Protein Structure Modeling Tutorial

Sina Kazemi & Peter Güntert

Part I: Homology Modeling

Introduction

Homology modeling of proteins, also known as comparative modeling, is a computational technique that allows constructing an atomic-resolution model of a "target" protein from its amino acid sequence and an experimental three-dimensional (3D) structure of a related homologous protein (the "template").

Homology modeling relies on the identification of one or more known protein structures likely to resemble the structure of the query sequence, and on the production of an alignment that maps residues in the query sequence to residues in the template sequence. It has been shown that protein structures are more conserved than protein sequences amongst homologues, but sequences falling below 20% sequence identity can lead to very different structures.[1] Evolutionarily related proteins have similar sequences and naturally occurring homologous proteins have similar structures. It has been shown that 3D protein structure is evolutionarily more conserved than expected due to sequence conservation.[2] In homology modeling the sequence alignment and the template structure are used to produce a structural model of the target protein. Because protein structures are more conserved than DNA sequences, detectable levels of sequence similarity usually imply significant structural similarity.[3]

The following special symbols are used in this tutorial:

Further reading

Questions

Critical points

The further reading and questions will be part of the topics for the exam.

In this tutorial, as an example, we will construct a 3D-structural model of the human variant of the

[1] Chothia, C. & Lesk, A. M. The relation between the divergence of sequence and structure in proteins. *EMBO J.* 5, 823–826 (1986).

[2] Kaczanowski, S. & Zielenkiewicz, P. Why similar protein sequences encode similar three-dimensional structures? *Theo. Chem. Acc.* 125, 643–650 (2010).

[3] Martí-Renom, M. A. et al. Comparative protein structure modeling of genes and genomes. Annu. Rev. Biophys. Biomol. Struct. 29 291–325 (2000).

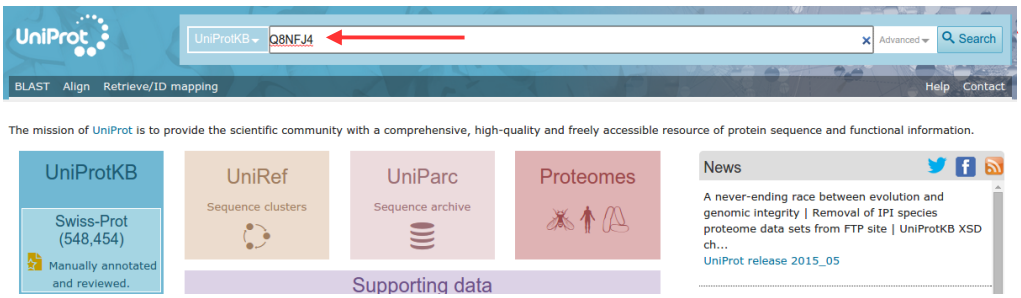
testis-specific kinase-1 (TESK1), which has not been experimentally resolved yet.

This gene product is a serine/threonine protein kinase that contains an N-terminal protein kinase domain and a C-terminal proline-rich domain. Its protein kinase domain is most closely related to those of the LIM motif-containing protein kinases (LIMKs). The encoded protein can phosphorylate myelin basic protein and histone in vitro. The testicular germ cell-specific expression and developmental pattern of expression of the mouse gene suggests that this gene plays an important role at and after the meiotic phase of spermatogenesis.[4]

Getting the target sequence

The first step in our procedure is to get the sequence of amino acids for the human TESK1. We use the UniProt (UNIversal PROTein) database reachable at the website:

http://www.uniprot.org/



Insert in the “Query” field the string:

Q8NFJ4

The field “Search in” can be left at the default:

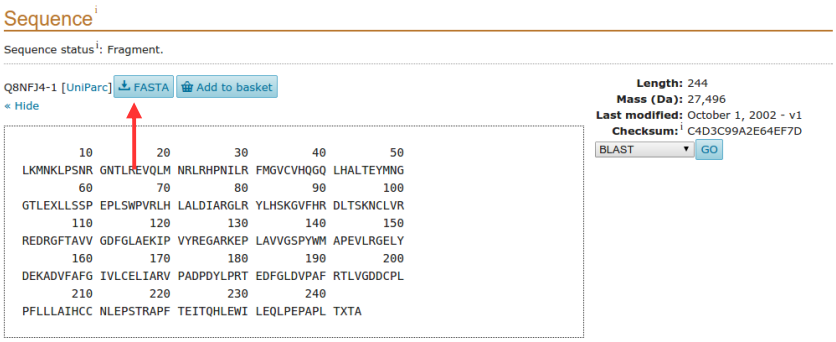
Protein Knowledgebase (UniProtKB)

which specifies the database to use for the search.

It is always a good idea to read as much information as possible from the database details about the file selected.

After studying the page scroll down to the 'Sequence' section and download the protein sequence in FASTA format. If the file opens in a browser window, please use the browser’s File > Save Page As command to save the file in your local directory. Please use the file extension ‘.fasta’, e.g. Q8NFJ4.fasta.

[4] From website: http://www.ncbi.nlm.nih.gov/sites/entrez?Db=gene&Cmd=ShowDetailView&TermToSearch=7016



Taking a look at this file we see a text similar to this:

```
>tr|Q8NFJ4|Q8NFJ4_HUMAN Testis-specific kinase-1 (Fragment) OS=Homo sapiens PE=2 SV=1
LKMNKLPSNRGNTLREVQLMNRLRHPNILRFMGVCVHQGQLHALTEYMNGGTLEXLLSSP
EPLSWPVRLHLALDIARGLRYLHSKGVFHRDLTSKNCLVRREDRGFTAVVGDFGLAEKIP
VYREGARKEPLAVVGSPYWMAPEVLRGELYDEKADVFAFGIVLCELIARVPADPDYLPRT
EDFGLDVPAFRTLVGDDCPLPFLLLAIHCCNLEPSTRAPFTEITQHLEWILEQLPEPAPL
TXTA
```

The first line after the symbol ">" is a comment that provides relevant information about the sequence. The remaining lines are the amino acid sequence in the single letter code.[5]

Template identification

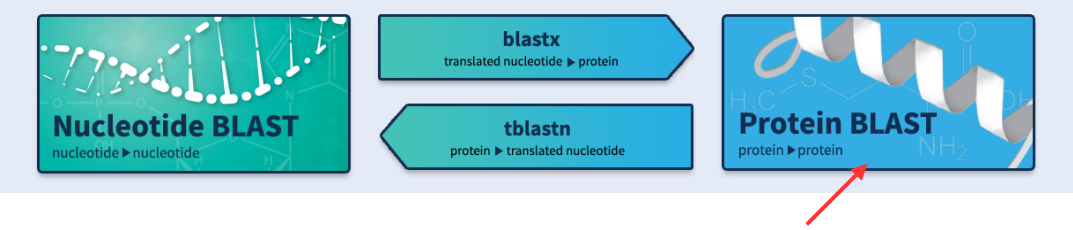
The second step is to find a protein of known 3D structure whose sequence is as similar as possible to our target: the procedure is called sequence alignment.

One of the most widely used algorithms for comparing primary biological sequences, such as amino acid sequences, is BLAST (Basic Local Alignment Search Tool). A BLAST search enables a researcher to compare a query sequence with a library or database of sequences, and identify library sequences that resemble the query sequence above a certain threshold.

