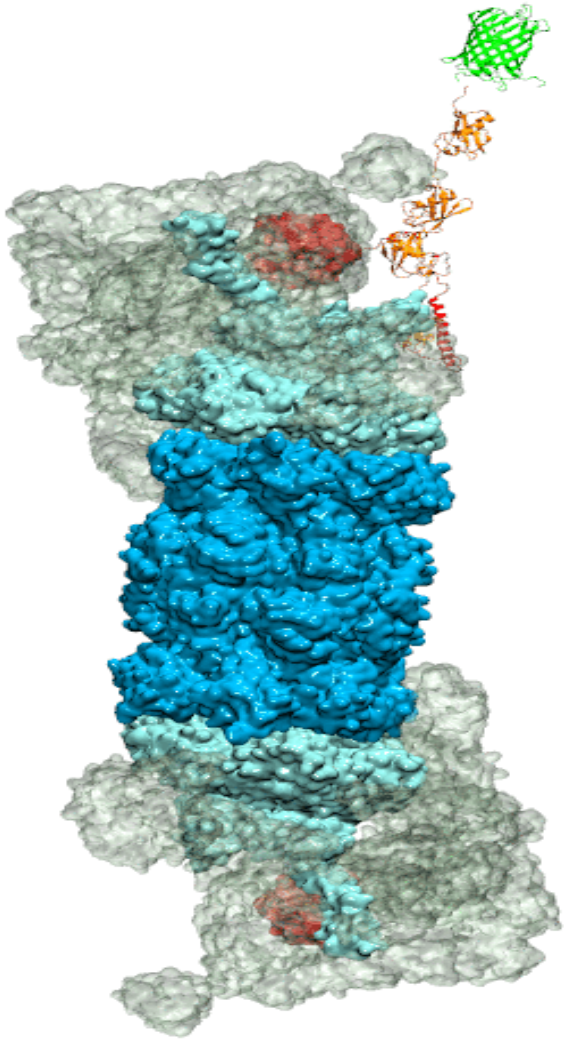
Structural Bioinformatics:

Deriving Protein Models from cryo-EM Images



Tutorial by Penelope Pesara and Till Rudack

Content

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-friendly 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 and a structurally unresolved protein insertion of the X-ray structure of the proteasomal subunit Rpn11 in yeast (chain B of PDB ID 4OCM). It will first be fit it to the mid-resolution (7.7 ˚A) and after to the high-resolution cryo-EM density of Rpn11 derived from the cryo-EM density of the yeast 26S proteasome (EMD-2594).

2 Required software

2.1 ModelMaker

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 will carry out. Download link: 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

Rosetta can be downloaded 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 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/BuildDocumentation). 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

In this first part we will concentrate on creating a structural model for the rpn11 subunit of the 26S proteasome in yeast. The 26S proteasome is a protein complex that degrades proteins marked by the protein ubiquitin. The subunit rpn11 is responsible for the removal of ubiquitin tags initiating substrate degradation. In our example the C-terminal tail from amino acid 288 to 306 of chain V within the rpn11 subunit is missing. In the next few steps we will explain how to create a model containing the full length of this subunit by the help of the robetta server and how to generate a structural model with the `modelmaker abinitio` and `analyse` option.

Pictures: Start structure of rpn11 containing amino acid 18-288 illustrated with Pymol.

3.1 Structure prediction

At first we are going to predict a model for the missing C-terminal tail from amino acid 288 to 306 of chain V based on an existing PDB structure from amino acid 18-288 of the rpn11 subunit.

3.1.1Generating the input files:

Save the starting PDB structure "rpn11_start_structure_18-288.pdb" from the master folder to a folder you create called "Terminus". In this structure the residues from 288-306 are missing. In order to generate a full length model containing amino acid 18 to 306 we will need two library files containing internal coordinates for the target sequence structure. To generate them first a fasta file including the primary structure of the desired sequence (from 18-306) of rpn11 must be cut. Save the file P43588.fasta from the master folder to your working directory. To obtain the desired sequence of amino acid 18 to 306 we have to cut the fasta file. Use the script called cut_fasta_18-306.tcl which you will also find in the master folder. With the following command you will execute the script:

vmd –dispdev text -e cut_fasta_18-306.tcl > cut_fasta_18-306.log

With the command "*vmd –dispdev text –e"* you will execute the program *vmd* without opening it. Additionally you created a new file "cut_fasta_18-306.log" which contains what the program is executing and possible errors. You could name this file as you like but it make sense to choose a name which reflects the content of the task you are working on. The script you executed includes the command:

modelmaker seqsub -i fasta P43588 -o rpn11_18-306.fasta -start 18 -end 306

This ModelMaker seqsub command cuts the input file (-i fasta P43588.fasta) and writes it with the "-o" option into a new fasta file called rpn11_18-306.fasta.

