

Modeling missing residues using Modeller tool.

By: H. Rasouli

Pre-requisites:

- 1- The FASTA file of your target protein.
- 2- Its 3D structure. You can get it through PDB databank
- 3- Python scripts
- 4- Installing the latest version of Modeller.
- 5- A working folder.

Introduction

The term “missing residues” is generally described to provide information about some residues that their topology and arrangement in the 3D structure of a target protein have lost. For this purpose, experts suggested that during computational calculations these types of residues can cause a problem for the interpretation of results. It has commonly been assumed that the position of missing residues in the structure of protein sequence can change the outcomes of computational results either through restricting the chance of a ligand to interact with a specific region of protein or producing false positive results. To solve this problem, it is highly recommended that before starting virtual analyses the structure of target proteins should be optimized for eliminating the number of missing residues if any. There are several tools widely endowed their time to prepare the right conformation of proteins via specific algorithms for remodeling the topology of missing residues. One of the most important suite's users can apply for this purpose is Modeller software developed by SaliLab, USA (<https://salilab.org/modeller/>). This command-line based tool, use python scripts for solving the issue of missing residues. So, after installation of this tool and getting an academic license from the developer website, the users will be able to land up on both feet for working with. Herein, I simplified the method of missing residues modeling for users to help them for saving time and confidence.

Steps

- 1- Select the target protein with missing residues position. I selected the 3D structure of human alpha-glucosidase enzyme that have several missing residues in its structure. This file can be obtained from PDB bank using PDB ID: 5NN8.
- 2- Provide the raw sequence file of your target protein. Note that this file should be prepared in “.ali” format. For this purpose, you can simply copy your sequence within the following box. Instead of bold words, the users can append their file name.

```
>P1;gluc
sequence:gluc:::::::::0.00: 0.00
QCDVPPNSRFDCAFDKAITQEQCEARGCCYIPAKQGLQGAQMGPWCFFPPSPYSYKLENLSSSEMGYTATLTRTTPTFFPKDILTLRLDVM
ETENRLHFTIKDPANRRYEVPLETPRVHSRAPSPLYSVEFSEEPFGVIVHRQLDGRVLLNTTVAPLFFADQFLQLSTSLPSQYITGLAEHLSP
LMLSTSWTRITLWNRDLAPTGANLYGSHPFYLALEDGGSAGHVFLLSNAMDVVLQSPALSWRSTGGILDVYIFLGPEPKSVVQQYLDVVG
YFPMPPYWGLGFHLCRWGSSTAITRQVVENMTRAHFPLDVQWNLDDYMDSRDFTFNKDGFRDFPAMVQELHQGGRRYMMIVDPAISSSGPA
GSYRPHYDEGLRRGVFITNETGQPLIGKVWPGSTAFPDFTNPTALAWWEDMVAEFHDQVPFDGMWIDMNEPSNFIRGSEDGCPNNELENPPYVP
GVVGGTLQAATICASSHQFLSTHYNLNLYGLTEAIAASHRALVKARGTRPFVISRSTFAGHGRYAGHWTGDVWSSWEQLASSVPEILQFNLLG
VPLVGADVCGFLGNTSEELCVRWTQLGAFYPFMRNHNLSLSLPQEPYSFSEPAQQAMRKALTLRYALPHLYTLFHHQAHVAGETVARPLFLEF
```

```
PKDSSTWTVDHQLLWGEALLITPVLQAGKAEVTGYFPLGTWYDLQTVPIEALGSLPPPPAAPREPAIHSEGQWVTLPAPLDTINVHLRAGYII  
PLQGPGGLTTTESRQQPMALAVALTKGGEARGELFWDDGESLEVLERGAYTQVIFLARNTIVNELVRVTSEGAGLQLQKVTVLGVATAPQQVL  
SNGVPVSNFTYSPDTKVLDDICVSLLMGEQFLVSWC*
```

After preparing your sequence, you should save it in “.ali” format. For this purpose, you can use Notepad++ tool. Note that this file must copy into your working folder. Next, add the pdb file of your target protein to you work folder. Now, you have two files including sequence file and 3D conformation. In the next step, you should perform a local alignment between your sequence and pdb file.