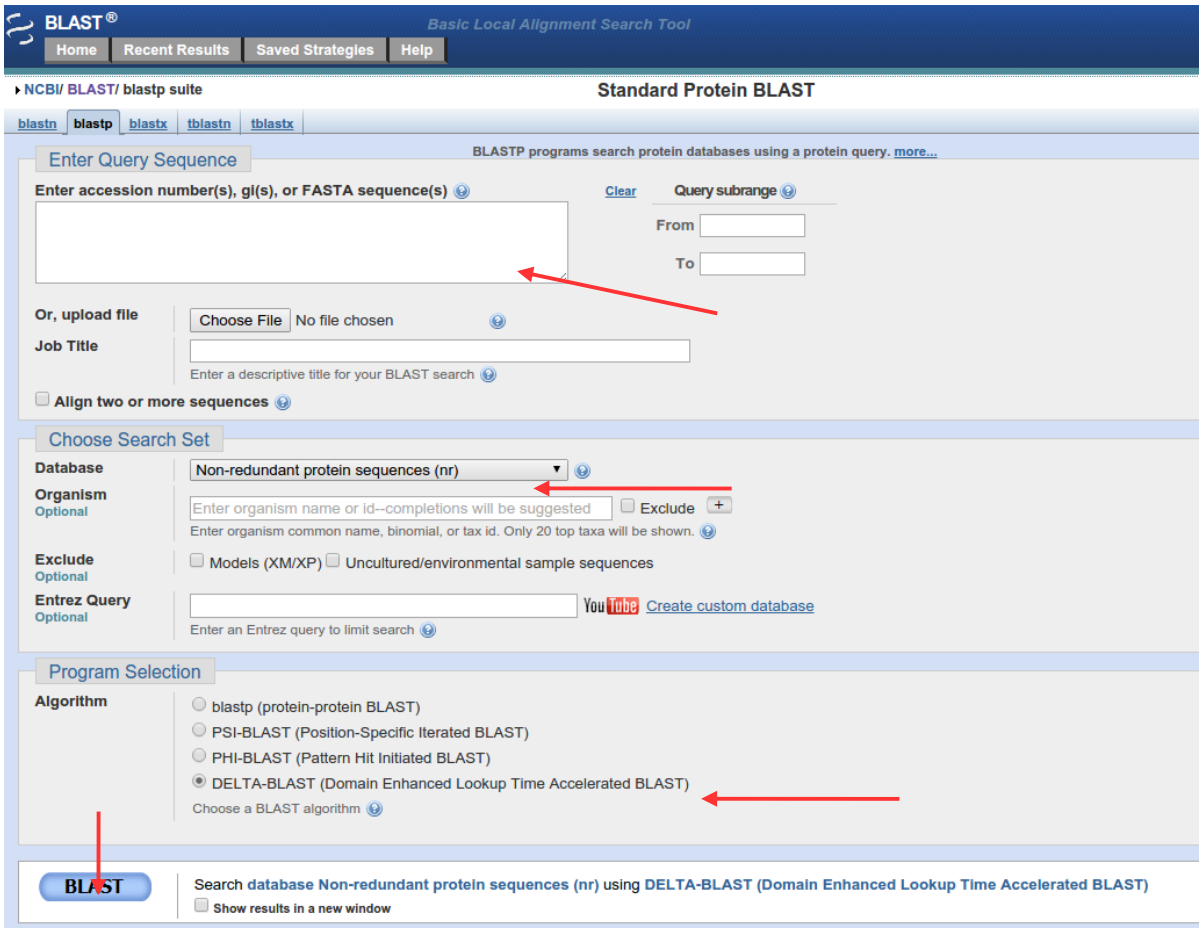
We can do a BLAST search online by exploiting the webserver:

http://blast.ncbi.nlm.nih.gov/Blast.cgi

In particular, click on the "protein blast" link to restrict the search on the protein database:



[5] See, for instance: http://en.wikipedia.org/wiki/Proteinogenic_amino_acid#Chemical_properties



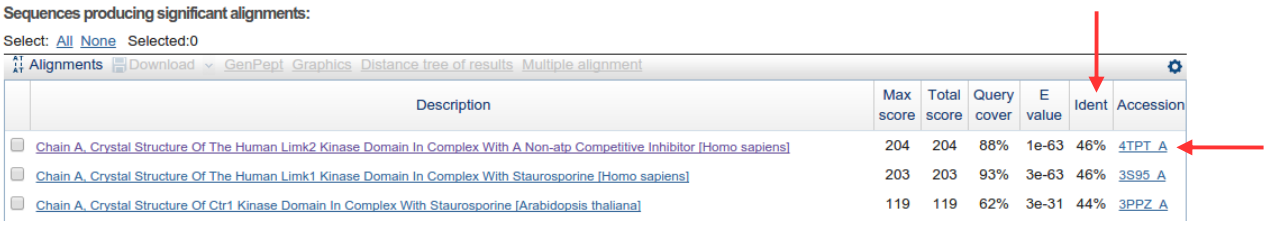
On the website, please:

1. Upload your FASTA file or copy and paste its content in the wide field at the top of the page.
2. Choose the "Protein Data Bank proteins (pdb)" as database since it is the largest protein database that contains only experimentally resolved structures (in contrast to theoretical models).
3. Choose the DELTA-BLAST algorithm.
4. Press the BLAST button to start the search.

After some seconds, the server will output the result as a list of 3D protein structures ordered according to their "sequence identity percentage" with the target sequence. The sequence similarity of each protein is summarized by the *E* value (Expected value): the closer to zero, the higher is the level of sequence similarity.

The quality of the homology model is dependent on the quality of the sequence alignment and template structure. The approach can be complicated by the presence of alignment gaps (commonly called indels) that indicate a structural region present in the target but not in the template, and by structure gaps in the template that arise from poor resolution in the experimental procedure (usually X-ray crystallography) used to solve the structure. Model quality declines with decreasing sequence identity; a typical model has ~1–2 Å root mean square deviation between the matched C^{α} atoms at 70% sequence identity but only 2–4 Å agreement at 25% sequence identity. However, the errors can be significantly higher in the loop regions, where the amino acid sequences of the target and template proteins may be completely different. As a rule of thumb, we should never use templates with an *E* value larger than 1.

In our case all the first two structures in the resulting list have a sequence identity of 46% and an *E*-value practically equal to zero, so they are reasonably good candidates.



However, since this step is crucial, several checks are mandatory before selecting the structure to be used as the template for the homology modeling:

Is it an X-ray crystallography structure? (NMR structures are usually less well resolved)

Are coordinates present for all atoms in the selected structure?

Is the chosen structure the best-resolved one (typical good resolution is ~2 Å for membrane proteins such as rhodopsin and ~1 Å for other proteins) among the structures with the same *E* value?