Alternatively all ModelMaker commands can also be executed in the TK console within *vmd* or directly in your terminal. Thus you can differ between executing the prepared scripts or using directly the commands. This refers to all steps.

Now with the prepared file "rpn11_18-306.fasta" we are able to generate two library files with the help of the Robetta server. 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. Within the framework of this tutorial the two library files (rpn11_18-306.frag3, rpn11_18-306.frag9) already exist and can be saved from the master directory to your working folder (Path: master/Fold_termini/Input). An explanation for generating the fragment files is contained in the attachment.

3.1.2 Building one complete model for the target amino acid sequence:

Now with all necessary input files we are able to build a structural model containing the complete subunit rpn11 from amino acid 18-306. Make sure all inputs are saved to your directory:

1. rpn11_18-306.fasta
2. rpn11_18-306.frag3 and rpn11_18-306.frag9
3. rpn11_start_structure_18-288.pdb

Save the script called "full_length_model.tcl" as well and execute it with the command:

vmd –dispdev text -e full_length_model.tcl > full_length_model.log

It includes the ModelMaker command:

modelmaker full_length_model -template rpn11_start_structure_288-306.pdb -fragfiles {rpn11_18-306.frag9 rpn11_18-306.frag3}-fasta rpn11_18-306.fasta –resstart 18

We will in fact get a complete structure but by no means optimal. The complete model will be saved to a new PDB file called: "rpn11_18-306_full_length.pdb". Because our model begins at amino acid 18 we have to use the *"-resstart"* command. If you didn't get an output open the

full_length_model.log file and look for possible errors. To obtain an optimized structure model we will fold the protein termini in the following step.

3.2 Folding protein termini

Now that all of our input files are generated, we are ready to fold the protein termini. To do so, go to the master folder and copy the script "fold_termini.tcl" to your working directory and run it with:

vmd –dispdev text -e fold_termini.tcl > fold_termini.log

Alternatively type the commands from the script into your terminal or in the TKconsole:

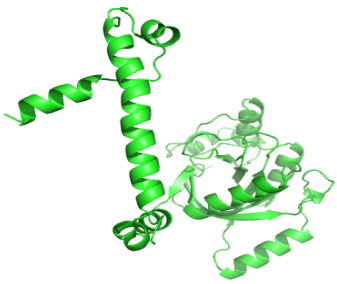
modelmaker abinitio -np 4 -model rpn11_18-306_full_length.pdb -jobname rpn11_18-306 -anchor "resid 200" -fragfiles {{rpn11_280-306.frag9 rpn11_280-306.frag3}} -sel "resid 288 to 306" -nstruct 4

This first step will calculate the structures. With *"-np 4"* you define the number of cores you would like to use. Check how many "core(s) per socket" are available for your calculations before changing (You can check by tying *"lscpu"* into your terminal). With the *"-model"* you define your input and with the *"-jobname"* option you can give your job a name if you like. Define with the *"-anchor"* a residue you like to hold fixed. Be careful this residue isn't part of your area to fold which you determine with the *"-sel "resid 288 to 306""* option. In the end you can choose how many structures to generate. In principle it is advisable to calculate at least as many structures as cores used. You can find the generated models in the workdir/run-rpn11_288-306/pdb-out/ folder. Within the next command your generated structures will be analysed, and thus one aligned structure will be created:

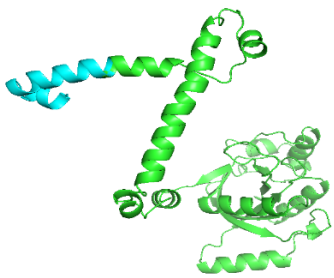
modelmaker analyze -model rpn11_18-306_full_length.pdb pdb -jobname rpn11_18-306 -template rpn11_18-306_full_length.pdb -nstruct 4 -align_template "resid 200" -comps {{ss 288 to 306 "V"} {cluster 288 306 "V" 4}}

It will produce a gnuplot file including a histogram showing the probability of every secondary structure for each residue and a .txt file containing the same probabilities in text format.

