EM density (3.5 Å). The generated model of yeast Rpn11 serves as basis for a homology model of human Rpn11 (6) which is then further refined through ModelMaker to fit the high-resolution cryo-EM density (3.9 Å) of the human 26S proteasome (EMD-4002) (7). In a similar manner as shown in this tutorial for the Rpn11 example, ModelMaker was employed to derive the structure of the human 26S proteasome (PDB ID 5L4G and 5L4K) from a 3.9 Åcryo-EM density of the human 26S proteasome (EMD-4002).



We recommend completing the QwikMD/MDFF, VMD, and MDFF tutorial first.

QwikMD/MDFF: `http://www.ks.uiuc.edu/~trudack/QwikMDFF/`

VMD: `http://www.ks.uiuc.edu/Training/Tutorials/vmd/tutorial-html/`

MDFF: `http://www.ks.uiuc.edu/Training/Tutorials/science/mdff/tutorial_mdff-html/`

2 Required software

2.1 ModelMaker

Download the ModelMaker folder, which contains the files for the ModelMaker plugin to VMD. As ModelMaker wraps a multitude of different programs for computational biology, there are several dependencies:

- **VMD:** Visual Molecular Dynamics; is the main program we are going to use throughout the tutorial. It will act as the central branch point between all the other tools as well as for visualization and analysis. Download link: http://www.ks.uiuc.edu/
- **NAMD:** Nanoscale Molecular Dynamics; is the calculation program for all Molecular Dynamics simulations we we will carry out.

http://www.ks.uiuc.edu/

- **Rosetta:** is the *de-novo* structure prediction and refinement tool we integrated in the ModelMaker tool. For build instructions, read the following subsection.
- **Gnuplot:** will be used as automated plotting program. Please install it with the package manager of your Linux distribution or any package manager on Mac OS X.
- **MODELLER:** will be used for building homology models in the tutorial. Download it from https://salilab.org/modeller/ and follow the given installation and license request instructions.
- **Situs:** is a program package designed for modeling atomic structures with EM densities, such as rigid body docking. See the download site http://situs.biomachina.org/ for download and build documentation.

2.2 Rosetta

Download Rosetta from https://www.rosettacommons.org. An academic license can be acquired on the homepage for free. Download the package for your Operating System (OS). The weekly release contains the up to date software and requires about 5GB of disc space.

1 Unpack the tar.zip file:

```
tar -xvf ROSETTA_filename.tar
```

2 The SCons tool is used to build Rosetta alternatively to the Make build tool. It is written in python and allows a simple multi-platform build process.

```
cd $PATH_TO_ROSETTA/main/source

./scons.py -j<number_of_processors> mode=release bin
```

The previous command builds all executables in the release and will activate optimization flags to increase the performance.

Hint: Check the OS of the workstation or cluster you want to use to run Rosetta. It is recommended to build Rosetta specifically for the target OS. If not, compatibility issues can occur during the run time. Also, make sure that build and OS versions match.

Windows Installation! Currently there is no simple way to build Rosetta on Windows. Rosetta recommends the use of a virtual machine or running linux in parallel on a local machine. For further information see the Rosetta webpage (https://www.rosettacommons.org/docs/latest/build_documentation/Build-Documentation). However, you can conduct the VMD part on a Windows machine. All necessary Rosetta outputfiles are provided with the tutorial files.

2.3 MODELLER

Go to the MODELLER website (https://salilab.org/modeller/tutorial/), download the newest version of MODELLER and request a license. Install MODELLER on your workstation.

3 Folding protein termini using ModelMaker

Here, we predict a structural model for the truncated C-terminal tail from amino acid 217 to 306 of chain B of 4OCM. Create a folder named `terminus` for this section and navigate to it. Sample input and output files are provided in the `3.terminus` folder in the tutorial folder. First you need to obtain a template structure which should be completed by modeling the structurally unresolved segments. In this tutorial we complete Rpn11, the deubiquitilyation subunit of the 26S proteasome. As template for modeling, we use chain B of PDB structure 4OCM. Go to the Protein Data Bank (PDB, http://www.rcsb.org/pdb/home/home.do) and download the PDB structure with the PDB-ID 4OCM. Alternatively you can download the PDB structure directly through VMD by going to File → New Molecule, type 4OCM in the Filename box and click the Load button. The PDB structure 4OCM contains two Rpn8/Rpn11 dimers in the unit cell. For now we only need one Rpn11, so we use chain B. In order to create a PDB file containing only chain B run the following command in the TK console (Extensions → Tk Console):

```
[atomselect top "chain B and protein"] writepdb rpn11_yeast_4ocm.pdb
```

3.1 Structure prediction

In order to use ModelMaker for structure prediction you first need to modify the configuration file.

