

Comparative Protein Structure Prediction

MODELLER tutorial

```
$>mod9v7 model.py
```

Marc A. Marti-Renom

<http://bioinfo.cipf.es/sgu/>

Structural Genomics Unit
Bioinformatics Department

Prince Felipe Research Center (CIPF), Valencia, Spain



Obtaining **MODELLER** and related information

- ◆ MODELLER (9v7) web page
- ◆ <http://www.salilab.org/modeller/>
 - ◆ Download Software (Linux/Windows/Mac/Solaris)
 - ◆ HTML Manual
 - ◆ **Join Mailing List**



Using MODELLER

- ◆ No GUI! 😞
- ◆ Controlled by command file 😞😞
- ◆ Script is written in PYTHON language 😊
- ◆ You may know Python language is simple 😊😊

Using MODELLER

◆ INPUT:

- ◆ Target Sequence (FASTA/PIR format)
- ◆ Template Structure (PDB format)
- ◆ Python file

◆ OUTPUT:

- ◆ Target-Template Alignment
- ◆ Model in PDB format
- ◆ Other data

Modeling of BLBP

Input

- ◆ Target: Brain lipid-binding protein (BLBP)
- ◆ BLBP sequence in PIR (MODELLER) format:

```
>P1;blbp
```

```
sequence:blbp:::::::::
```

```
VDAFCATWKLTDSQNFDEYMKALGVGFATRQVGNVTKPTV IISQEGGKVVIRTQCTFKNTEINFQLGEEFEETSIDDRNCKSVV  
RLDGDKLIHVQKWDGKETNCTREIKDGKMVVTLTFGDIVAVRCYEKA*
```

Modeling of BLBP

STEP 1: Align **blbp** and **1hms** sequences

Python script for target-template alignment

```
# Example for: alignment.align()

# This will read two sequences, align them, and write the alignment
# to a file:

log.verbose()
env = environ()

aln = alignment(env)
mdl = model(env, file='1hms')
aln.append_model(mdl, align_codes='1hms')
aln.append(file='blbp.seq', align_codes=('blbp'))

# The as1.sim.mat similarity matrix is used by default:
aln.align(gap_penalties_1d=(-600, -400))
aln.write(file='blbp-1hms.ali', alignment_format='PIR')
aln.write(file='blbp-1hms.pap', alignment_format='PAP')
```

Run by typing `mod9v7 align.py` in the directory where you have the python file.
MODELLER will produce a `align.log` file

Modeling of BLBP

STEP 1: Align **blbp** and **1hms** sequences

Python script for target-template alignment

```
# Example for: alignment.align()

# This will read two sequences, align them, and write the alignment
# to a file:

log.verbose()
env = environ()

aln = alignment(env)
mdl = model(env, file='1hms')
aln.append_model(mdl, align_codes='1hms')
aln.append(file='blbp.seq', align_codes=('blbp'))

# The as1.sim.mat similarity matrix is used by default:
aln.align(gap_penalties_1d=(-600, -400))
aln.write(file='blbp-1hms.ali', alignment_format='PIR')
aln.write(file='blbp-1hms.pap', alignment_format='PAP')
```

Run by typing `mod9v7 align.py` in the directory where you have the python file.
MODELLER will produce a `align.log` file

Modeling of BLBP

STEP 1: Align **blbp** and **1hms** sequences

Python script for target-template alignment

```
# Example for: alignment.align()

# This will read two sequences, align them, and write the alignment
# to a file:

log.verbose()
env = environ()

aln = alignment(env)
mdl = model(env, file='1hms')
aln.append_model(mdl, align_codes='1hms')
aln.append(file='blbp.seq', align_codes=('blbp'))

# The as1.sim.mat similarity matrix is used by default:
aln.align(gap_penalties_ld=(-600, -400))
aln.write(file='blbp-1hms.ali', alignment_format='PIR')
aln.write(file='blbp-1hms.pap', alignment_format='PAP')
```

Run by typing `mod9v7 align.py` in the directory where you have the python file.
MODELLER will produce a `align.log` file

Modeling of BLBP

STEP 1: Align **blbp** and **1hms** sequences

Python script for target-template alignment

```
# Example for: alignment.align()

# This will read two sequences, align them, and write the alignment
# to a file:

log.verbose()
env = environ()

aln = alignment(env)
mdl = model(env, file='1hms')
aln.append_model(mdl, align_codes='1hms')
aln.append(file='blbp.seq', align_codes=('blbp'))

# The as1.sim.mat similarity matrix is used by default:
aln.align(gap_penalties_1d=(-600, -400))
aln.write(file='blbp-1hms.ali', alignment_format='PIR')
aln.write(file='blbp-1hms.pap', alignment_format='PAP')
```

Run by typing `mod9v7 align.py` in the directory where you have the python file.
MODELLER will produce a `align.log` file