For comparison the start structure and the created output structure are shown in the following:



The rpn11 start structure with residue 18-288.



The rpn11 end structure with residue 18-306. Residue 288-306 illustrated in cyan.

To make the calculation of termini folding more practicable and easier to execute, a script containing all the steps you have done separately has been provided. You can find this script ("run_fold_termini.tcl") in the master folder. If you would like to run it you have to create a new folder and copy the required inputs and the script into it:

1. rpn11_start_structure_18-288.pdb
2. 2. rpn11_18-306.frag3 and rpn11_18-306.frag9
3. P43588.fasta
4. run_fold_termini.tcl

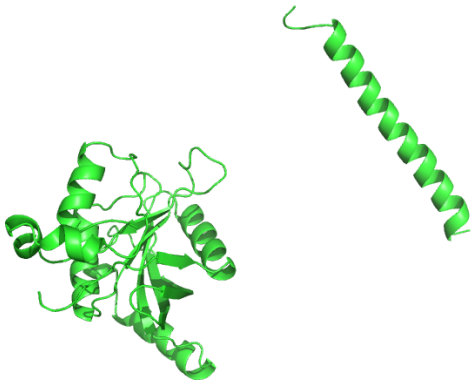
Run it with the command:

vmd –dispdev text -e run_fold_termini.tcl > all_fold_termini.log

3.2.1 Using different clustering methods

4 Folding protein insertion using ModelMaker

Often structural segments within the protein are missing which can also be completed by the same algorithm used to finish terminal structures. To fold insertions the proceeding stay the same as folding protein termini.



Picture: rpn11 start structure from residue 18-200 and 271-306 illustrated in Pymol.

4.1 Structure prediction

In this section we are going to predict a structural model for the missing part from amino acid 200 to 271 of chain V within the rpn11 subunit of the 26S proteasome.

8.1 Building one complete model for the target amino acid sequence

Save the PDB structure "rpn11_start_structure_18-200_271-306.pdb" from the master folder to a folder you create called "Insertion". In this structure the residues from 200-271 are missing. In order to generate a full length model containing amino acid 18 to 306 we will need the same input files as described above:

1. rpn11_18-306.fasta

2. rpn11_18-306.frag3 and rpn11_18-306.frag9

3. rpn11_start_structure_18-200_271-306.pdb

Save the script called "full_length_model.tcl" and change the residues we want to predict as described in the following:

set start "200"

set end "271"

Execute the script with the following command:

vmd –dispdev text -e full_length_model.tcl > full_length_model.log

The full length model will be saved to a new PDB file called: "rpn11_18-306_full_length.pdb". Check if your output file contains residue 18-306. To obtain the secondary structure we will fold the protein insertion in the following step.

8.2 Folding protein insertion

To fold insertion go to the master folder and copy the script "fold_insertion.tcl" to your working directory and run it with:

vmd –dispdev text -e fold_insertion.tcl > fold_insertion.log

Alternatively you could again type the commands included in your terminal or in the TKconsole:

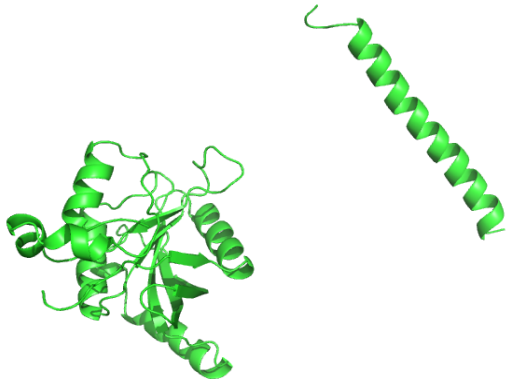
modelmaker insertion -np 4 -model rpn11_start_structure_18-306_full_length.pdb -jobname rpn11_200-271 -fasta rpn11_18-306.fasta -fragfiles {rpn11_18-306.frag9 rpn11_18-306.frag3} -sel "resid 200 to 271" -nstruct 4

modelmaker analyze -insertion yes -model rpn11_start_structure_18-306_full_length.pdb -jobname rpn11_200-271 -template rpn11_start_structure_18-306_full_length.pdb -nstruct 4 -align_template "resid 100 to 180" -comps {{ss 200 to 271 "V"} {cluster 200 271 "V" 4}}

