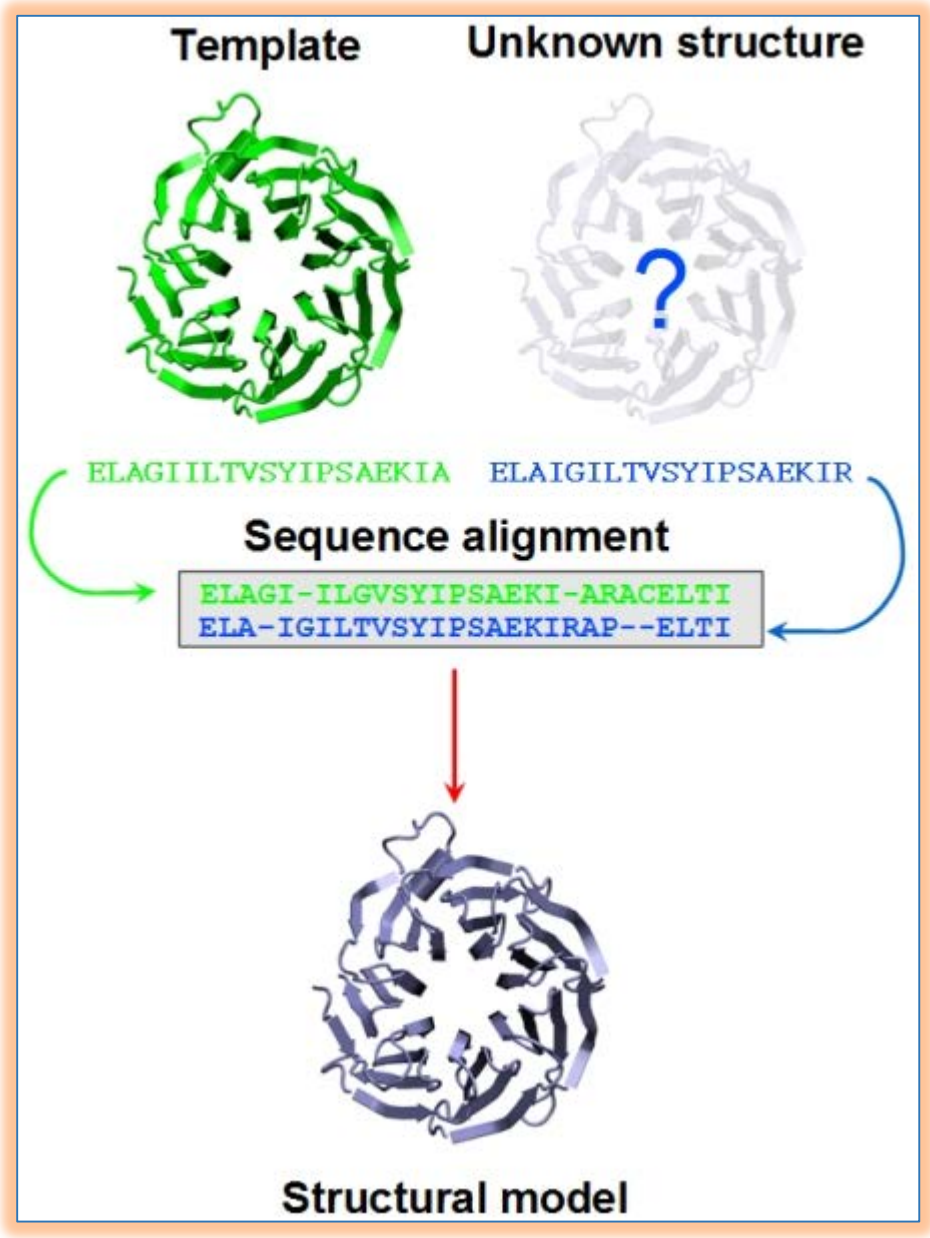
Protein Structure Prediction using HOMOLOGY MODELLING

Dr. Abdul Rajjak Shaikh

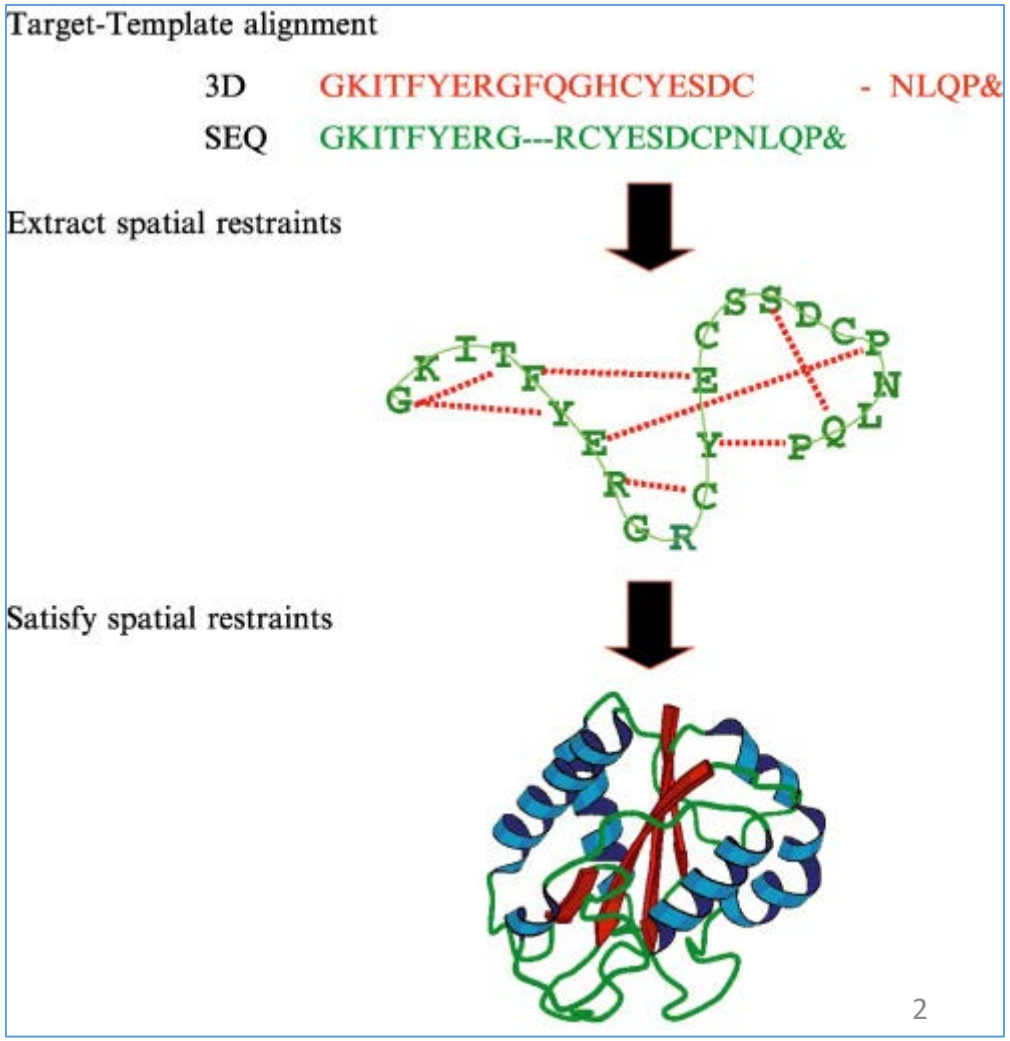
STEMskills Research and Education Lab Pvt. Ltd.