1 **Preparing the configuration file:**
Copy the prepared configuration file `fold_rpn11_terminus.tcl` from the tutorial files to your `terminus` folder, open it in a text editor and change the following variables to fit your workstation configuration.

variable	description
`packagePath`	path of the ModelMaker plugin files
`vmdexe`	path of your vmd executable
`gnuplotexe`	path of your gnuplot executable
`rosettapath`	directory path containing the Rosetta binaries
`rosettaDBpath`	Rosetta database path
`platform`	`"linuxgccrelease"` or `"macosclangrelease"`

Table 1: Default configuration variables

2 Obtaining the target amino acid residue sequence:
As we are only going to fold the C-terminus of Rpn11, we will discard the missing N-terminus in our following procedures. Go to the *uniprot* website (http://www.uniprot.org) and download the fasta sequence of Rpn11 in yeast (UniprotID: P43588). Create a folder called `input`. With a text editor, remove the first 22 amino acids from the sequence and save the file in the `input` folder as `rpn11_yeast_23-306.fasta`. To facilitate the sub-sequence extraction, a Python script `subrange.py` is provided in the `scripts` folder, that you can use to get the amino acid codes for a given range. Simply copy the script to your working directory and make the following changes:

- **line 3:** `start` defines the start of the sequence
- **line 4:** `end` defines the end of the sequence
- **line 6:** `name` defines the input name of the given fasta sequence

Running

```
python subrange.py
```

creates a file `<name>_<start>-<end>.fasta` from the input fasta sequence.

3 Generating the fragment file library:
Use the Robetta server to generate two library files containing internal coordinates for the target sequence structure. The server performs a homolgy search algorithm in the PDB Data Bank based on a running window of 3 and 9 amino acid length and produce two files (3mer and 9mer) presenting the best 200 results for each window. To do so, go to http://www.robetta.org and set an academic user account. Submit the target sequence `rpn11_yeast_23-306.fasta` to the Fragment file server and as soon as the search is finished you will receive an email with a link to download the results. Save the 3mer and 9mer files as

rpn11_yeast_23-306.frag3 and rpn11_yeast_23-306.frag9 in your input folder.

4 **Building one complete model for the target amino acid sequence:**
Create a full_length_model folder and copy the file rpn11_yeast_4ocm.pdb to it. Furthermore, copy the file run_full_length_model.sh from the tutorial files to your full_length_model folder. Make the following changes in run_full_length_model.sh:

- **line 9:** rosetta=/path/to/rosetta/bin
  indicates the path to your Rosetta binary directory.
- **line 11:** platform="linuxgccrelease"
  supplies the platform Rosetta has been built on, thus either platform="linuxgccrelease" or platform="macosclangrelease" is accepted.

Run run_full_length_model.sh to generate a complete template model. Afterwards, rename the output file rpn11_yeast_4ocm.pdb_full_length.pdb to rpn11_yeast_23-306_complete.pdb.

```
./run_full_length_model.sh
```

```
mv rpn11_yeast_4ocm.pdb_full_length.pdb rpn11_yeast_23-306_complete.pdb
```

Rosetta's full length modell application yields PDB files that do not keep the original amino acid numbering. To keep it, copy the script renumber.tcl from the scripts folder to the full_length_model folder. In the mols list, you can specify the file name of the input PDB file, in this case, the line should look like:

```
set mols [list rpn11_yeast_23-306_complete.pdb]
```

In the next line, you can set the newstart varible to 23, so that the output PDB file starts its numbering from 23, as in the fasta file. Run

```
vmd -dispdev text -e renumber.tcl
```

to get the output file rpn11_yeast_23-306_complete-numb.pdb, then replace the old file with the new one:
mv rpn11_yeast_23-306_complete-numb.pdb rpn11_yeast_23-306_complete.pdb.

5 **Running Rosetta from VMD:**
Now that we have prepared all necessary input files, we can complete the configuration file to finally run Rosetta from VMD to predict the C-terminal structure of Rpn11. The recommendation from the literature is to predict between 5,000 and 20,000 modelss. In our test case we predict only 100 structures for demonstration purpose. We use RosettaScripts

with a Brokered Environment and execute the classic Rosetta *de novo* protocol upon it. The ModelMaker plugin can handle input file generation for Rosetta automatically, so we just need to add a few lines to `fold_rpn11_terminus.tcl`.

- **line 14:** `set nstruct 100`
  creates a variable that is later on passed to Rosetta and indicates the number of structures to generate.
- **line 15:** `set bestN 25`
  shows the number of highest scored structures that should be taken into account for analysis.
- **line 18:** `set tempPath [pwd]/full_length_model`
  points to the path containing the full length Rpn11 model, created in the step before.
- **line 20:** `set tempdir [pwd]/full_length_model`
  points to the path containing the PDB file for alignment in the analysis step. If you have an alternative PDB file you want to align the predicted models to, you can specify it by setting `tempdir`. In our test case, we simply use the full length model for alignment as well.
- **line 22:** `set comps [list [list ss 196 284 "A"]]`
  sets a list to define analysis tasks for the predicted structures. The ModelMaker packages can take a list of several analysis tasks. Here, we only define a secondary structure analysis task, that scans the amino acids 196 to 284 of chain A for the average secondary structure. The secondary structure analysis task is configured by the list in `$comps`, where the single elements stand for: `[list ss` ⟨`start residue ID`⟩ ⟨`end residue ID`⟩ ⟨`chain`⟩`]`
- **line 25:** `start_rosetta_abinitio ...`
  starts a Rosetta structure prediction task with the given arguments in Tab. 2.
- **line 27:** `analyze_abinitio ...`
  starts the analysis procedure. The argument configuration is explained in Tab. 3. This command automatically calls Rosetta energy scoring, aligns the best N (`$bestN`) `structures and performs the analysis tasks defined in $comps.`