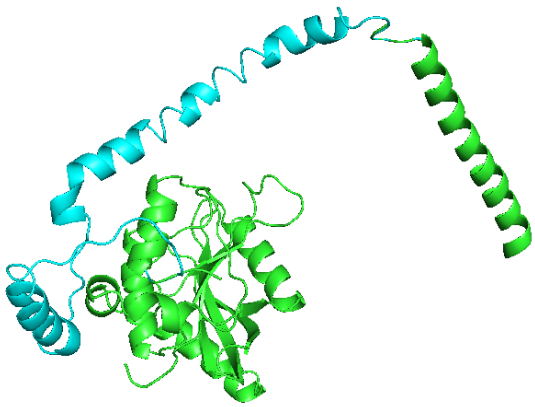
Different to the "fold termini" abinitio calculation which is able to fold more than one structure simultaneously, this insertion folding can only fold one single region at a time. Within the termini abinitio calculation the program is able to load different fragment files in to fold the different

residues thus the fragment files have to be surrounded by two curly braces. Whereas the insertion abinitio command just needs one curly brace to surround the fragment files: *-fragfiles {rpn11_18-306.frag9 rpn11_18-306.frag3}*. The second thing changed is the region to fold and thus the anchor which is here defined as residues 100-180.

When the calculation is complete, go to the folder "workdir/run-rpn11_200-271/analysis/ss_200-271". Here you will find another histogram and a .txt file including the probabilities of every residue for each secondary structure.



The start structure (residues 18-200 and 271-306)



The end structure containing residue 18-306. Residue 200-271 illustrated in cyan.

To make the calculation of insertion folding more practicable and easier there also exists a script containing all steps in one, similar to the script "run_fold_termini.tcl" you used in step 6. You can find this script ("run_fold_insertion.tcl") in the master folder. If you would like to run it you have to create a new folder before and copy the required inputs and the script to it:

1. rpn11_start_structure_18-200_271-306.pdb
2. 2. rpn11_18-306.frag3 and rpn11_18-306.frag9
3. P43588.fasta
4. run_fold_insertion.tcl

Run it with the command:

vmd –dispdev text -e run_fold_termini.tcl > all_fold_termini.log

5 Molecular dynamics flexible fitting (MDFF)

The molecular dynamics flexible fitting (MDFF) method can be used to flexibly fit atomic structures into density maps. Create a folder called "MDFF" and copy the script run_mdff.tcl to your working directory. It will perform a mdff run on the base of your created full length model and fit it into a low resolution density map which is already prepared and can be saved from the master folder. Copy the following files and scripts to your working directory:

- rpn11_s1_lowres_3_7.7_moved_density.mrc
- rpn11_s1_lowres_3_7.7_moved_density.dx
- rpn11_s1_lowres_aligned.pdb
- run_mdff_low.tcl

The files rpn11_s1_lowres_3_7.7_moved_density.mrc and rpn11_s1_lowres_3_7.7_moved_density.dx contain the density map in two different file formats. The program *vmd* works with the file format .dx, whereas other programs we will use only read .mrc files. The PDB structure includes the complete model of the rpn11 subunit within the proteasome from amino acid 18-306. With the help of the script "run_mdff_low.tcl we fit the PDB structure into the density map with a low resolution of 7.7 Å. Execute the script run_mdff_low.tcl with the command:

vmd-1.9.4 –dispdev text -e run_mdff_low.tcl > run_mdff.log

You will get a file called "rpn11_mdff-step1-result.pdb" in the folder mdff_step1, which contains the fitted structure. Open *vmd* in your terminal by typing *vmd*. Open the start structure, the rpn11_mdff-step1-result.pdb and the rpn11_s1_lowres_3_7.7_moved_density.dx. Represent the PDB structures in "New Cartoon" and color them differently. Represent the density in "Solid surface" and choose "Transparency" as the material. Study the differences between start and end structure. Overlay the both structures well? Lie the both structures within the density? Take a picture for the protocol. Color the background white (VMD Main -> Graphics -> Color -> Display -> Background -> white) and remove the axes (Display -> Axes -> off).