Homology Modelling

Given an unknown protein, make an informed guess on its 3D structure based on its sequence:

- Search structure databases for homologous sequences
- Transfer coordinates of known protein onto unknown



Homology Modelling using Modeller

This tutorial assumes you have installed Modeller in Windows platform.

First create a new folder/directory. As shown below we created a new folder in Documents\HOMOLOGY\MDM2

C:\Users\stemskillslab\Documents\HOMOLOGY\MDM2

First find out query sequence of homology model

Go to Universal Protein Database and search for sequence

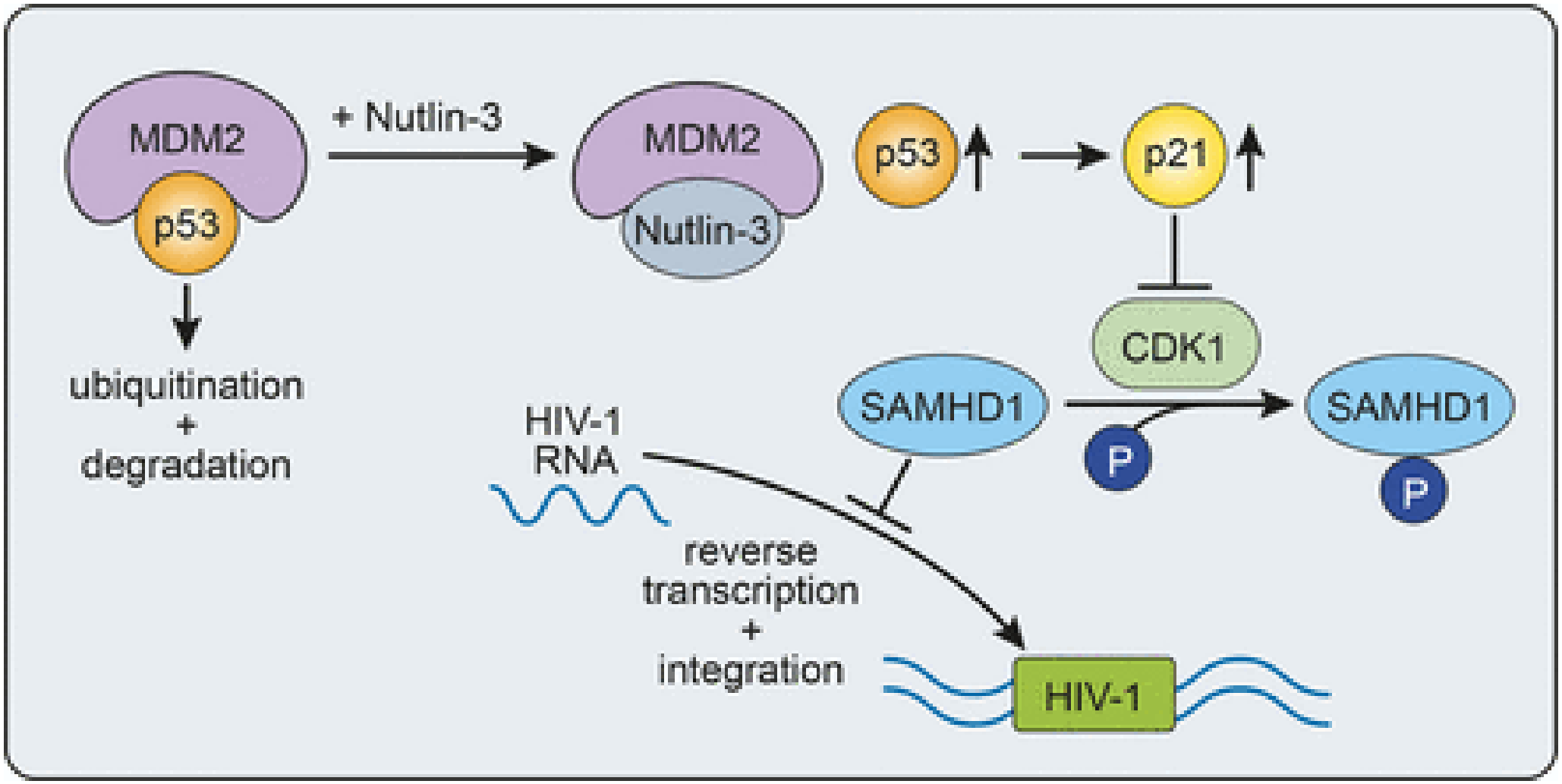
https://www.uniprot.org/

In search window type name of query sequence
For example: MDM2 and click search

For example our goal is to create a model of the MDM2 mouse protein (Mouse double minute 2), in particular of its N-terminal region that binds to the p53 trans-activation domain.

From Uniprot results, download P23804 entry for mouse MDM2. From the literature search, it is found that residues 27-110 are essential for p53 binding. Hence we can only use residues 1-110 to model our homology model for MDM2 protein.