- 3- The easiest way to do alignment between your sequence and PDB file is using the following script. Instead of bold words you can add the name of your sequence and pdb files. Note that the following script should be saved in “.py” format. Simply copy and paste scripts words into Notepad++ tool and then save it as “**ScriptA.py**”

```
from modeller import *  
  
env = environ()  
aln = alignment(env)  
mdl = model(env, file='5NN8', model_segment=('FIRST:A','LAST:A'))  
aln.append_model(mdl, align_codes='5NN8', atom_files='5NN8.pdb')  
aln.append(file='gluc.ali', align_codes='gluc')  
aln.align2d()  
aln.write(file='alignment.ali', alignment_format='PIR')  
aln.write(file='alignment.pap', alignment_format='PAP')
```

In the next step, try to open modeler command line in that path your working folder is. Also, you can use “cd” command to usher its command line to where the input files are stored. Then, using **> mod9.21 ScriptA.py** build your alignment profile for the target inputs. You can find the results of alignment in “alignment.ali” file in your working folder. Since building alignment between your input queries takes time, you should wait until time the software gives you the output records. As shown in the following figure, the missing residues found in the structure of pdb file are depicted with dashes. Another python script helps you to re-model these residues.

```
>P1;5NN8
structureX:5NN8.pdb: 81 :A:+844 :A:MOL_ID 1; MOLECULE LYSOSOMAL ALPHA-GLI
QCDVPPNSRFDCAPDKAITQEQCEARGCCYIPAKQG-----QPWCFFPPSYPSYKLENLSSSEMGYTATLTRT
TPTFFPKDILTRLRDVMMETENRLHFTIKDPANRRYEVPLEA-----PSPLYSVEFSEEPFGVIVHRQLDGRV
LLNTTVAPLFFADQFLQLSTSLPSQYITGLAEHLSPLMLSTSWTRITLWNRDLAPTGPANLYGSHPFYLALEDGG
SAHG VFLLNSNAMDVVLQPSPALSWRSTGGILDVYIFLGPEPKSVVQQYLDVVGYPFMPYWG LGFHL CRWGYSS
TAITRQVVENMTRAHFPLDVQWNDLDYMSRRDFTFNKDGFRDFPAMVQELHQGGRYMMIVDPAISSSGPAGSY
RPYDEGLRRGVFITNETGQPLIGKVWPGSTAFPDFTNPTALAWWEDMVAEFHDQVPFDGMWIDMNEPSNFIRGSE
DGC PNNELENPPYVPGVVG GTLQAATICASSHQFLSTHYNLHNL YGLTEA IASHRALVKARGTRPFVISRSTFAG
HGRYAGHWTGDVWSSWEQLASSVPEILQFNLLGVPLVGADVCGFLGNTSEELCVRWTLGAFYPFMRNHNLSLLSL
PQEPYSFSEPAQQAMRKALTRYALLPHLYTLFHQAHVAGETVARPLFLEFPKDSSTWTVDHQLLWGEALLITPV
LQAGKAEVTGYFPLGTWYDLQTVPIEE-----PAIHSEGQWVTLPAPLDTINVHLRAGYIIPLQGGPG
LTTTESRQQPMALAVALTKGGEARGELFWDDGESLEVLERGAYTQVIFLARNTIVNELVRVTSEGAGLQLQKVT
VLGVATAPQQVLSNGVPVSNFTYSPDTKVLDI-VSLLMGEQFLVSWC*
```

```
>P1;gluc
sequence:gluc:      : :      : :: 0.00: 0.00
QCDVPPNSRFDCAPDKAITQEQCEARGCCYIPAKQGLQGAQMGPWCFFPPSYPSYKLENLSSSEMGYTATLTRT
TPTFFPKDILTRLRDVMMETENRLHFTIKDPANRRYEVPLETPRVHSRAPSPLYSVEFSEEPFGVIVHRQLDGRV
LLNTTVAPLFFADQFLQLSTSLPSQYITGLAEHLSPLMLSTSWTRITLWNRDLAPTGPANLYGSHPFYLALEDGG
SAHG VFLLNSNAMDVVLQPSPALSWRSTGGILDVYIFLGPEPKSVVQQYLDVVGYPFMPYWG LGFHL CRWGYSS
TAITRQVVENMTRAHFPLDVQWNDLDYMSRRDFTFNKDGFRDFPAMVQELHQGGRYMMIVDPAISSSGPAGSY
RPYDEGLRRGVFITNETGQPLIGKVWPGSTAFPDFTNPTALAWWEDMVAEFHDQVPFDGMWIDMNEPSNFIRGSE
DGC PNNELENPPYVPGVVG GTLQAATICASSHQFLSTHYNLHNL YGLTEA IASHRALVKARGTRPFVISRSTFAG
HGRYAGHWTGDVWSSWEQLASSVPEILQFNLLGVPLVGADVCGFLGNTSEELCVRWTLGAFYPFMRNHNLSLLSL
```