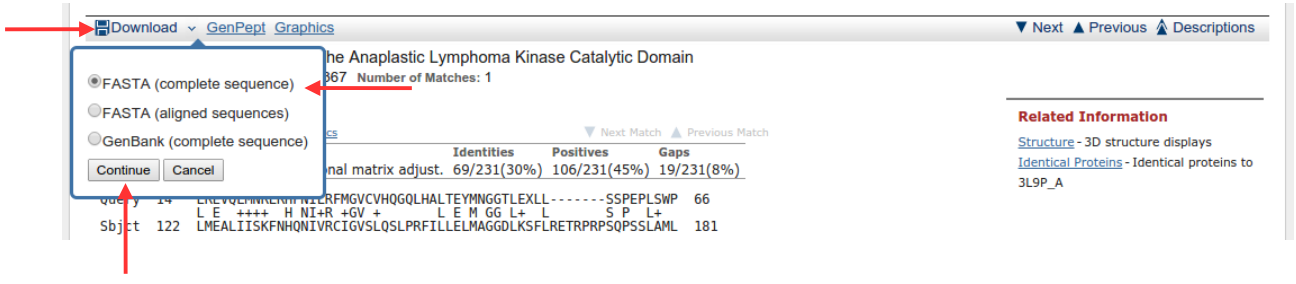
Please check the X-ray properties of the possible templates using the Protein Data Bank (PDB):

http://www.rcsb.org

You can use the PDB-ID of the listed structures for the search in the column 'Accession'. The PDB-ID (four digits or letters) is followed by an underscore and the chain that was used for the alignment. (E.g. 4TPT_A meaning PDB-ID '4PTP' and from this protein the chain A)

Checking includes also a careful inspection of each candidate structure with a visualization program such as PyMOL. Note that inside the webpage associated to any BLAST result entry you can ask for the list of all the structures corresponding to the same sequence. In this way, you can identify for each protein the best-resolved structure, etc. To do so, click on the link "Identical Proteins". However, with this procedure also results from other databases are reported. Therefore, you should take into account only the experimentally resolved 3D structures as opposed to models. Moreover, you should always read the articles associated to the structures (retrievable from the database) to understand all the conditions and the limitations related with them.

After careful inspection of the possible candidates download the sequence of the most appropriate candidate for the modeling of the 3D structure in the next step.



Save the template sequence file using a suitable filename and the file extension '.fasta' (e.g. 'template.fasta').

Why did you choose it?

Sequence alignment

To create the model (with the procedure in the next chapter) we need a sequence alignment file (.aln file) between our target and the selected template sequence. For the later modeling step with the program Modeller it is crucial to use the sequence of the template protein for residues for which atoms are resolved in the PDB file. This sequence can differ from the actual coded protein sequence as some regions of the protein (e.g. high flexible loops) cannot be resolved. To extract this sequence a script (makeModel.pl) is provided that will automatically download the PDB-file from the database and extract the sequence. For this please login on the server (honshu) and create a directory with your name (if not done so far) and switch to this directory:

```
mkdir YourName
cd YourName
```

In addition, please create a directory for each model you want to make. Here we will call the model directory "TESK1". Use the makeModel.pl script to download the PDB-file for your selected template (here e.g. 1RDQ).

```
mkdir tesk1
cd tesk1
makeModel.pl 1RDQ
```

The script will immediately start to download and parse the PDB-Entry. You will see an output like the following:

```
getting file '/home/guest/PDBs/1rdq.pdb.gz'...
File '1rdq.pdb.gz' sucessfully downloaded from
'http://www.rcsb.org/pdb/files/1rdq.pdb.gz'
gunzip -v /home/guest/PDBs/1rdq.pdb.gz...
/home/guest/PDBs/1rdq.pdb.gz: 75.6% -- replaced with
/home/guest/PDBs/1rdq.pdb
done.
/home/guest/PDBs/1rdq.pdb
Path:/home/guest/PDBs/1rdq.pdb
2758 lines written to file '1rdq_E.pdb'
153 lines written to file '1rdq_I.pdb'
2 sequences written to file 1rdq.fasta
```

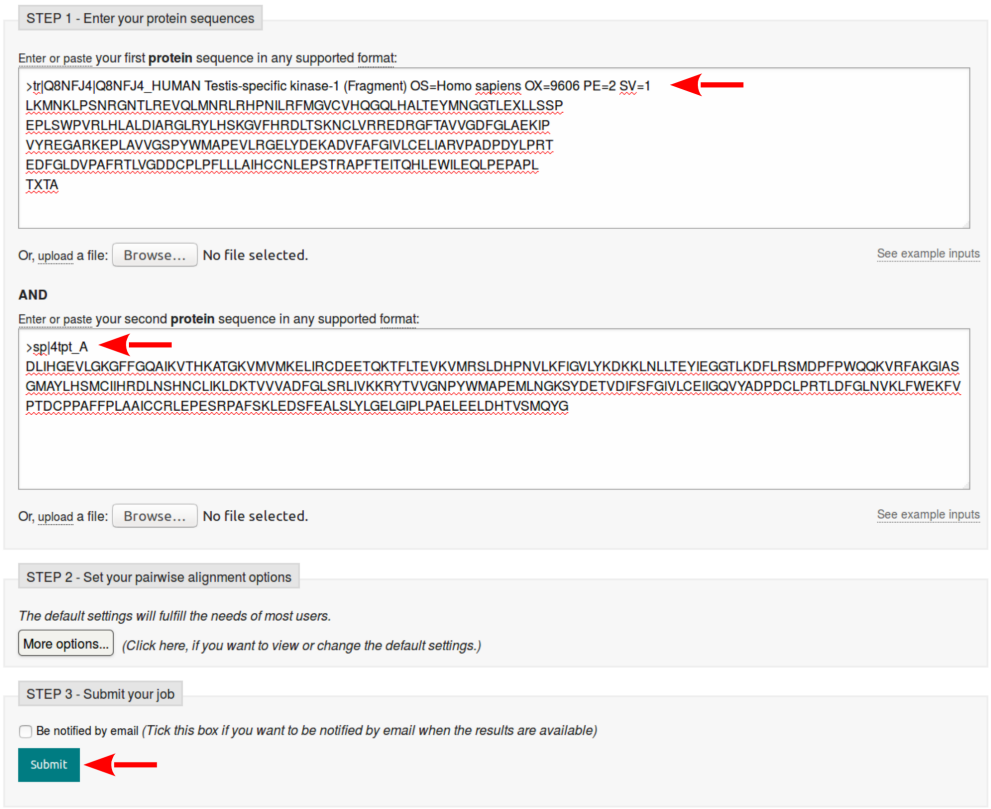
You find the sequences of each protein chain in the written FASTA (here 1rdq.fasta) file. As we want to generate a global sequence alignment between the target and the selected template sequence we need to use the Needleman-Wunsch algorithm.[6,7] To this aim, we will use another online server:

https://www.ebi.ac.uk/Tools/psa/emboss_needle

which needs the two sequences in FASTA format. Insert the two sequences one after the other in the

[6] Gotoh, O. An improved algorithm for matching biological sequences. *J. Mol. Biol.* 162, 705–708 (1982).
[7] Needleman, S. B. & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* 48, 443–53 (1970).

top box and click on the “Submit” button: all the default settings usually fulfill the needs of most queries. You can also upload the sequences as text files.



The result page will provide the sequence alignment. Download the result into an empty text file (e.g. aligment.aln) for later use.