MDM2-p53 Interaction



MDM2 fasta sequence

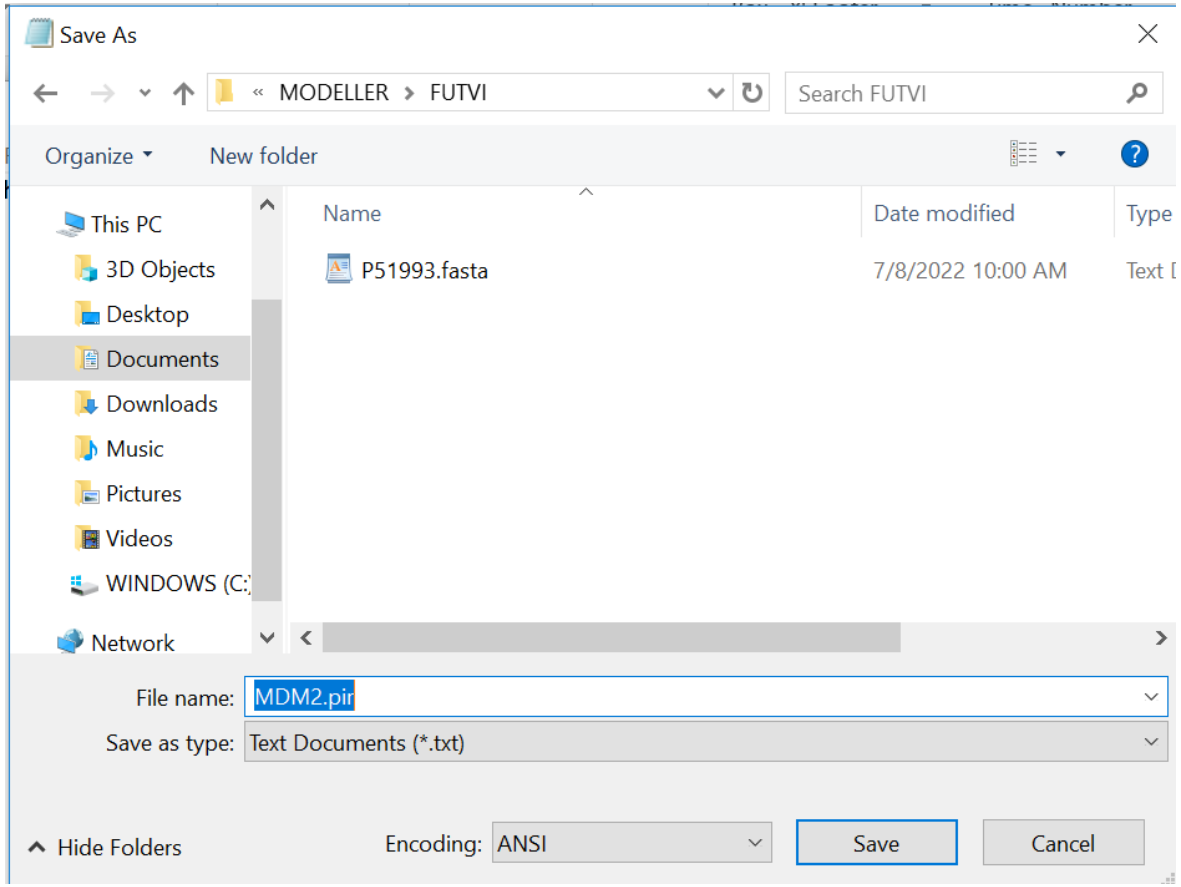
>sp|P23804|1-110
MCNTNMSVSTEGAASTSQIPASEQETLVRPKPLLLKLLKSVGAQNDTYTMKEIIFYIGQY
IMTKRLYDEKQQHIVYCSNDLLGDVFGVPSFSVKEHRKIYAMIYRNLVAV

Save the fasta sequence as MDM2_mouse.fasta in directory which we created earlier

Now, in order to run modeller, we need to convert this sequence with modeler format.

>P1;MDM2
sequence:MDM2:::::::0.00: 0.00
MCNTNMSVSTEGAASTSQIPASEQETLVRPKPLLLKLLKSVGAQNDTYTMKEIIFYIGQY
IMTKRLYDEKQQHIVYCSNDLLGDVFGVPSFSVKEHRKIYAMIYRNLVAV*

In order to make modeler alignment file, copy the fasta sequence. Open the notepad and paste the fasta sequence. Modify the part as shown in red color and then save it as MDM2.ali. Save as type “Encoding: ANSI” Save the file as MDM2.ali.



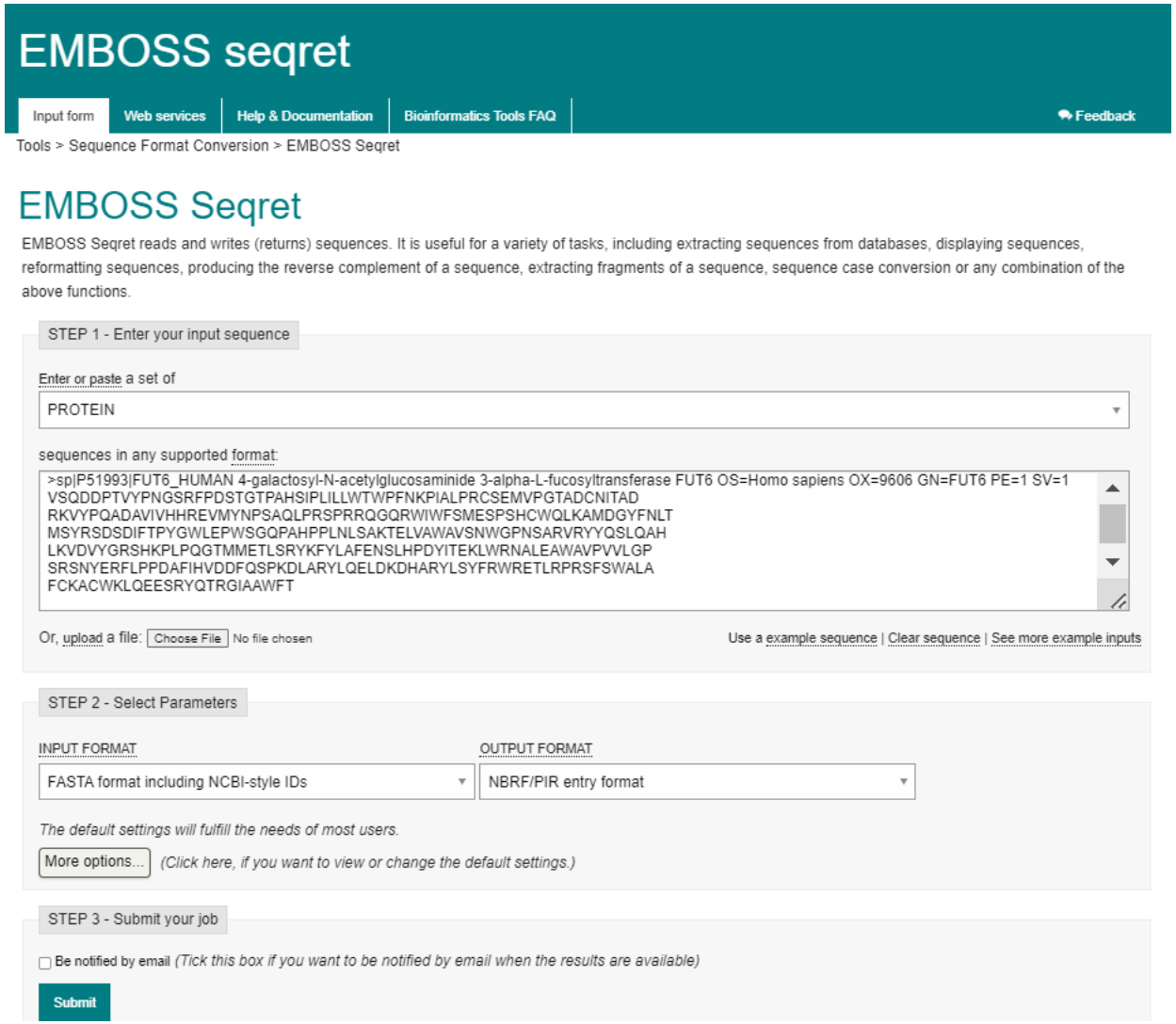
Make sure file type is ALI file.