Execute the configuration file in VMD text mode and wait for the tasks to finish. Depending on the number of structures to generate, this may take a while.

```
vmd -dispdev text -e fold_rpn11_terminus.tcl
```

If no error occurs, go to the folder called `rosetta_output_rpn11_terminus` containing the results of your run.

arg.	description	tutorial example
1	task name	`rpn11_terminus`
2	full length name	`rpn11_yeast_23-306_complete`
3	list of VMD atomselection texts with the selections to fold	`[list "resid 196 to 284"]`
4	anchor residue for coordinate constraints (select a residue ID that is **not** contained in the predicted residue range)	`1`
5	list of fragment files (multiple lists can be provided to fold multiple chains at once)	`[list [list "rpn11_yeast_23-306_frag9" "rpn11_yeast_23-306_frag3"]]`
6	fragment file and fasta path	`[pwd]/input`
7	number of structures to create	`$nstruct`
8	run on cluster flag, 0 or 1 (**not supported in tutorial version!)**	`0`
9	tasks per job on cluster (**not supported in tutorial version!)**	`0`
10	test run flag, 0 or 1 (**only set to 1 for test cases!)**	`0`

Table 2: Rosetta *ab initio* procedure arguments

arg.	description	tutorial example
1	task name of `start_rosetta_abinitio run to analyze`	`rpn11_terminus`
2	full length name	`rpn11_yeast_23-306_complete`
3	best N structures concerning Rosetta energy score to analyze	`$bestN`
4	number of created structures	`$nstruct`
5	cluster flag to analyze a cluster run, 0 or 1 (**not supported in tutorial version!**)	0
6	selection text in the template PDB file for alignment	`"resid 100 to 195"`
7	selection text in the generated models for alignment	`"resid 100 to 195"`
8	list of analysis tasks (previously defined)	`$comps`

Table 3: Rosetta ab initio analysis procedure arguments

3.2 Interactive fitting to a cryo-EM density with iMDFF and QwikMD

Create a new folder named `mdff` in your working directory.

1 Aligning the predicted model with the cryo-EM density map:
In order to perform interartive molecular dynamics flexible fitting you first need to place the modeled structure in the right position inside the density map. Download the cryo-EM density map of the 26S proteasome (EMDB-ID 2594) from the electron microscopy database (http://www.ebi.ac.uk/pdbe/emdb/): `emd_2594.map`. In this special case there exist already a near-atomic structural model (PDB-ID 4CR2) for this map. Download the structure with the PDB-ID 4CR2 from the PDB (http://www.rcsb.org/pdb/home/home.do). Use `align_segments.tcl` to align the output structure from the structure prediction (`ss_average_100.pdb`) to chain V of `4CR2.pdb` in order to get `ss_average_100_aligned.pdb`.

2 Generating a density map file for MDFF:
In order to generate a readable density map file for MDFF the file `emd_2594.map` first needs to be renamed to `emd_2594.ccp4` and then run the command in your terminal (The script is contained in the `scripts` folder):

```
vmd -dispdev text -e get_density.tcl
```

which will excecute

```
mdff griddx -i emdb_2594.ccp4 -o emdb_2594_potential.dx
mdff griddx -i emdb_2594_potential.dx -o emdb_2594_density.dx
```

to obtain the density file `emdb_2594_density.dx`, which can be read by MDFF.

3 Crop the density:
In order to crop the density to the area of interest, which is here around the predicted structural model of Rpn11, run `crop_density.tcl`.

```
vmd -dispdev text -e crop_density.tcl
```

The script generates the density file `rpn11_model_5_2594_density.dx`, which contains the density of `emd_2594_density.dx` within a cutoff of 5 $\mathring{A}$ `around` .

4 Fitting the modeled part to the cryo EM density:
We are going to use the VMD plugin QwikMD to setup the interactive MDFF run as it automatically generates all the required input files and structures.

Structure preparation with QwikMD

- Run VMD and open the QwikMD plugin (Extensions →Simulation →QwikMD).
- Browse to the average secondary structure PDB file `ss_average_25.pdb` located at `./rosetta_output_rpn11_terminus/analysis/ss_196_284/` and load it into QwikMD.
- Click on Structure Manipulation and ignore occurring errors.
- On top, navigate to the Advanced Run tab and select the MDFF tab.
- In the Protocol dropdown menu, adjust Fixed to `"resid 1 to 195"` and select `"same fragment as protein"` for Sec. Structure, Chirality and Cispeptide restraints.
- Click on Prepare and give your QwikMD file the name `rpn11_terminus_mdff` when prompted. QwikMD automatically generates the necessary PSF file and restraint files and redirects you to the MDFF graphical user interface.
- Open the MDFF files dropdown and add the cropped density `rpn11_model_5_2594_density.dx`
- To improve performance, you can adjust the number of CPU cores NAMD should use for the MDFF run by changing the value for Processors in the IMD parameters dropdown.