Building the 3D model

We will use the program Modeller for the model generation. The program is freely available for academic use and can be downloaded an installed locally. For this assignment, the program is already installed on the sever (honshu) and can be called from the console directly. However, the usage of the program can be quite cumbersome and demands some knowledge of the programming language Python to operate the program via its Python-API. Therefore, we provide you with a script (same a above makeModel.pl) that will allow you to generate all necessarily input file for parallel run of Modeller on the Linux-Cluster system.

You can create a model by running the script with your alignment file (e.g. alignment.fasta) that you created in the previous step. Simply run in the console:

```
makeModel.pl alignment.aln
```

If the program finishes successfully, you will see after some output the following output indicating that a job file called MODEL.job was created for you and that this file can be used to run the actual modeling on the cluster.

```
!!!!!!!!!!!!!!!!!!!!!!
Please run the command:
qsub -N modelling MODEL.job
in this directory to start the modelling.
!!!!!!!!!!!!!!!!!!!!!!
```

You can now run your job by the specified command:

```
qsub -N modelling MODEL.job
```

This results in the job ID on the calculation cluster. E.g.:

```
1404531.honshu.bpc.uni-frankfurt.de
```

You can identify your job using this number. With the parameter "-N modelling" you can in addition specify a name for your job. (Here "modelling" was specified). Given a meaningful name (please use a more specific job name for your current task so you can distinguish your job from all other groups in the course) you can recognize your job in the calculation queue. To see the status of your job, you can use the command

```
qstat
```

resulting in the list of all jobs you submitted so far e.g.:

```
Job id                    Name             User            Time Use S Queue
------------------------- ---------------- --------------- -------- - -----
1404531.honshu            modelling        guest                  0 R batch
```

The entry in Column "S" of this table indicates the status of the job. The possibilities are **R**, **Q**, **C** indicating a **r**unning, **q**ueued or **c**ompleted job, respectively. When your job is in status **C** after a short while (around 2–3 minutes if your job runs immediately) the modeling is finished.

If you check the content of the directory you will find a large number of files starting with the prefix target... These files contain the result of the modeling e.g. the course of the minimization and the restraints that were used. The most important files for our further work are the models, in the files target.B9999000[1-48].pdb (We specified the program to calculate 48 models.)

We select the best model for the evaluation. To do this, find a file called by the job name and number (in this example, modelling.o1404531). At the end of this file, we find a table in which all models and their respective Modeller scores are listed. Additionally, another knowledge-based score called DOPE[8] (Discrete Optimized Protein) is listed. Use the script grepModellerTable.pl with the Modeller output file to generate a DOPE sorted table. E.g. here we would call the command:

```
grepModellerTable.pl modelling.o1404531
```

This results in a table that will be written automatically to a file with ".csv" extension. (It is possible to change the filename by adding a second parameter with a filename of your choice.) The script indicates this by its output

[8] Shen, M.-Y. & Sali, A. Statistical potential for assessment and prediction of protein structures. *Prot. Sci.* 15, 2507–2524 (2006).

```
table of result with 48 results written to file 'modeller.csv'
```

If you find a positive DOPE score, the modeling failed. If you find structures with nearly the same DOPE score (normally 2–5 top models in the list), it is always wise to check and validate all of these models. For the sake of simplicity here we want to use only the best model. However, consider to compare another model from the top 5 to top 10 with the evaluation tools in the following section.

Look at the Modeller publication at
http://onlinelibrary.wiley.com/doi/10.1002/0471140864.ps0209s50/abstract

Look at the Modeller website http://salilab.org/modeller/ and the documentation provided therein.

Model evaluation

Evaluation of model quality is a fundamental step in homology modeling. While the performance of Modeller has been evaluated extensively and updates are benchmarked carefully, the quality of individual models can vary significantly. Therefore, a lot of tests were developed to this aim. Some of these are also calculated automatically for the resulting structure. How to use these servers is described in an appendix of this script. Among these tests please use the following to evaluate your model.

- QMEAN[9] (http://swissmodel.expasy.org/qmean/)
- MolProbity[10] (http://molprobity.biochem.duke.edu/)

In order to be able to rank alternative models of the same target protein, pseudo energies for the entire model can be calculated by QMEAN. MolProbity is another tool for the evaluation of your models. MolProbity also analyses the side chain packing of the model and calculates some additional properties. In any case it is you are well advised to take a look at the models you created and compare them also to the template. (Are the models looking reasonable? Which parts diverge from the template and are these the same positions for the low-quality prediction?)

A number of additional models that where generated automatically by the programs I-Tasser[11], Pyre2[12], RaptorX[13], and hhpred[14] (automated single and multiple template selection) will be provided to you as the calculation on these servers can occasionally take several days.

---

[9] Benkert, P. et al. QMEAN: A comprehensive scoring function for model quality assessment. *Proteins* 71, 261–277 (2008).

[10] Chen, V. B. et al. MolProbity: all-atom structure validation for macromolecular crystallography. *Acta Cryst. Sect. D* 66, 12–21 (2009).

[11] Yang, J. et al. The I-TASSER Suite: protein structure and function prediction. *Nat. Meth*. 12, 7-8 (2015).
Ambrish, R. et al. I-TASSER: a unified platform for automated protein structure and function prediction. *Nat. Protocols.* 5, 725–738 (2010).
Zhang, Y. I-TASSER server for protein 3D structure prediction. *BMC Bioinf.* 9, 40 (2008).

[12] Kelley, L. A. & Sternberg, M. J. E. Protein structure prediction on the Web: a case study using the Phyre server. *Nat. Protocols* 4, 363–371 (2009).

[13] Källberg, M. et al. Template-based protein structure modeling using the RaptorX web server. *Nat. Protocols* 7, 1511–1522 (2012).

[14] Remmert, M. et al. HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nat. Meth.* 9, 173–175 (2012).
Söding, J. Protein homology detection by HMM–HMM comparison. *Bioinformatics* 21, 951–960 (2005).
Söding, J. et al. The HHpred interactive server for protein homology detection and structure prediction. *Nucl. Acids Res.* 33 Suppl 2, W244–W248 (2005).

Download the file automated_modelling.zip from OLAT and compare the model or models you created with the results automatically generated by these servers using the above described Methods.

How do the automated and unsupervised methods perform for this target sequence? Is the overall and local quality of the models better or worse than your model? What is the difference between the models of each server and what could be the reason for that? (Please see the publications of the automated modeling servers.)

Model evaluation with experimental data

There are many experimental methods to obtain information about the global shape, secondary structure elements, distances or local environment information of proteins. Small-angle X-ray scattering (SAXS) and low-resolution cryo-electron microscopy provide information about the global shape of the molecule. The secondary structure composition can be determined by circular dichroism spectroscopy (CD-spectroscopy) or by nuclear magnetic resonance (NMR) spectroscopy (without need for chemical shift assignment). Distance information insufficient for a structure determination can be obtained by NMR spectroscopy with partial chemical shift assignment, Förster resonance energy transfer (FRET), and electron paramagnetic resonance (EPR) spectroscopy. In addition, mutations studies in combination with functional assays can offer information about functionally and structurally important residues. They can also, in combination with the previously listed structural methods, report about structurally important residues of the target.