You may also download zip file given in Basic Tutorial in Modeller website. Link is given below

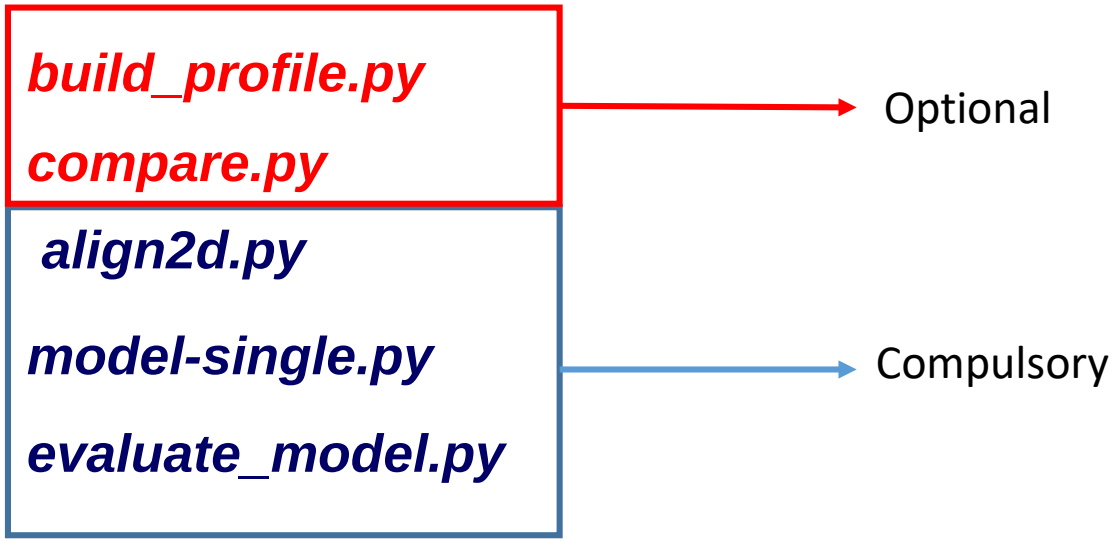
https://salilab.org/modeller/tutorial/basic-example.zip

You may use EMBOSS seqret server to convert fasta sequence to .ali format using following server

https://www.ebi.ac.uk/Tools/sfc/emboss_seqret/



From the **Basic tutorial**, we will copy python scripts and some other files.



In addition, **pdb_95.pir** also copy and paste in your directory.

pdb_95.pir is a database for existing pdb structures. However, this database is not updated. Hence we will use Protein Blast to see recent homologues structures in the Pdb database.

Once you have all the files and MDM2.ali sequence, you can run Modeller.
It is advisable to install Modeller other than in Program Files in case you get errors.

Searching for structures related to MDM2

Now open the **build1_profile.py** in wordpad

Change the TvLDH.ali filename with your .ali filename

Ex. TvLDH.ali to MDM2.ali

optional

```
from modeller import *

log.verbose()
env = environ()

#-- Prepare the input files

#-- Read in the sequence database
sdb = sequence_db(env)
sdb.read(seq_database_file='pdb_95.pir', seq_database_format='PIR',
         chains_list='ALL', minmax_db_seq_len=(30, 4000), clean_sequences=True)

#-- Write the sequence database in binary form
sdb.write(seq_database_file='pdb_95.bin', seq_database_format='BINARY',
          chains_list='ALL')

#-- Now, read in the binary database
sdb.read(seq_database_file='pdb_95.bin', seq_database_format='BINARY',
         chains_list='ALL')

#-- Read in the target sequence/alignment
aln = alignment(env)
aln.append(file='TvLDH.ali', alignment_format='PIR', align_codes='ALL')

#-- Convert the input sequence/alignment into
#   profile format
prf = aln.to_profile()

#-- Scan sequence database to pick up homologous sequences
prf.build(sdb, matrix_offset=-450, rr_file='${LIB}/blosum62.sim.mat',
          gap_penalties_1d=(-500, -50), n_prof_iterations=1,
          check_profile=False, max_aln_evalue=0.01)

#-- Write out the profile in text format
prf.write(file='build_profile.prf', profile_format='TEXT')

#-- Convert the profile back to alignment format
aln = prf.to_alignment()

#-- Write out the alignment file
aln.write(file='build_profile.ali', alignment_format='PIR')
```

Now we are ready to run Modeller. Find out the directory where you have installed Modeller. In our case it is: C:\Users\stemskillslab\MODELLER\Modeller9.25 (change 9.25 with Modeller version which you installed)

Now copy MDM2 directory which we created earlier and paste it in bin folder. Now path of directoy is as given below

C:\Users\stemskillslab\MODELLER\Modeller9.25\bin\MDM2

Now go to windows search and type Modeller and run the Modeller App (you may run as Adminitrator if required).

In the Modeller window, type following command

cd bin [press enter command]
Cd MDM2 [enter]
Now to run build1_profile, type python build1_profile.py (press enter). If python does not run type mod9.25.

In Modeller window, it will look like this

C:\Users\stemskillslab\MODELLER\Modeller9.25\bin\MDM2>*python build1_profile.py > build1_profile.log*

This command if does not show any error, then it will create following outputs in bin/MDM2 directory
build1_profile.log
build_profile.prf
build_profile.ali

optional

Now open the build_profile.prf in wordpad, and see the output

```
# Number of sequences:      3
# Length of profile  :    110
# N_PROF_ITERATIONS  :      1
# GAP_PENALTIES_1D   :   -500.0   -50.0
# MATRIX_OFFSET      : -450.0
# RR_FILE            : ${LIB}/blosum62.sim.mat
    1 MDM2                                    S     0   110     1   110     0     0     0    0.    0.0
MCNTNMSVSIEGAASTSQIPASEQETLVRPKPLLLKLLKSVGAQNDTYTMKEIIFYIGQYIMTKRLYDEKQQHIVYCSNDLLGDVFGVPSFSVKEHRKIYAMIYRNLVAV
    2 1t4fM                                   X     1    88    23   110     1    88    88   92.    0.0     --------------------
--EQETLVRPKPLLLKLLKSVGAQKDTYTMKEVLFYLGQYIMTKRLYDEKQQHIVYCSNDLLGDLFGVPSFSVKEHRKIYTMIYRNLVVV
    3 1ycqA                                   X     1    88    25   109     1    85    85   71.    0.0     --------------------
----EKLVQPTPLLLSLLKSAGAQKETFTMKEVIYHLGQYIMAKQLYDEKQQHIVHCSNDPLGELFGVQEFSVKEPRRLYAMISRNLVS-
```