To generate better results, the mdff and backbone refinement runs were repeated. Within these runs densities of different resolution alternate. In the following we will fit the backbone of our subunit into another density map with a higher resolution of 4.2 Å.

6 Backbone Refinement

Within this Refinement we will fit the atoms within the backbone into a high resolution density map, which is saved to the master folder.

Save the files and scripts into a new directory you call BB_Ref:

- rpn11_s1_3_4.2_density.dx
- rpn11_s1_3_4.2_density.mrc
- rpn11_s1.pdb
- run_bb_refine.tcl

Start the backbone refinement with:

vmd-1.9.4 –dispdev text -e run_bb.tcl > run_bb.log

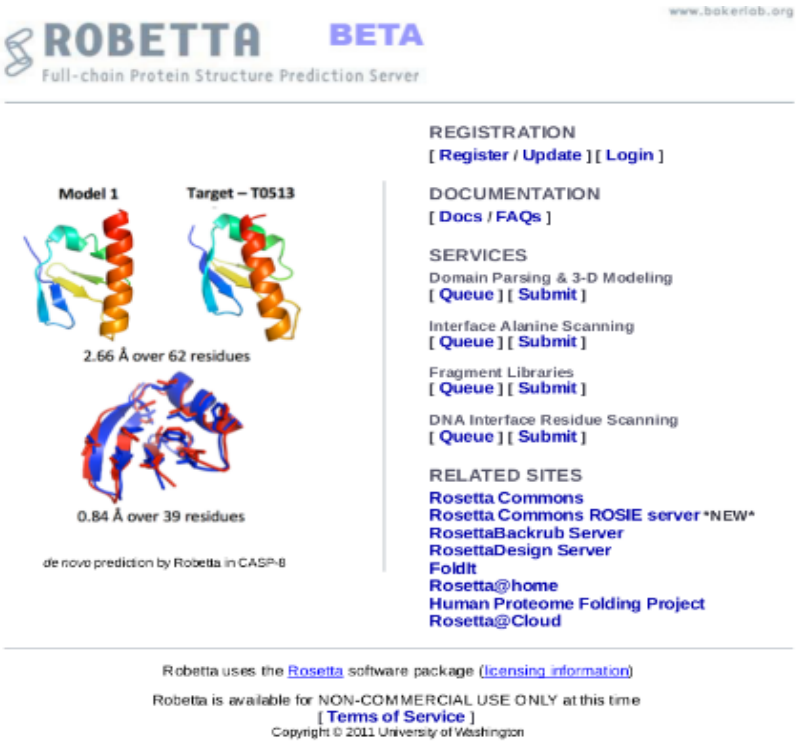
Beside the modelmaker refine command the script contains the following:

-model rpn11_s1.pdb	which defines the model to fit
-anchor "resid 200"	sets the anchor at residue 200
-workdir refine_bb_step1	names the output folder
-sel {"resid 288 to 306"}	defines the residues to fit
-density rpn11_s1_3_4.2_density.mrc	determines the density we want to use
-res 4.2	informs about the resolution of the density
-nstruct 2	defines number of structures to generate
-bestN 1	determines to use the best structure
-jobname rpn11_refine	gives our job a name
-score -0.3	determines the score to consider our density
-np 2	defines number of cores used

As output we get a file called "rpn11_refine_best1.pdb" in our workdir (/workdir/run-rpn11_refine). Open the start and end structure as well as the density (dx file format) in *vmd*. Represent the molecules as described above, compare the results you created with the mdff run with the results you get from the backbone refinement and take some pictures for the protocol.

7 Generating a fragment file with the help of the Robetta server

To generate fragment files on the base of an existing fasta file go to the Robetta server (http://www.robetta.org) and set up an academic user account.



Go to “Fragment Libraries” and submit the target sequence rpn11_18-306.fasta. 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 with the favoured name in your folder