All of these methods have important applications of their own and are used frequently. However, they cannot provide information on an atomic level or even on the global fold of the protein. Nevertheless, if such data is available, it is crucial, in order to create a reliable model, to either include or to be able to explain experimental data. This is especially important in the frequent cases when multiple models are assessed as of comparable quality by the computational methods. The most prominent and classical experimental data used are those from mutation studies. This is due to several reasons. Mutations studies are often performed early on molecules that become of biological interest. Thus mutation data is available on nearly every target of interest. Additionally, mutation studies provide direct information about functionally relevant residues and explain e.g. why a patient with a certain mutation shows a condition, why only one pathogen strand is infectious, or even immune to a certain antibiotic. On the other hand, understanding the underlying structural reason for the effect of a mutation can not only help to understand it but also allow for successful drug development. Hence, in many cases molecular models are created specifically to explain the experimental data observed in mutation studies.

In this exercise, we want to simulate this situation and try to explain some mutations with our models. We will take a closer look at human lysozyme C, which is a very well investigated protein and was subject to thousands of studies already. Therefore, a lot is known about this molecule and naturally the structure has been known for many years and was investigated by many different methods and most mutations could be understood just by looking into the native structure. Nevertheless, we can use this protein to understand the influence of the model selection and validation based on experimental mutation data.

For this, we assume that you were only able to find three suitable templates for your target protein: 1B9O, 4CGE, 5O29. Please create three models with these templates and use the lysozyme sequence provided in the directory /home/guest/modeling_tutorial_files.

In Table 1 you find several known mutations of the target protein. In addition, you have data on the

stability change of the mutated protein from unfolding experiments of the native and mutated protein. Try to assess which of these templates are more suited for the characterization of your target using the mutation data.

Table 1: Mutations of human lysozyme C and their effect on the fold stability of the molecule

Mutation*	Effect	Experimental free energy change (kcal/mol)
C77A	less stable	-4.60
I56F	less stable	-4.09
I59G	less stable	-3.83
L8T	less stable	-3.73
A9S	neutral	-0.02
D49N	neutral	0.00
I59L	neutral	0.00
R21A	more stable	1.32
Q58G	more stable	1.87

* For instance, 'C77A' indicates that amino acid residue C (Cys) at sequence position 77 has been replaced with A (Ala).

First, please refrain from using the known lysozyme structure. Use only your three models and the method employing PyMOL described below. This should give you an impression of how the real case would look like, where you don't have any idea of the protein's fold or maybe even function. After you decided for a template and a model, you can check your model by comparing it to the known native lysozyme structure (1LZ1).

To get a feeling of what effects the mutations could have on your model, please use the mutagenesis plugin of PyMOL as described at https://PyMOLwiki.org/index.php/Mutagenesis.

Go through the rotamers as described in the plugin specification to find the best suited sidechain conformation. After that try to explain the experimental mutation data with you model to validate it.

Assessment of model druggability

The process of so-called structure-based drug design (or direct drug design) relies on knowledge of the three-dimensional structure of the biological target obtained through methods such as X-ray crystallography or NMR spectroscopy.[15] In addition the knowledge of the structure of the complex is important for the optimization of known ligands and the development of high-affinity inhibitors with different chemical composition. This can be important to avoid undesired interactions with other molecules in the cell and hence side effects and toxicity.

Lack of knowledge of 3D structures has hindered efforts to understand the binding specificities of ligands with proteins. With advances in modeling software and the growing number of known protein structures, homology modeling has become a method of choice for quickly obtaining 3D coordinates of proteins. If an experimental structure of a target is not available, it may be possible to create a homology model of the target based on the experimental structure of a related protein. In the absence

[15] Jhoti, H. & Leach, A. R. Eds. Structure-based drug discovery. Dordrecht: Springer, 2007.

of experimental data, model building on the basis of a known 3D structure of a homologous protein is at present the most reliable (but by no means perfect!) method to obtain the structural information. Knowledge of the 3D structures of proteins provides invaluable insights into the molecular basis of their functions.[16]

Regions of the model that were constructed without a template, usually by loop modeling, are generally much less accurate than the rest of the model. Errors in side chain packing and position also increase with decreasing sequence identity, and variations in these packing configurations have been suggested as a major reason for poor model quality at low sequence identity.[17] Taken together, these various atomic-position errors are significant and hinder the use of homology models for purposes that require atomic-resolution data, such as drug design and protein–protein interaction predictions.

In the following exercise, we will investigate the influence of the homology model on a potential drug discovery study. Therefore, we compare homology models of the protein dihydrofolate reductase (DHFR) generated on the basis of templates with different sequence identity.

Dihydrofolate reductase is a small enzyme that plays a supporting, but essential, role in the building of DNA and other processes. It manages the state of folate, an organic molecule that shuttles carbon atoms to enzymes that need them in their reactions. Of particular importance is the enzyme thymidylate synthase that uses these carbon atoms to build thymine bases, an essential component of DNA. After folate has released its carbon atoms, it has to be recycled. This is the job performed by dihydrofolate reductase.

Enzymes with essential roles are sensitive targets for drug therapy. Dihydrofolate reductase was the first enzyme to be targeted for cancer chemotherapy. The first drug used for cancer chemotherapy was aminopterin.[18] It binds to dihydrofolate reductase a thousand times more tightly than folate, blocking the action of the enzyme. Today, methotrexate and other variations on aminopterin are used because of their tighter binding and better clinical characteristics. Since these drugs attack a key step in the production of DNA, they tend to kill cells that are actively growing rather than cells that are not growing. Since cancer cells are often the most rapidly reproducing cells in a patient, the drug will have the strongest effect on the cancer cells. The side effects of chemotherapy, however, are the effects of the drug on other normally growing tissues, such as hair follicles and the lining of the stomach.[19]

As such an important drug target, DHFR is a well-studied protein with a great number of structures available. In this exercise, we will simulate a likely case for new drug targets where only a small number of template structures are available and try to assess the models only based on these templates. The reference structure (from which we extract the target sequence) we want to focus on human DHFR complexed with the cofactor NADPH and its native ligand folate (PDB-ID: 2W3M). In many studies, the native ligand like folate serves as the starting point for the development of new drugs. Here, we want to assume such a case and evaluate the quality of the homology model not only by the aforementioned methods, but also with respect to the interactions and its capacity to accommodate folate. To do so, we can analyze the environment of the ligand in the binding pocket of the respective model.

In many cases the structure of the target protein is not available but only a homologue from another organism. In the following, we will consider or simulate three different cases:

1. A sequence identity above 25–30% is considered to result in a structure with moderate quality.

---

[16] Vyas, V. K. et al. Homology modeling a fast tool for drug discovery: current perspectives. *Indian J. Pharm. Sci.* 74, 1 (2012).

[17] Chung, S. Y. & Subbiah, S. A structural explanation for the twilight zone of protein sequence homology. *Structure* 4, 1123–1127 (1996).

[18] Farber, S. et al., Temporary remissions in acute leukemia in children produced by folic acid antagonist, 4-aminopteroyl-glutamic acid (Aminopterin). New Engl. J. Med. 238, 787–793 (1948).