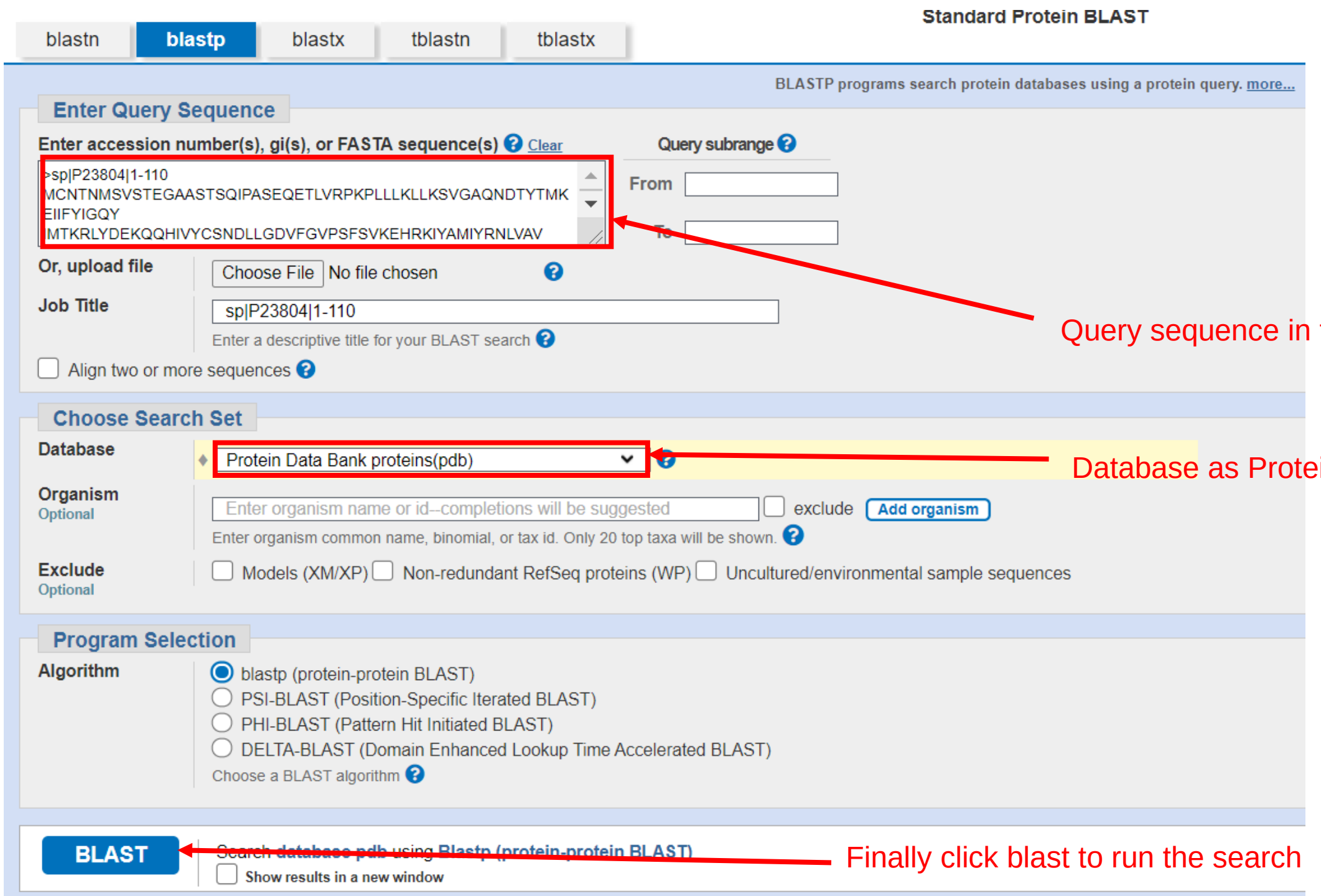
optional

Based on search for template, Modeller predicted that 1T4F-M and 1YCQ-A as a template for homology model. M and A represents chain name.

We will verify this information using Protein Blast

Compulsory

https://blast.ncbi.nlm.nih.gov/Blast.cgi?PAGE=Proteins


Standard Protein BLAST
blastn
blastp
blastx
tblastn
tblastx
BLASTP programs search protein databases using a protein query. more...
Enter Query Sequence
Enter accession number(s), gi(s), or FASTA sequence(s) Clear
>sp|P23804|1-110
MCNTNMSVSTEGAASTSQIPASEQETLVRPKPLLLKLLKSVGAQNDTYTMK
EIIFYIGQY
MTKRLYDEKQQHIVYCSNDLLGDVFGVPSFSVKEHRKIYAMIYRNLVAV
Query subrange
From
To
Or, upload file
Choose File
No file chosen
Job Title
sp|P23804|1-110
Enter a descriptive title for your BLAST search
Align two or more sequences
Choose Search Set
Database
Protein Data Bank proteins(pdb)
Organism
Optional
Enter organism name or id--completions will be suggested
exclude
Add organism
Enter organism common name, binomial, or tax id. Only 20 top taxa will be shown.
Exclude
Optional
Models (XM/XP)
Non-redundant RefSeq proteins (WP)
Uncultured/environmental sample sequences
Program Selection
Algorithm
blastp (protein-protein BLAST)
PSI-BLAST (Position-Specific Iterated BLAST)
PHI-BLAST (Pattern Hit Initiated BLAST)
DELTA-BLAST (Domain Enhanced Lookup Time Accelerated BLAST)
Choose a BLAST algorithm
BLAST
Search database pdb using Blastp (protein-protein BLAST)
Show results in a new window
Query sequence in fasta format
Database as Protein Data Bank
Finally click blast to run the search

Sequences producing significant alignments

Download | New Select columns | Show 100

select all *65 sequences selected* | GenPept | Graphics | Distance tree of results | Multiple alignment | New MSA Viewer

Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
NMR structure of unliganded MDM2 [Homo sapiens]	Homo sapiens	191	191	100%	4e-64	90.00%	119	1Z1M_A
NMR Structure of Mdm2 (6-125) with Pip-1 [Homo sapiens]	Homo sapiens	191	191	100%	5e-64	90.00%	125	2LZG_A
Crystal structure of p53 epitope-scaffold based on a inhibitor of cysteine proteases in complex with human MDM2 [Hom...	Homo sapiens	190	190	100%	2e-63	90.00%	153	5SWK_A
Structure of complex of MDM2(3-109) and P73 TAD(10-25) [Homo sapiens]	Homo sapiens	183	183	96%	3e-61	90.57%	107	2MPS_A
Ordering of the N Terminus of Human MDM2 by Small Molecule Inhibitors [Homo sapiens]	Homo sapiens	179	179	95%	2e-59	89.52%	120	4HBM_A
Co-Crystal Structure of MDM2 with Inhibitor Compound 4 [Homo sapiens]	Homo sapiens	178	178	95%	3e-59	89.52%	105	4ODE_A
Green fluorescent protein linked MTide-02 inhibitor in complex with mdm2 [Homo sapiens]	Homo sapiens	178	178	95%	4e-59	89.52%	122	5WTS_B
Structure of a stapled peptide antagonist bound to Nutlin-resistant Mdm2 [Homo sapiens]	Homo sapiens	176	176	95%	2e-58	88.57%	120	4UMN_A
Structure-activity studies of Mdm2/Mdm4-binding stapled peptides comprising non-natural amino acids [Homo sapiens]	Homo sapiens	176	176	95%	4e-58	88.57%	122	5XXK_A
MDM2 in complex with SAR405838 [Homo sapiens]	Homo sapiens	174	174	91%	1e-57	90.10%	109	5TRF_A
Chemical Shift Assignments for MIP and MDM2 in bound state [Homo sapiens]	Homo sapiens	165	165	90%	9e-54	87.88%	131	2RUH_A
Structure of the stapled peptide YS-02 bound to MDM2 [Homo sapiens]	Homo sapiens	162	162	90%	8e-53	86.87%	114	4UD7_A
Crystal Structure of HdmX bound to the p53-peptidomimetic Ac-Phe-Met-Aib-Pmp-Trp-Glu-Ac3c-Leu-NH2 at 1.35A [Ho...	Homo sapiens	85.5	85.5	90%	1e-22	49.49%	100	3FE7_A
Solution structure of Hdm2 with engineered cyclotide [Homo sapiens]	Homo sapiens	164	164	89%	2e-53	88.78%	129	2M86_B
Crystal Structure of an MDM2/Nutlin-3a complex [Homo sapiens]	Homo sapiens	159	159	88%	4e-52	87.63%	97	4HG7_A
The Central Valine Concept Provides an Entry in a New Class of Non Peptide Inhibitors of the P53-MDM2 Interaction [...	Homo sapiens	184	184	85%	8e-62	91.49%	96	4DIJ_A
Structure of human MDM2 in complex with an optimized p53 peptide [Homo sapiens]	Homo sapiens	165	165	85%	5e-54	92.55%	110	1T4F_M