Run MDFF

- Click on the IMDFF Connect tab on top of the MDFF GUI and open the Cross Correlation Analysis dropdown.
- Check Calculate real-time Cross Correlation, set Experimental Density (Mol ID) to the Mol ID of the loaded density (you can obtain the ID from the first column of the VMD Main menu) and set the Map Resolution to 7.7.
- Click on Submit and Connect to start the simulation and interact with it through the VMD window.

Hint: If you have problems moving the structure to the correct positions, load the final Rpn11 structure `rpn11_yeast.pdb` from the tutorial folder into VMD while performing iMDFF.

Short introduction to iMDFF: In the first step drag the predicted structure to the density apply forces (Mouse → Forces → Atom) by clicking on an atom. In this step a grid spacing of 0.3 should be applied. For detailed instructions on the usage of the MDFF GUI and interactive fitting see the MDFF tutorial and the Youtube tutorial https://www.youtube.com/watch?v=-KJiH_WF65s. As soon as the predicted segment fits the density a second MDFF run with a grid spacing of 0.6 can be performed.
Hint: Use cartoon representation for the protein with different coloring for the fixed and flexible segments. Represent the density as solid surface with white color and transparent material. Use the CPK representation for the backbone atoms of the flexible segment and apply only forces to these backbone atoms.

4 Modeling amino acid insertions

Here, we predict a structural model for the structurally unresolved amino acid residues 162 to 177 of chain B of 4OCM. Create the folder `domain_insertion` for the files generated in this section.

4.1 Structure prediction with Rosetta

For the domain insertion folding, you can take the correctly numbered full length model that was generated in section 3.1 and continue. Again, create a new folder named `insertion`, create a folder named `full_length_model` in it and copy the already generated full length PDB to it.

1 Prediction of an ensemble of possible structures:
The recommendation from the literature is to predict between 5,000 and 20,000 structures. In our test case we predict only 100 structures for demonstration purpose. We use the topology Broker as direct input to the classic Rosetta *de novo* protocol, wrapped by the ModelMaker package, to predict a possible ensemble of structural models. As in section 3.1, we create a configuration file `fold_insertion_rpn11.tcl`. Copy your configuration file with the already adapted variables to the working directory. To perform a *de-novo* structure prediction of the missing domain, we only need to change the analysis tasks and the commands to run Rosetta:

- **line 22:** `set comps [list [list ss 138 157 "A"] [list cluster 138 157 "A" 2]]`
  In this case, we want to analyze the outcoming secondary structure from residue 138 to 157 and cluster the best 25 structures with the Rosetta cluster application. The cluster analysis task is configured by the second list in `$comps`, where the single elements stand for: `[list cluster` ⟨`start residue ID`⟩ ⟨`end residue ID`⟩ ⟨`chain`⟩ ⟨`max. cluster number`⟩`]`
- **line 24:** `start_rosetta_insertion ...`
  As in section 3.1, we call a ModelMaker function to run Rosetta with the given arguments (Tab. 4).
- **line 26:** `analyze_abinitio ...`
  The analysis arguments are to be obtained from Tab. 3. In this case, Rosetta renames the outcoming structure files, so that we need to append the original name of the template PDB file at the end and append `_S` to the full length name. The whole line should now be:

```
analyze_abinitio "rpn11_insertion" "rpn11_yeast_23-306_complete_S"
$bestN $nstruct 0 "resid 1 to 137 or resid 158 to 284"
"resid 1 to 137 or resid 158 to 284" $comps "rpn11_yeast_23-306_complete"
```

Execute the configuration file in VMD text mode and wait for the tasks to finish. Depending on the number of structures to generate, this may take a while.

```
vmd -dispdev text -e fold_rpn11_insertion.tcl
```

If no error occurs, go to the folder called `rosetta_output_rpn11_insertion` containing the results of your run.

arg.	description	tutorial example
1	task name	`rpn11_terminus`
2	full length name	`rpn11_yeast_23-306_complete`
3	list of VMD atomselection texts with the selections to fold	`[list "resid 138 to 157"]`
4	list of fragment files	`[list "rpn11_yeast_23-306_frag9" "rpn11_yeast_23-306_frag3"]`
5	fragment file and fasta path	`[pwd]/input`
6	fasta name	`rpn11_yeast_23-306`
7	number of structures to create	`$nstruct`

Table 4: Rosetta domain insertion procedure arguments

4.2 Interactive fitting to a cryo-EM density with iMDFF

Follow the steps explained in section 3.2.

1 **Aligning the predicted model with the cryo-EM density map:** See step 1 in in section 3.2.

2 **Generating a density map file for MDFF:** See step 2 in in section 3.2.

3 **Crop the density map:** See step 3 in in section 3.2.

4 **Fitting the modeled part to the cryo EM density:** See step 4 in in section 3.2.

5 Model fit to mid-resolution EM density

To refine the completed model, we are going to perform an interactive MDFF simulation of the overall protein structure. Afterwards, we will analyze the outcoming structure for quality of fit to the EM density and refine it if possible.