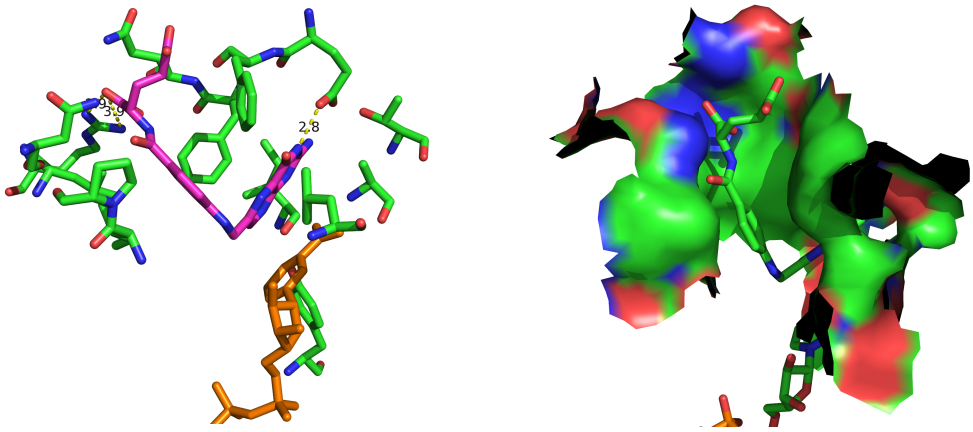
[19] http://dx.doi.org/10.2210/rcsb_pdb/mom_2002_10

Use the PDB-ID 2BL9 with a sequence identity above 30% as a moderate example.

2. The two templates 8DFR and 3D80 would be considered as good or very good templates as their sequence identity to the target is above 70%.
3. Finally, we want to take a look at an optimal case for modeling where the sequence identity is close to 100% (1YHO and 4M6J). Here, the differences in the structures could be a result of the structure determination process. E.g. 1YHO was resolved by NMR spectroscopy. In other cases, only an apo form of the protein has been crystallized or, as in the case of 4M6J, only the cofactor is bound to DHFR.

Try to model at least one structure from each of the above three groups and compare the binding pocket to our known reference. Use the servers mentioned above to create a sequence alignment (which sequence identity do you observe?) and a molecular model.

After you have obtained the models, you should analyze them with PyMOL and MolProbity.



Using PyMOL, we can first analyze the interactions of folate in the native complex conformation from 2W3M. For this purpose, hydrogen bond contacts and hydrophobic contacts can be analyzed (see left picture above). Please take also the shape of the binding pocket into consideration using the solvent accessible surface representation of PyMOL (see right picture above). You can use the following PyMOL commands to make the appropriate selections to create these kinds of representations.

To align the structures use (replace all entries enclosed in < ... > with your respective selections):

```
align <moving_structure>, 2W3M, object=align
```

By specifying an align object, you create a representation in which the residues aligned to each other are connected by lines. Additionally, this allows you to see the exact alignment in the sequence viewer, which on the other hand allows you to select aligned residues directly. Please keep in mind that the alignment of PyMOL does not necessarily have to be identical to the one you created with Clustal Omega for the modeling.

To select folate and NADPH, respectively, and extract both as objects:

```
select resn FOL
select resn NDP
```

Assuming that the two selections above have been renamed to 'ligand' and 'NADPH', respectively,

we can select the binding pocket of all models using the PyMOL command:

```
select byres sidechain within 4 of ligand
```

What do you observe? Is the binding pocket model adequate to predict the native ligand?

How are these observations correlated to the quality assessment of the server you used before?

Is there a difference between general model quality and its usability for drug design?

Do you have any idea how we can overcome these problems?

Part II: Modeling of Protein-Protein Complexes

Introduction

Protein-protein interactions are present at almost every level of cell function. Many functional proteins and molecular machines are based on the formation of transient or permanent protein-protein complexes. One reason for the generation of multimers instead of longer single-chain proteins is the possibility to form protein-complexes with different functions by modular construction of functional units. Thus, a small number of genes can be used to generate a variety of complexes with diverse characteristics. On the other hand, multimers and their formation allow for an additional regulatory step that can be controlled simply by the abundance of the monomers but also by secondary binding partners that either facilitate or prevent multimerization.

Many soluble and membrane proteins form homo-oligomeric complexes, which are responsible for the diversity and specificity of many pathways, may regulate gene expression, the activity of enzymes, ion channels, receptors, and cell adhesion processes. The evolutionary and physical mechanisms of oligomerization are very diverse and its general principles have not yet been formulated. Homo-oligomeric states may be conserved within certain protein subfamilies and be important in providing specificity to certain substrates while minimizing interactions with other unwanted partners.[20]

The p53 protein is a sequence-specific transcriptional activator that has a key function in tumor suppression.[21] Inactivation of its tumor suppressor activity, either through mutation or by association with viral or cellular proteins, contributes to the development of as many as 50% of human cancers. It is composed of four domains: an N-terminal transactivation domain, a central DNA-binding domain, an oligomerization domain, and a basic C-terminal nuclear localization domain. Although most mutations found in human cancers are located within the DNA binding domain, several observations suggest that the oligomerization domain may also play a key role in cell transformation.[22]

95% of tumorigenic p53 mutations are found in the central DNA-binding domain where they abrogate the protein fold and its sequence-specific DNA binding affinity.[23] In the early years of p53 research, little attention was given to the oligomerization domain because, in contrast to the DNA-binding domain, it is not often mutated in cancer. However, various experimental studies have shown that the tetramerization domain is essential for DNA binding, protein-protein interactions, post-translational modifications, and p53 degradation. Moreover, single point mutations in the tetramerization domain can inactivate the wild-type protein similarly to mutations in the DNA-binding domain.[24]

The determination of the positions of the atoms within a single polypeptide chain of a protein has been the primary focus of protein structure determination and of the first part of this tutorial.

However, the positioning of subunits in multi-subunit proteins is, as described above for p53, equally important, but still a challenging task for experimental methods like NMR as well as for

---

[20] Hashimoto, K. et al. Caught in self-interaction: evolutionary and functional mechanisms of protein homooligomerization. *Phys. Biol.* 8, 035007 (2011).
[21] http://www.rcsb.org/pdb/101/motm.do?momID=31
[22] Clore, G. M. et al. High-resolution structure of the oligomerization domain of p53 by multidimensional NMR. *Science* 265.5170 (1994): 386–391.
[23] Cho, Y. et al. Crystal structure of a p53 tumor suppressor-DNA complex: understanding tumorigenic mutations. *Science* 265, 346–355 (1994).
[24] Chène, P. The role of tetramerization in p53 function. *Oncogene* 20, 2611–2617 (2001).

computational prediction methods.

Here, we want to evaluate the applicability of modeling methods in the case of (homo-) dimers. In particular, we want to investigate the effect of different mutations in the oligomerization domain region of p53. We assume a scenario in which we are asked to determine which single point mutations introduced into the p53 oligomerization domain create a p53 variant that is no longer capable of forming a dimer. The structure of the p53 tetramer is known form NMR and X-ray experiments. To investigate a specific mutation, we need to perform three main steps:

1. Modeling of the mutated monomers (using Modeller)
2. Forming a rough structure of the dimer from the mutated monomers (using FRODOCK)
3. Generating a refined and minimized model of the complex (using FiberDock)

Why do you think the molecular (homology) modeling is focused on monomers or even on folding domains?