Ignore NMR structure as a template

5SWK, 4HBM and 4ODE are good template with ~90% sequence identity

Download 5SWK, 4HBM and 4ODE crystal structure from protein data bank and save it in MDM2 folder

https://www.rcsb.org/

Selecting a template

Now, open compare.py in WordPad and modify the pdb id and chain using previous information

```
from modeller import *

env = environ()
aln = alignment(env)
for (pdb, chain) in (('1b8p', 'A'), ('1bdm', 'A'), ('1civ', 'A'),
                     ('5mdh', 'A'), ('7mdh', 'A'), ('1smk', 'A')):
    m = model(env, file=pdb, model_segment=('FIRST:'+chain, 'LAST:'+chain))
    aln.append_model(m, atom_files=pdb, align_codes=pdb+chain)
aln.malign()
aln.malign3d()
aln.compare_structures()
aln.id_table(matrix_file='family.mat')
env.dendrogram(matrix_file='family.mat', cluster_cut=-1.0)
```

```
from modeller import *

env = environ()
aln = alignment(env)
for (pdb, chain) in (('5swk', 'A'), ('4hbm', 'A'), ('4ode', 'A')):
    m = model(env, file=pdb, model_segment=('FIRST:'+chain, 'LAST:'+chain))
    aln.append_model(m, atom_files=pdb, align_codes=pdb+chain)
aln.malign()
aln.malign3d()
aln.compare_structures()
aln.id_table(matrix_file='family.mat')
env.dendrogram(matrix_file='family.mat', cluster_cut=-1.0)
```

Selecting a template

Now, open compare.py in WordPad and modify the pdb id and chain using previous information

```
from modeller import *

env = environ()
aln = alignment(env)
for (pdb, chain) in (('1b8p', 'A'), ('1bdm', 'A'), ('1civ', 'A'),
                     ('5mdh', 'A'), ('7mdh', 'A'), ('1smk', 'A')):
    m = model(env, file=pdb, model_segment=('FIRST:'+chain, 'LAST:'+chain))
    aln.append_model(m, atom_files=pdb, align_codes=pdb+chain)
aln.malign()
aln.malign3d()
aln.compare_structures()
aln.id_table(matrix_file='family.mat')
env.dendrogram(matrix_file='family.mat', cluster_cut=-1.0)
```

```
from modeller import *

env = environ()
aln = alignment(env)
for (pdb, chain) in (('5swk', 'A'), ('4hbm', 'A'), ('4ode', 'A')):
    m = model(env, file=pdb, model_segment=('FIRST:'+chain, 'LAST:'+chain))
    aln.append_model(m, atom_files=pdb, align_codes=pdb+chain)
aln.malign()
aln.malign3d()
aln.compare_structures()
aln.id_table(matrix_file='family.mat')
env.dendrogram(matrix_file='family.mat', cluster_cut=-1.0)
```

After modifying compare.py, run it using Modeller

```
C:\Users\stemskillslab\MODELLER\Modeller9.25\bin\MDM2>python compare.py
```

After running the command, check if program shows some errors. If not, script run successfully. Open the compare.log file and scroll down

```
Sequence identity comparison (ID_TABLE):

   Diagonal        ... number of residues;
   Upper triangle ... number of identical residues;
   Lower triangle ... % sequence identity, id/min(length).

          temp1A@1temp2A@1temp3A@1
temp1A@1       84      81      81
temp2A@1       96     106      95
temp3A@1       96      90     105


Weighted pair-group average clustering based on a distance matrix:


                                                                  .--
- temp1A@1.9     4.0000                                           |
                                                                  |
         .-------------------------------------------------------------
- temp2A@1.9     7.0000
         |
      .-------------------------------------------------------------
- temp3A@1.8


+----+----+----+----+----+----+----+----+----+----+----+----+
     7.1200    6.5800    6.0400    5.5000    4.9600    4.4200
3.8800
          6.8500    6.3100    5.7700    5.2300    4.6900    4.1500
```

We will select 4ode to build our homology model. It has highest resolution and structure is available from resid 6-110 for 4ODE (4ode). Temp1 (5SWK) has structure from 27-110

Aligning MDM2 with the template

Now open align2d.py and modify it

```
from modeller import *

env = environ()
aln = alignment(env)
mdl = model(env, file='1bdm', model_segment=('FIRST:A','LAST:A'))
aln.append_model(mdl, align_codes='1bdmA', atom_files='1bdm.pdb')
aln.append(file='TvLDH.ali', align_codes='TvLDH')
aln.align2d()
aln.write(file='TvLDH-1bdmA.ali', alignment_format='PIR')
aln.write(file='TvLDH-1bdmA.pap', alignment_format='PAP')
```

```
from modeller import *

env = environ()
aln = alignment(env)
mdl = model(env, file='4ode', model_segment=('FIRST:A','LAST:A'))
aln.append_model(mdl, align_codes='4odeA', atom_files='4ode.pdb')
aln.append(file='MDM2.ali', align_codes='MDM2')
aln.align2d()
aln.write(file='MDM2-4odeA.ali', alignment_format='PIR')
aln.write(file=' MDM2-4odeA.pap', alignment_format='PAP')
```

After modifying align2d.py, run the Modeller

C:\Users\stemskillslab\MODELLER\Modeller9.25\bin\MDM2>*python align2d.py (press Enter]*

After successful completion of job, two output files will be created.

MDM2-4odeA.ali
MDM2-4odeA.pap

```
>P1;4odeA
structureX:4ode.pdb:   6 :A:+105 :A:MOL_ID  1; MOLECULE  E3 UBIQUITIN-PROTEIN LIGASE MDM2; CHAIN  A; ESCHERICHIA COLI;
EXPRESSION_SYSTEM_TAXID  562: 1.80: 0.20
M-----SVPTDGAVTTSQIPASEQETLVRPKPLLLKLLKSVGAQKDTYTMKEVLFYLGQYIMTKRLYDEKQQHIV
YCSNDLLGDLFGVPSFSVKEHRKIYTMIYRNLVVV*

>P1;MDM2
sequence:MDM2:     : :     : :::-1.00:-1.00
MCNTNMSVSTEGAASTSQIPASEQETLVRPKPLLLKLLKSVGAQNDTYTMKEIIFYIGQYIMTKRLYDEKQQHIV
YCSNDLLGDVFGVPSFSVKEHRKIYAMIYRNLVAV*
```

MDM2-4odeA.ali file give alignment as shown above. Bold sequence means same sequence in template and query sequence, red color indicates similar sequence.