5.1 MDFF run

For the iMDFF run, we will use the QwikMD interface as before.

1 **Merging the final structures for input:**
Open the final structures from the MDFF runs in section 3.1 and 4 in a text editor and merge the refined segments of the protein into a single PDB file.

2 **Performing an interactive MDFF run:**
Follow the steps explained in section 3.2.

5.2 Cross correlation coloring

Now, we are going to take advantage of the implemented analysis tool for Cross correlations (CC): The quality of fit of an atomic model to an EM density is evaluated by calculating the "overlap" with the density. The higher the cross correlation value is, the better the quality of fit is. Thus, we will check the quality of fit of our models by cross correlation coloring. The result will be visualized with VMD.

1 **Preparing the configuration file** Create a folder called `cccolor` in your workspace and copy the last frame PDB of the previous MDFF run to it as well as the density file `rpn11_model_5_2594_density.dx` that the MDFF run was performed with. Additionally, create a folder called `tmp` in the `cccolor` folder. Copy the script `color_rpn11_cc.tcl` to this folder and make the following changes in the script:

- **line 2:** `set packagePath ...`
  defines the path to the ModelMaker package files.
- **line 6:** `cccolor ...`
  executes the CC coloring. The options are explained in Tab. 5. In this case, we will only color the secondary structure elements.
- **line 7:** `cccolor ...`
  In this line, we execute the CC coloring for the residues' backbone atoms. Therefore, we also have to switch the `residue CC` argument to `1` as well as the `backbone analysis only flag`.

The following script lines will load the colored structures and set the coloring mode to Beta, as the CC values have been stored in the beta columns of the produced PDB files. Thus, execute the `color_rpn11_cc.tcl` in

VMD:

```
vmd -e color_rpn11_cc.tcl
```

and check the structures in VMD for red sections, which indicate a low CC value.

arg.	description	tutorial example
1	PDB name	`rpn11_yeast_fit_lastframe-numb`
2	map name (.dx file format)	`rpn11_model_5_2594_density`
3	map resolution	`7.7`
4	threshold; see `mdff ccc` command documentation	`1.0`
5	spacing; see `mdff ccc` command documentation	1.0
6	cutoff value for CC calculation	`2.0`
7	secondary structure CC; 0 or 1; turns analysis of CC of every secondary structure element on or off	-
8	residue CC; 0 or 1; turns analysis of single residue CC on or off	-
9	general selection; specifies which part of the input PDB is to be analyzed for CC	`"all"`
10	residue selection; specifies which residues of general selection are to be analyzed for CC	
11	temporary folder path for Stride output	`[pwd]/tmp`
12	backbone analysis only (optional), 0 or 1	-

Table 5: Cross correlation coloring command arguments

5.3 Mid-resolution refinement

We will use ModelMaker to refine the segments that have a low cross correlation to the mid-resolution EM density. The executed command will run Rosetta with the CartesianSampler mover, refolding the selected segments by scoring the outcome with the given density. Furthermore, a relax step scored by the quality of fit to density is run. Additionally, a short MDFF run will be performed.

1 **Preparing the configuration file** Create a new folder called `midres_refinement` and put a new directory `full_length_model` in it, which you copy the last frame PDB from the previous MDFF run to. Rename this PDB file to `rpn11_yeast.pdb`. Copy the configuration file template `refine_rpn11_midres.tcl` from the corresponding tutorial directory to your working directory and adapt the paths as explained before (Tab. 1). Furthermore, copy the density file `rpn11_model_5_2594_density.dx` to the folder. The following lines configure our refinement protocol run:

- **line 27 and 28:**
  define the variables `nstruct` and `bestN`, which will be used for the number of decoys to generate and the best N structures to output, respectively.
- **line 29:** `set ch_seg [list "rpn11" "A" "AP1"]`
  sets a list for the chains and segments for PSF generation. In this case, we have one protein chain that we name "rpn11", the PDB chain identifier is "A" and the `segname` is `AP1`.
- **line 36:** `make_mrc_file rpn11_model_5_2594_density`
  creates an .mrc file from the given .dx file for Rosetta.
- **line 38:** `start_rosetta_refine ...`
  starts the Rosetta refinement run. The single arguments are explained in Tab. 6. The whole line should look like

  ```
  start_rosetta_refine rpn11_midres_bb rpn11_yeast [list "resid
  212 to 228" "resid 296 to 306"] 1 1 rpn11_model_5_2594_density
  7.7 -0.3 $bestN $nstruct
  ```

  which refines the amino acids 212 to 228 and 296 to 306 from our input structure to the density.
- **line 39:** `start_mdff_run ...`
  We can now automatically pass the resulting Rosetta decoys to MDFF. Therefore, the job name for the MDFF run ought to be the same as for the corresponding Rosetta run, as well as the input PDB name. The whole line should look like

  ```
  start_mdff_run rpn11_midres_bb rpn11_yeast rpn11_model_5_2594_density
  "not (resid 212 to 228 or resid 296 to 306)" 0.6 400 20000
  7.7 $bestN
  ```

  and the individual arguments are explained in Tab. 7

2 **Running the refinement in VMD** Run the following command:

```
vmd -dispdev text -e refine_rpn11_midres.tcl
```