Which challenges do we face in case of protein-protein docking? Why algorithms used for docking small molecular ligands to proteins cannot be applied here directly?

Which parameters are important for the estimation of the binding affinity?

Getting to know p53

Please download the structure of the p53 tetramer from the Protein Data Bank (with PDB-ID: 1OLG):

http://www.rcsb.org

Please take your time to check the information on the PDB-page of this entry and also the information available on the following website for further information about p53:

http://www.rcsb.org/pdb/101/motm.do?momID=31

Take a look at the structure using the program PyMOL. Since this is a NMR structure, more than one model is provided to us. It is always advisable to consider all structural models and decide based on the given problem which one or which ones to use. In this tutorial, we use the simplest approach and only consider the first model, which is usually also the model best fitting to the NMR data.

Find out which chain forms a dimer with chain A. Let's call the chain forming a dimer with chain A chain *X* (in your case *X* is a chain out of B, C, D).

After detecting the dimer chain please extract the structure of the dimer using the PyMOL commands (replacing *X* with the chain you found and <*full directory path*> with the full path to the directory of the file on the Windows system, for example U:\Downloads):

```
select chain A+X
save <full directory path>\chainAX.pdb, sele
```

We will also need the sequence and structure of the monomer. Here we use the sequence of chain A employing the following PyMOL commands (you could also use the same approach as in 'Part I: Sequence alignment'):

```
select chain A
save <full directory path>\chainA.pdb, sele
save <full directory path>\p53.fasta, sele
```

Mutant modeling

To compare the binding affinity of mutated complexes, we need to generate structures of the monomers by homology modeling. To achieve comparable energies and also to check if our modeling scheme is capable to generate acceptable models of p53, we will also generate a model of p53 even though its 3D structure is already known to us. For that, we use the Modeller software as introduced in the first part of this tutorial.

We need to provide the program with the sequence of the template and the target structure. For p53 we always use the chainA sequence and structure we saved above as the template. You need to prepare an alignment in FASTA format:

```
>myModel
KKKPL...
>chainA
KKKPL...
```

The second part of this alignment beginning with chainA will be the same for all your following models. Please replace the entry 'myModel' with a reasonable name and if necessary adapt the sequence here with the mutation you want to model.

You also need to specify the template file and tell Modeler which entry contains the template file and also specify the template file (chainA.pdb). The modelling is then performed as described in 'Part I: Building the 3D-model'.

For the resulting structure, as described previously, check the structure by alignment with the known structure of chainA in PyMOL and proceed to the first docking. Please note that here we didn't need to create a sequence alignment as the alignment will always be obvious for a single point mutation.

Protein-protein docking

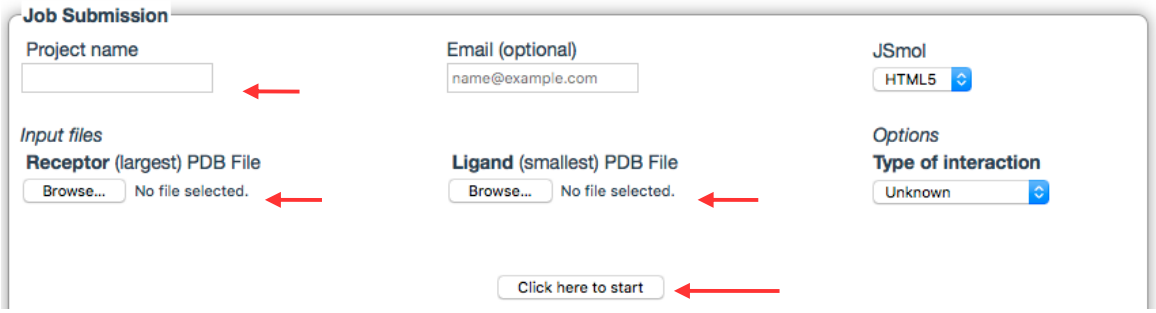
To generate a roughly oriented dimer from our generated monomer we want to use the program FRODOCK[25,26] that is provided by the web-server

[25] Garzon, J. I. et al. FRODOCK: a new approach for fast rotational protein-protein docking. *Bioinformatics* 25, 2544-2551 (2009).

[26] Ramírez-Aportela, E. et al. FRODOCK 2.0: fast protein-protein docking server. *Bioinformatics* 32, 2386–2388 (2016).

http://frodock.chaconlab.org

The usage of FRODOCK is quite simple, as we only need to specify the two p53 monomers. Choose a project name (to distinguish different docking runs) and provide FRODOCK with your monomer models both as 'Receptor' and 'Ligand'.



FRODOCK will return results after a short while. Take a look at the FRODOCK result (download and unpack solutions.tar; use ligand_1.pdb and ligand_ASA.pdb) in PyMOL.

What do you observe? What about the quality of the dimer model?

As mentioned before, this model is quite crude. It was generated mainly to optimize the global orientation of the two chains towards each other and needs to be optimized in order to calculate proper interaction energies.

For this, we use the webserver of the FiberDock[27] program:

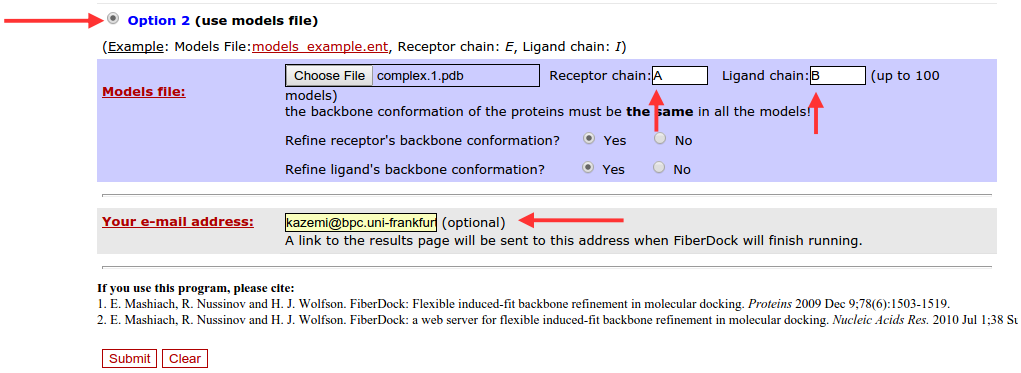
http://bioinfo3d.cs.tau.ac.il/FiberDock

The FiberDock program is capable to optimize a given dimer in more detail than FRODOCK if it is provided with an already formed dimer. We want to use the result of FRODOCK as the input of FiberDock. Unfortunately, the FRODOCK output contains a wrong chain numbering, which needs to be updated. You can achieve this with the following PyMOL commands:

```
alter ligand_1, chain='A'
alter ligand_ASA, chain='C'
save complex.pdb, ligand_1 + ligand_ASA
```

In FiberDock, select option 2 to provide the result of the docking, upload your renumbered FRODOCK result and specify the chain (which has to be A and C due to the renumbering).

[27] Mashiach E.et al. FiberDock: a web server for flexible induced-fit backbone refinement in molecular docking. *Nucl. Acids Res*. 38, W457–W461 (2010).



After a short while you will receive the results from the server. On the result page you find a summary of the energy and two links, 'download solutions table' and 'download best structures'. Following those, you can get the optimized structure and the energy estimation. You should again analyze the results and take a look at the structures.