MDM2-4odeA.pap

```
 _aln.pos        10        20        30        40        50        60
4odeA    M-----SVPTDGAVTTSQIPASEQETLVRPKPLLLKLLKSVGAQKDTYTMKEVLFYLGQYIMTKRLYD
MDM2       MCNTNMSVSTEGAASTSQIPASEQETLVRPKPLLLKLLKSVGAQNDTYTMKEIIFYIGQYIMTKRLYD
 _consrvd *     ** * **  ****************************** *******  ** ***********

 _aln.p   70        80        90       100       110
4odeA    EKQQHIVYCSNDLLGDLFGVPSFSVKEHRKIYTMIYRNLVVV
MDM2       EKQQHIVYCSNDLLGDVFGVPSFSVKEHRKIYAMIYRNLVAV
 _consrvd ***************** *************** ******* *
```

Model Building (model-single.py)

```
from modeller import *
from modeller.automodel import *
#from modeller import soap_protein_od

env = environ()
a = automodel(env, alnfile='TvLDH-1bdmA.ali',
              knowns='1bdmA', sequence='TvLDH',
              assess_methods=(assess.DOPE,
                              #soap_protein_od.Scorer(),
                              assess.GA341))
a.starting_model = 1
a.ending_model = 5
a.make()
```

```
from modeller import *
from modeller.automodel import *
#from modeller import soap_protein_od

env = environ()
a = automodel(env, alnfile='MDM2-4odeA.ali',
              knowns='4odeA', sequence='MDM2',
              assess_methods=(assess.DOPE,
                              #soap_protein_od.Scorer(),
                              assess.GA341))
a.starting_model = 1
a.ending_model = 5
a.make()
```

```
C:\Users\stemskillslab\MODELLER\Modeller9.25\bin\MDM2>python model-single.py
```

model-single.py will generate five different models such as
MDM2.B99990001.pdb to MDM2.B99990005.pdb

Open model-single.log to see the summary of results

```
>> Summary of successfully produced models:
Filename                    molpdf     DOPE score    GA341 score
------------------------------------------------------------------
MDM2.B99990001.pdb          702.97705  -11947.25488      1.00000
MDM2.B99990002.pdb          675.47394  -12101.97168      1.00000
MDM2.B99990003.pdb          687.27246  -11985.36133      1.00000
MDM2.B99990004.pdb          603.79675  -11894.02539      1.00000
MDM2.B99990005.pdb          711.04773  -12066.50000      1.00000
```

MDM2.B99990002.pdb shows lowest DOPE score. Hence we will select model 2 to refine using evaluate_model.py

you could pick the model with the lowest value of the MODELLER objective function or the DOPE or SOAP assessment scores, or with the highest GA341 assessment score, which are reported at the end of the log file, above.

Model Evaluation (evaluate_model.py)

```
from modeller import *
from modeller.scripts import complete_pdb

log.verbose()    # request verbose output
env = environ()
env.libs.topology.read(file='$(LIB)/top_heav.lib') # read topology
env.libs.parameters.read(file='$(LIB)/par.lib') # read parameters

# read model file
mdl = complete_pdb(env, 'MDM2.B99990002.pdb')

# Assess with DOPE:
s = selection(mdl)   # all atom selection
s.assess_dope(output='ENERGY_PROFILE NO_REPORT', file='MDM2.profile',
              normalize_profile=True, smoothing_window=15)
```

Change the name of file which is having lowest DPOE score

```
C:\Users\stemskillslab\MODELLER\Modeller9.25\bin\MDM2>python evaluate_model.py
```

It will generate MDM2.profile and evaluate_model.log output.

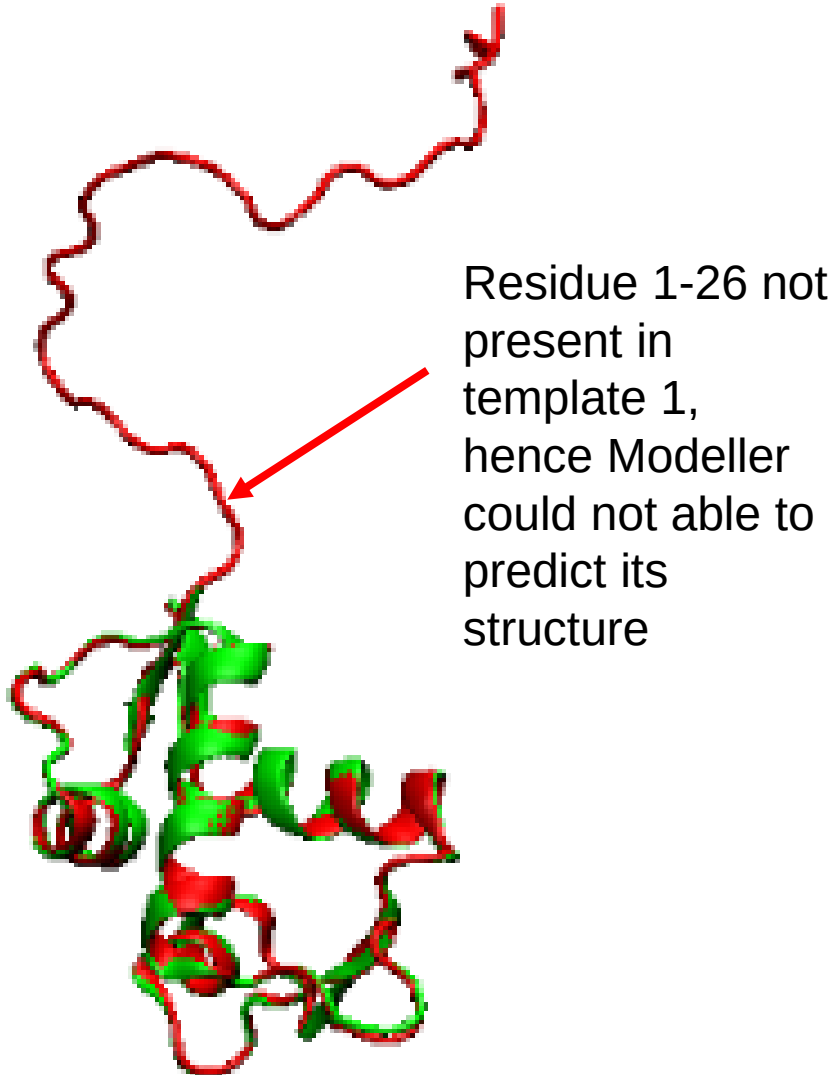
```
DOPE score               : -12101.677734
```

Here you can see that after model refinement, DOPE score is more negative value

Selection of good template is essential

MDM2 Homology Model using template 3 (4ODE)