If no errors occur, you can find the final results in the MDFF output folder, including RMSD and CC plots. The last frame of the MDFF run has also been written to your disk. If you compare the input structure to the refined model, you should see structural improvements in the regions you selected for refinement. In case no improvements are visible, you can play around with the Rosetta density score argument to force Rosetta only to produce models that have a high density score. Be careful that overfitting is most likely occuring, if the score value is too high!

arg.	description	tutorial example
1	job name	`rpn11_midres_bb`
2	input structure name (file in full_length_model)	`rpn11_yeast`
3	list of selection texts for refinement	`[list "resid 212 to 228" "resid 296 to 306"]`
4	anchor residue for CoordinateConstraint (choose a residue outside the refined region!)	`1`
5	CartesianSample flag, 0 or 1	`1`
6	map name, dx format required	`rpn11_model_5_2594_density`
7	map resolution	`7.7`
8	Rosetta density score, low value means that the density weighted lower	`-0.3`
9	best N structure files to write to `full_length_model` folder	`$bestN`
10	structure number to generate	`$nstruct`

Table 6: Rosetta refinement command arguments

arg.	description	tutorial example
1	job name	`rpn11_midres_bb`
2	input structure name (file in full_length_model)	`rpn11_yeast`
3	map name, dx format required	`rpn11_model_5_2594_density`
4	MDFF fix selection text	`"not (resid 212 to 228 or resid 296 to 306)"`
5	grid scale	`0.6`
6	minimization steps	`400`
7	MDFF run steps	`20000`
8	map resolution	`7.7`
9	best N structures to run in MDFF	`$bestN`

Table 7: MDFF command arguments

5.4 Structure check

There is a yeast Rpn11 model available that is derived from a high resolution 3.5 $\AA$ EM density (PDB 3jck, EMDB 6479). To check your built model, we will transition our model to the high-resolution EM density. First, we need to do a rigid body docking to the Rpn11 density from the high-resolution density. Then, we will perform an interactive MDFF run to further refine the model.
Create a new folder `structure_comparison`. Furthermore, create the folder `dock_density` in it.

1 Installing the Situs package
Go to the Situs website (http://situs.biomachina.org) and follow the download and installation instructions. Add the binary folder to your $PATH environment variable.

2 Downloading the Rpn11 model from PDB 3jck
Type the following commands in VMD to obtain the Rpn11 model from PDB 3jck:

```
mol new 3jck
[atomselect top "chain G"] writepdb rpn11_yeast_3jck.pdb
```

3 Extracting the Rpn11 density
Copy the script `crop_density.tcl` from `3.3.structure_comparison` and

the density file `emdb_6479.mrc` to your directory. Set the `packagePath` variable to the location of the ModelMaker plugin files. Run the `crop_density.tcl` script in VMD text mode to yield the density of Rpn11 in EMDB 6479 `rpn11_yeast_3jck_2_6479_density.mrc`.

4 Rigid body docking to the Rpn11 density

Copy the refined Rpn11 yeast model from the last refinement step to the previously created `dock_density` folder and name it `rpn11_yeast_midres.pdb`. As well, copy the `rpn11_yeast_3jck_2_6479_density.mrc` density file to this folder and navigate to it. Run the following command in the terminal:

```
colores rpn11_yeast_3jck_2_6479_density.mrc rpn11_yeast_midres.pdb
-res 3.9 -nprocs ⟨cores⟩
```

Adjust ⟨cores⟩ to the number of cores you want to run Situs on.

Rename the output `col_best_001.pdb` to `rpn11_yeast_midres_docked_3jck.pdb`.

5 Interactive MDFF run

As the Rpn11 model from EMDB 6470 is in another conformational state than our predicted model, we need to further adapt the conformation to the density. To do so, perform an interactive MDFF run with QwikMD as explained in section 3.2. Use the previously docked PDB file `rpn11_yeast_midres_docked_3jck.pdb` and the cropped density file `rpn11_yeast_3jck_2_6479_density.dx`. You can then superimpose the yeast Rpn11 models in VMD, color them differently and compare the structures.

6 Homology model

For the 26S proteasome in human, a 3.9 Åcryo-EM density is available, which we will use to reap the full benefits of our model building tool for high resolutions.

6.1 Building a homology model with MODELLER

1 Download and install MODELLER
Go to the MODELLER website (https://salilab.org/modeller/tutorial/), download the newest version of MODELLER and request a license. Install MODELLER on your workstation.

2 Sequence alignment
Go to http://www.uniprot.org/align/nd align the sequence of yeast Rpn11 (Uniprot ID P43588) and human Rpn11 (Uniprot ID O00487). The alignment reveals that the structurally resolved part of the sequence from amino acid 23 to 306 of yeast Rpn11 aligns with the human sequence range of 27-310 with a similarity of 81 %. Download the human Rpn11 fasta sequence and extract the range from amino acid 27 to 310 to a new text file called `rpn11_human_27-310.fasta`. You can use the aforementioned `subrange.py` script to do so. For MODELLER input, create the file `rpn11_human_27-310.seq` by adding the header lines

```
>P1;rpn11_human_27-310
sequence:::::::::
```

to the fasta file. Furthermore, append an asterisk (*) to the last amino acid code.