What has changed by the optimization with FiberDock?

Predict like a boss

A couple of p53 mutations have been investigated experimentally. Please model some of the following mutations and analyze them:

L330H, L330P, L330R

R337L, R337P

L344P, L344R

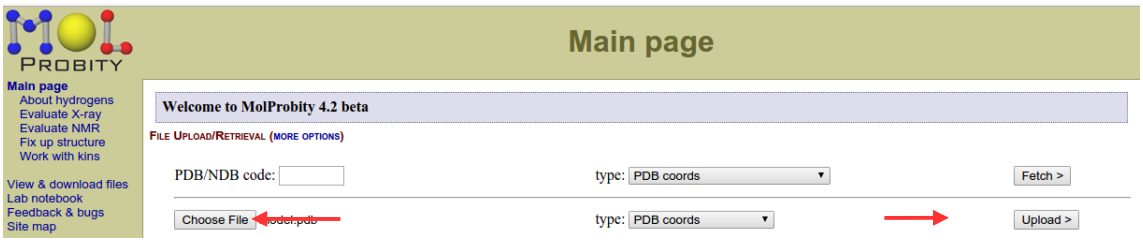
Why do you think some mutations lead to a much lower interaction energy?

Mutations involving proline have a specific influence on the local secondary structure. How could this influence dimerization (or tetramerization)?

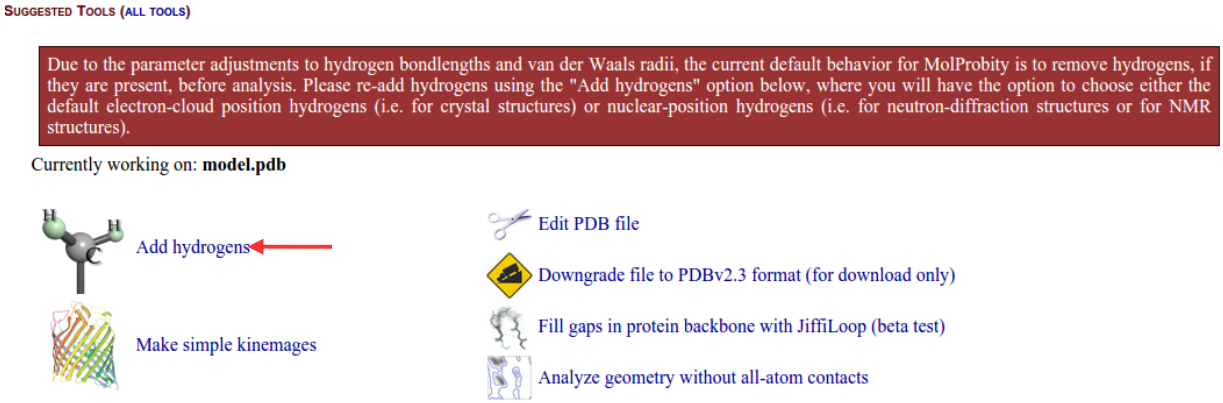
Appendix: Structure evaluation servers

MolProbity

On the website upload the PDB file of the model you want to evaluate.

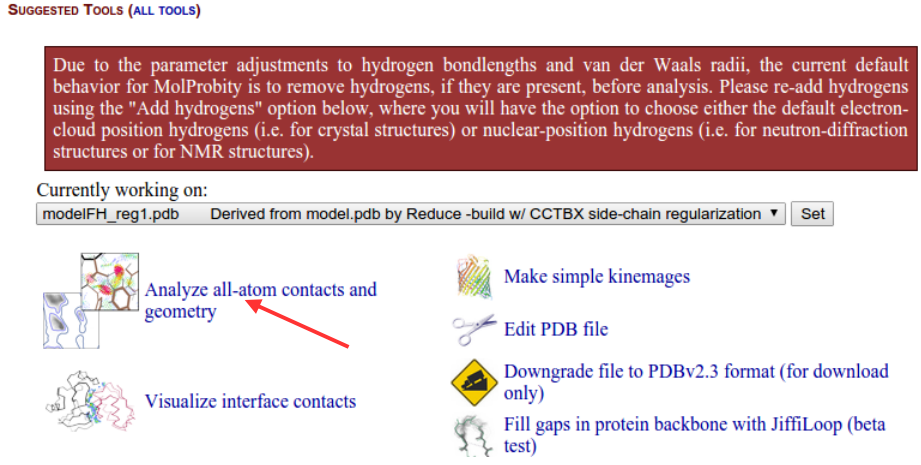


After a while, you will see a page with a summary of the structure you uploaded named 'Uploaded PDB file as XXXX.pdb'. After checking whether the uploaded file has the desired properties, click continue.



On the following page, select *add hydrogens* and *start adding H* with the default parameters (Asn/Gln/His flips). On the next page click *'Regenerate H, applying only selected flips >'* with all preselections and *continue* to the next page.

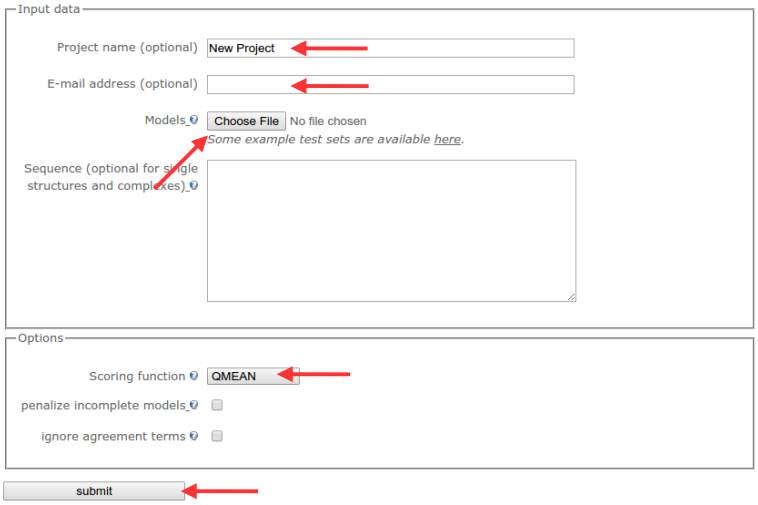
Afterwards, use the page to *analyze the all-atom contacts and geometry* and follow the instructions. Choose the analyses you want to perform (uncheck *3D-kinemage graphics* and otherwise leave the suggested defaults checked) and *run the program to perform the analyses.*



Analyze the results carefully. Check the Ramachandran plot of your model, the global scores, etc.

QMEAN

Select a project name (e.g. Q8NFJ4), enter your email address, and upload the model to be analyzed.



Depending on the load of the server the procedure will take a while. After the analysis is finished, you will receive an email with a link to an overview page of your results. It is also possible to download the complete results as an archive. These results allow you to estimate the quality of your structure and to put it into perspective in comparison to published high-quality X-ray structures.

Pseudo energies are calculated for each residue of your model. This allows you to analyze those parts of your model that can be problematic and need further care. For that, you can download a PDB-file with the estimated per residue error stored in the B-value column. This allows you to inspect the error more closely using molecular visualization programs (e.g. PyMOL). Please download the PDB-file provided and check deviations of quality over your model structure.