- 4- Modeling script will apply the “*alignment.ali*” file for modeling the missing residues. In the following script, “*knowns*” item is your pdb file and “*sequence*” item is its complete sequence you have. Also, *alignment.ali* is alignment file you gained by running the previous script. Like previous step, replace the name of your files in the defined sections and then try to run this script. You can call it **ScriptB.py**. This scripts enables you to produce n numbers of 3D models based on your complete sequence. Here, I set it to produce 5 models based on the primary template. The script also generates a log file in txt format so that you can find the score and other feature of each model.

```
from modeller import *
from modeller.automodel import *
























log.verbose()
env = environ()
env.io.atom_files_directory = ['.', '../atom_files']

a = loopmodel(env, alnfile = 'alignment.ali',
              knowns = '5NN8', sequence = 'gluc')
a.starting_model = 1
a.ending_model = 5

a.loop.starting_model = 1
a.loop.ending_model = 2
a.loop.md_level = refine.fast

a.make()
```

For modeling loops, this script also performs some analysis. You can limit or increase the number of modeled loops based on your expectations.

Name	Date modified	Type	Size
 5nn8	6/1/2019 5:54 AM	PDB Files	1,200 KB
 alignment.ali	6/1/2019 6:13 AM	ALI File	3 KB
 alignment.pap	6/1/2019 6:13 AM	PAP File	4 KB
 gluc.ali	6/1/2019 5:54 AM	ALI File	1 KB
 gluc.B99990001	6/1/2019 6:34 AM	PDB Files	529 KB
 gluc.B99990002	6/1/2019 6:35 AM	PDB Files	529 KB
 gluc.B99990003	6/1/2019 6:36 AM	PDB Files	529 KB
 gluc.B99990004	6/1/2019 6:38 AM	PDB Files	529 KB
 gluc.D00000001	6/1/2019 6:34 AM	D00000001 File	23 KB
 gluc.D00000002	6/1/2019 6:35 AM	D00000002 File	23 KB
 gluc.D00000003	6/1/2019 6:36 AM	D00000003 File	23 KB
 gluc.D00000004	6/1/2019 6:38 AM	D00000004 File	23 KB
 gluc.D00000005	6/1/2019 6:38 AM	D00000005 File	0 KB
 gluc	6/1/2019 6:32 AM	Configuration sett...	529 KB
 gluc.rsr	6/1/2019 6:32 AM	RSR File	6,797 KB
 gluc.sch	6/1/2019 6:38 AM	SCH File	4 KB
 gluc.V99990001	6/1/2019 6:33 AM	V99990001 File	422 KB
 gluc.V99990002	6/1/2019 6:35 AM	V99990002 File	422 KB
 gluc.V99990003	6/1/2019 6:36 AM	V99990003 File	422 KB
 gluc.V99990004	6/1/2019 6:38 AM	V99990004 File	422 KB
 ScriptA	6/1/2019 6:13 AM	Text Document	3 KB
 ScriptA	6/1/2019 5:56 AM	Python File	1 KB
 scriptB	6/1/2019 6:32 AM	Text Document	13 KB

528 KB

Now, everything is prepared and you can get the best model based on its modeling score. To check the quality of your model, ERRAT server can support you for this query. To enhance the quality of your model, performing 20 to 34 ns molecular dynamics simulation using Gromacs tool can optimize the topology and geometry files of your model amino acids.

For more details and finding further scripts, you can check other repositories in my GitHub page.