3 Preparing MODELLER run
Create a new folder `homology` and another folder `modeller_run` in it. Navigate to the `modeller_run` directory and copy the Python scripts from the corresponding tutorial files to it. Furthermore, copy the refined model of yeast Rpn11 `rpn11_yeast_midres.pdb`, `rpn11_human_27-310.seq`, `rpn11_human_27-310.fasta` and `rpn11_yeast_23-306.fasta` to the MODELLER folder. Make the following changes to the `align2d.py` script:

- **line 5:** Change the argument to `file=''` to `rpn11_yeast_midres.pdb`.

Execute the `align2d.py` script by running the following command in the terminal:

```
mod9.17 align2d.py
```

4 Running MODELLER
If the alignment script ran without errors, you can build the homology model by simply running

```
mod9.17 model-single.py
```

in the terminal to generate a set of 10 homology models. The `assess_methods` section in `model-single.py` defines the scoring functions for the homology model. After the run, we will continue with the model that has the lowest score in the DOPEHR column, specified in the log file `model-single.log`.

6.2 Rigid body docking to the Rpn11 human density

To position the generated homology model in the Rpn11 human high-resolution density, we need to perform a rigid body docking. As docking software, we will use the Situs `colores` tool.

1 **Installing the Situs package** Go to the Situs website (http://situs.biomachina.org) and follow the download and installation instructions. Add the binary folder to your $PATH environment variable.

2 **Docking the Rpn11 yeast model to the human density** Create a folder `dock_density` in the homology model working directory. Copy the density file `rpn11_human_3_3.9_density.mrc` from the corresponding tutorial directory to `dock_density`, as well as the best homology model PDB structure of the MODELLER run. Rename the MODELLER output PDB to `rpn11_human_27-310_notDocked.pdb`.
Run the following Situs command:

```
colores rpn11_human_3_3.9_density.mrc rpn11_human_27-310_notDocked.pdb
-res 3.9 -nprocs ⟨cores⟩
```

Adjust ⟨cores⟩ to the number of cores you want to run Situs on.

Rename the output `col_best_001.pdb` to `rpn11_human_27-310.pdb`.

6.3 Interactive MDFF run to the human Rpn11 density

In the `homology` folder, create a folder `mdff` and navigate to it. Copy the docking output PDB `col_best_001.pdb` to it, as well as the density file `rpn11_human_3_3.9_density.dx`, provided in the tutorial files. As Situs destroys both correct amino acid numbering and chain identifier, use the script `4.homology/mdff/renumber.tcl` to obtain the correctly labeled PDB file `col_best_001-numb.pdb` and rename it to `rpn11_human_27-310.pdb`.

According to section 3.2, interactively fit the docked structure to the high-resolution density. Save the last frame as `rpn11_human_27-310_fit.pdb` To elucidate regions where the structure needs to be refined to the high-resolution density using Rosetta, you can run a CC coloring as before. You can use the template configuration file in `4.homology/mdff/cccolor` to get the following CC-colored files:

1. `rpn11_human_27-310_fit-ss.pdb:` Secondary structure elements are colored by their CC to the density. If a segment does not fit the density, the CartesianSampler protocol ought to be used to fix this.

2. `rpn11_human_27-310_fit-res-bb.pdb:` The backbone atoms are colored according to their CC with the density. For outliers, we can later on do a backbone refinement to the high-resolution density.

3. `rpn11_human_27-310_fit-res.pdb:` Single residue sidechain outliers can be identified in this pdb file.

With the CC coloring, regions for every single step in the high-resolution density refinement are determined.

7 High-resolution real-space refinement

We will use the ModelMaker plugin with Rosetta and MDFF to refine our homology model fitted to the density to a final structure, taking advantage of the 3.9 $\mathring{A}$ density for Rpn11. This allows a high density weight during the Rosetta runs, as well as high grid scales for the MDFF runs. In this section, we will perform three different refinement steps, so that we reach a convergence to the density by increasing a) the Rosetta density weight and b) the grid scale during the corresponding MDFF run. The refinement steps are:

1. Secondary structure outliers: refinement with CartesianSampler protocol, fitting the given residue ranges to the density + MDFF run
2. backbone outliers: the amino acids refined in the first step are further improved by only running a refinement of the backbone atoms + MDFF run
3. sidechain refinement of all amino acids with a high Rosetta density weight + MDFF run

As before, we only need one configuration file which we define every refinement step in.