4ode (4ODE)
MDM2 model

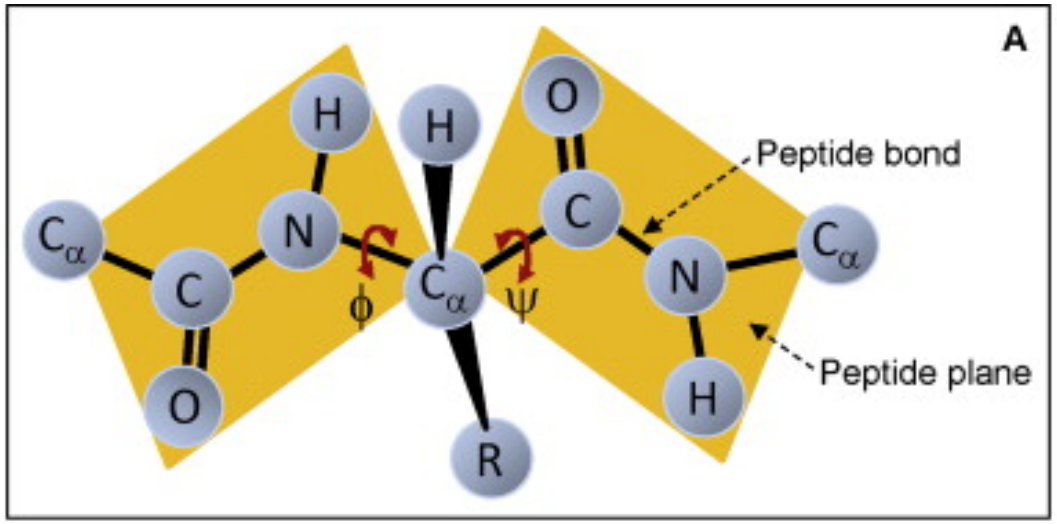


MDM2 Homology Model using template 1 (5SWK)

Temp1 (5SWK)
MDM2 model

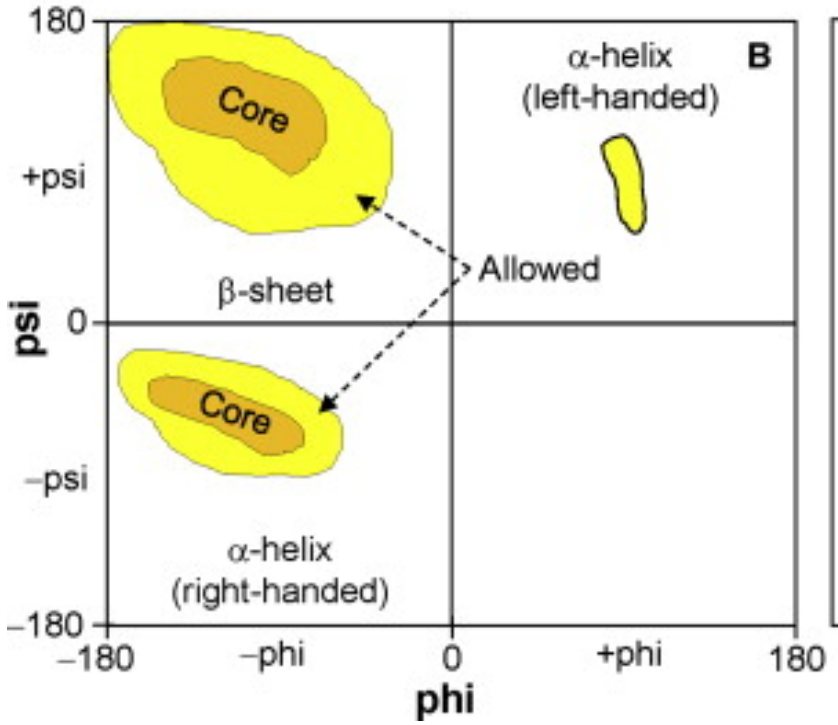
Model Evaluation

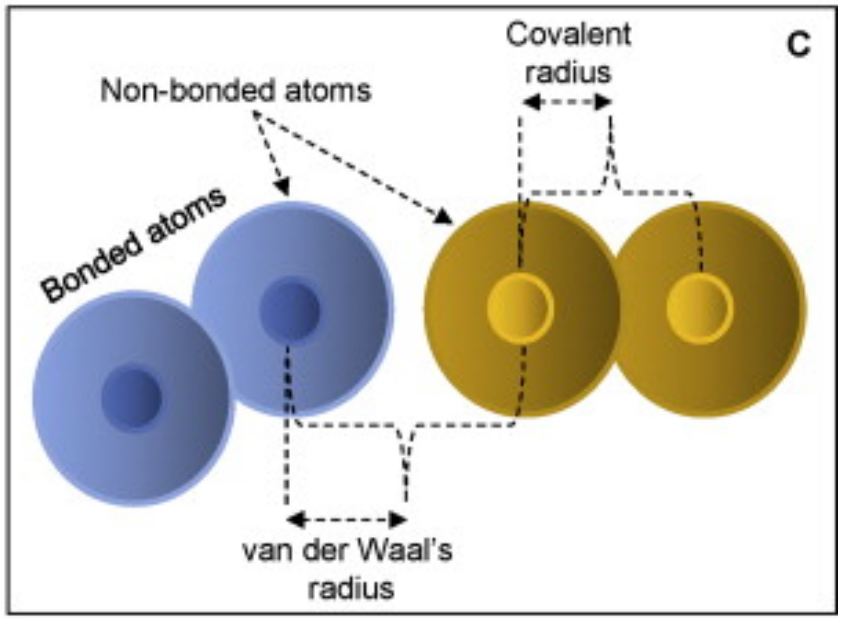
Ramachandran Plot



ϕ = C(i-1),N(i),Cα(i),C(i)

ψ = N(i),Cα(i),C(i),N(i+1)





Discrete Optimized Protein Energy (DOPE Score)

Using the probability theory, we derive an atomic distance-dependent statistical potential from a sample of native structures that does not depend on any adjustable parameters (Discrete Optimized Protein Energy, or DOPE). It is grounded entirely in the probability theory.

$$p(\vec{x}_1, \vec{x}_2, \vec{x}_3, \ldots, \vec{x}_N | I)$$

here N is the number of atoms in the protein and χ_i are the Cartesian coordinates of atom i.

In general, information I may include the sequence of the protein, a molecular mechanics force field, experimental structural information, a sample of known native structures, and an alignment of the sequence to a related known protein structure.

the joint probability density function (pdf) p can be approximated by a normalized product of the pair pdfs for all protein atom pairs:

$$p(\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N) \approx \prod_{i \neq j}^{N} p(\vec{x}_i, \vec{x}_j) / \left(\prod_{i}^{N} p(\vec{x}_i)\right)^{N-2} \propto \prod_{i \neq j}^{N} p(\vec{x}_i, \vec{x}_j) \quad (1)$$

Protein Sci. 2006 Nov; 15(11): 2507–2524

GA341 score

The GA341 score depends on three variables: the percentage sequence identity is calculated from the alignment that was used to build the model, while the compactness (Methods) and the combined statistical potential z-score (Melo et al. 2002) are calculated from the 3D model itself.

Melo *et al.*, 2002
Melo, F., Sánchez, R., & Šali, A. (2002).
Protein Sci. **11**, 430-448.