1 Preparing the configuration file
Like in the runs before, create a new folder named `highres_refinement` and copy the corresponding template configuration file `5.highres_refinement/refine_rpn11_highres.tcl` to it. Create a directory named `full_length_model` in the `highres_refinement` folder and copy the outcoming structure of the last MDFF run to it. Rename the PDB file to `rpn11_human_23-310_fit.pdb`. Make the usual changes in the header of the file to match your workstation configuration. To run NAMD, we need to add a few lines to the file:

- `set path <path that contains namd2 executable>`
- `set topdir <topology and parameter file directory>`
- `set topfiles <list of topology file names>`
- `set parfiles <list of parameter file names>`
- `set ch_seg [list "rpn11" "A" "AP1"]`

In the `ch_seg` variable, we define a list of chains and segnames for PSF generation. The arguments in the list denote `[list <name> <chain> <segname>]`, where the `<name>` argument is arbitrary. The `mutations` list can specify a list of mutations for PSF generation. Here, the list elements denote `[list <segname> <resid> <new-resname>]`. In this case, we don't need any mutations in the PSF, so we define it as an empty list. The `start_rosetta_refine` arguments have already been explained in

Tab. 6 as well as the MDFF run arguments in Tab. 7. In this configuration file, we first use the CartesianSampler to refine segments where both backbone and secondary structure need to be refined. You can examine the residue ranges by CC coloring. The amino acids you want to refine depend on the fitness on your given model to the density. The Rosetta density weight value is increased from step to step, so that the structure more and more converges to its native state, avoiding overfitting in the first place.
`start_rosetta_refine_sidechains_density` takes the same arguments as the `start_rosetta_refine` command, **except for the CartesianSampler** argument you have to leave out.

2 Running the high-resolution refinement Run

```
vmd -dispdev text -e refine_rpn11_highres.tcl
```

and wait until all steps are finished. The outcome PDB file of every step will be automatically copied to the `full_length_model` folder.

3 CC coloring of the final model Use the already explained CC coloring configuration file to compare your input and output models to the last high-resolution refinement step. Load the colorings of secondary structures, backbone and residues to VMD and compare them. If you further want to refine a range of amino acids, you can iteratively repeat the high-resolution step until you reach the maximum of accuracy. However, Figs. 1 to 4 show the power and usability of the automated refinement with Rosetta and MDFF, wrapped by VMD and simply configurable in one TCL script.

The region around K277 shows low fitness to the high-resolution density before the refinement. After the protocol run described before, the backbone and the sidechain have adapted a decent conformation due to Rosetta rotamer optimization, relax steps and MDFF (Figs. 1 to 4).

Figure 1: Backbone before refinement

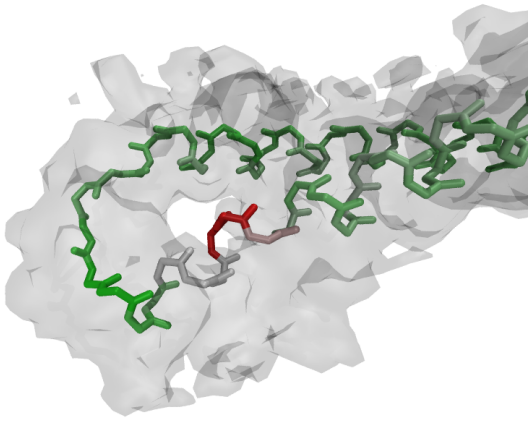


Figure 2: Backbone after refinement

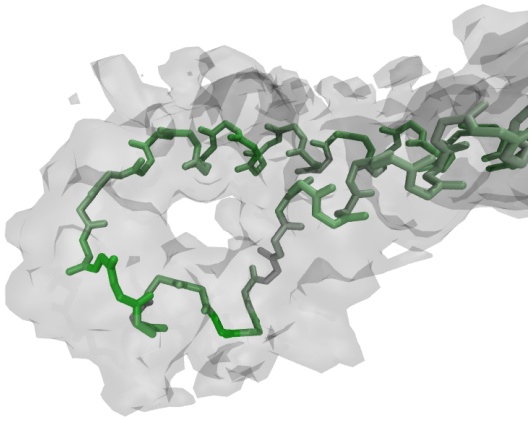


Figure 3: Residues before refinement

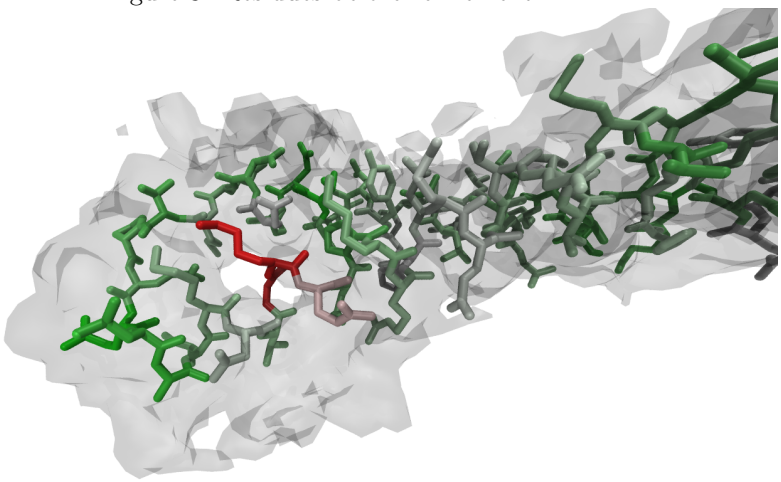


Figure 4: Residues after refinement