Modeling of BLBP

STEP 1: Align **blbp** and **1hms** sequences

Output

```
>P1 ; 1hms
```

```
structureX:1hms: 1 : : 131 : :undefined:undefined:-1.00:-1.00
```

```
VDAFLGTWKLVD SKNFDDYMKSLGVGFATRQVASMTKPTTIEKNGDILTLKTHSTFKNTEISFKLGVEFDETTA  
DDRKVKSIVTLDGGKLVHLQKWDGQETTLVRELIDGKLILTLTHGTAVCTRTRYEKE*
```

```
>P1 ; blbp
```

```
sequence:blbp: : : : : : 0.00: 0.00
```

```
VDAFCATWKLTD SQNFDEYMKALGVGFATRQVGNVTKPTV IISQEGGKV VIRTQCTFKNTEINFQLGEEFEETSI  
DDRNCKSVVRLDGD KLIHVQKWDGKETNCTREIKDGKMVVTLTFGDIVAVRCYEKA*
```

Modeling of BLBP

STEP 1: Align **blbp** and **1hms** sequences

Output

```
>P1 ;1hms
```

```
structureX:1hms: 1 : : 131 : :undefined:undefined:-1.00:-1.00
```

```
VDAFLGTWKLVD SKNFDDYMKSLGVGFATRQVASMTKPTTIEKNGDILTLKTHSTFKNTEISFKLGVEFDETTA  
DDRKVKSIVTLDGGKLVHLQKWDGQETTLVRELIDGKLILTLTHGTAVCTRTRYEKE*
```

```
>P1 ;blbp
```

```
sequence:blbp: : : : : : 0.00: 0.00
```

```
VDAFCATWKLTD SQNFDEYMKALGVGFATRQVGNVTKPTV IISQEGGKV VIRTQCTFKNTEINFQLGEEFEETSI  
DDRNCKSVVRLDGD KLIHVQKWDGKETNCTREIKDGKMVVTLTFGDIVAVRCYEKA*
```

Modeling of BLBP

STEP 1: Align **blbp** and **1hms** sequences

Output

```

aln.pos      10      20      30      40      50      60
1hms         VDAFLGTWKLVD SKNFDDYMKSLGVGFATRQVASMTKPTTIEKNGDILTLKTHSTFKNTEISFKLGV
blbp         VDAFCATWKLTD SQNFDEYMKALGVGFATRQVGNTKPTVIISQEGGKV VIRTQCTFKNTEINFQLGE
_consrvd     ****  ****  **  ***  ***  ****  ****  ****  **  *  *  ****  **  **

aln.p        70      80      90     100     110     120     130
1hms         EFDETTADDRKVKSIVTLDGGKLVHLQKWDGQETTLVRELIDGKLILTLTHGTAVCTRITYEKE
blbp         EFEETSIDDRNCKSVVRLDGDKLIHVQKWDGKETNCTREIKDGKMMVTLTFGDIVAVRCYEKA
_consrvd     **  **  ***  **  *  ***  **  *  ****  **  **  ***  ***  *  *  *  ***

```

Modeling of BLBP

STEP 2: Model the **blbp structure using the alignment from step 1.**

Python script for model building

```
# Homology modelling by the automodel class
from modeller.automodel import *      # Load the automodel class
log.verbose()                        # request verbose output
env = environ()                      # create a new MODELLER environment

# directories for input atom files
env.io.atom_files_directory = '.:../atom_files'

a = automodel(env,
               alnfile = 'blbp-1hms.ali',      # alignment filename
               knowns   = '1hms',              # codes of the templates
               sequence = 'blbp')              # code of the target
a.starting_model= 1                    # index of the first model
a.ending_model  = 1                    # index of the last model
                                           # (determines how many models to calculate)
a.make()                               # do the actual homology modelling
```

Run by typing `mod9v7 model.py` in the directory where you have the python file.
MODELLER will produce a `model.log` file

Modeling of BLBP

STEP 2: Model the **blbp structure using the alignment from step 1.**

Python script for model building

```
# Homology modelling by the automodel class
from modeller.automodel import *      # Load the automodel class
log.verbose()                        # request verbose output
env = environ()                      # create a new MODELLER environment

# directories for input atom files
env.io.atom_files_directory = '.:../atom_files'

a = automodel(env,
               alnfile = 'blbp-1hms.ali',      # alignment filename
               knowns   = '1hms',              # codes of the templates
               sequence = 'blbp')              # code of the target
a.starting_model= 1                    # index of the first model
a.ending_model  = 1                    # index of the last model
                                           # (determines how many models to calculate)
a.make()                               # do the actual homology modelling
```

Run by typing `mod9v7 model.py` in the directory where you have the python file.
MODELLER will produce a `model.log` file

Modeling of BLBP

STEP 2: Model the **blbp structure using the alignment from step 1.**

Python script for model building

```
# Homology modelling by the automodel class
from modeller.automodel import *      # Load the automodel class
log.verbose()                        # request verbose output
env = environ()                      # create a new MODELLER environment

# directories for input atom files
env.io.atom_files_directory = '.:../atom_files'

a = automodel(env,
               alnfile = 'blbp-1hms.ali',      # alignment filename
               knowns   = '1hms',              # codes of the templates
               sequence = 'blbp')              # code of the target
a.starting_model= 1                    # index of the first model
a.ending_model  = 1                    # index of the last model
                                           # (determines how many models to calculate)
a.make()                               # do the actual homology modelling
```

Run by typing `mod9v7 model.py` in the directory where you have the python file.
MODELLER will produce a `model.log` file

Modeling of BLBP

STEP 2: Model the **blbp structure using the alignment from step 1.**

Python script for model building

PDB file

Can be viewed with Chimera

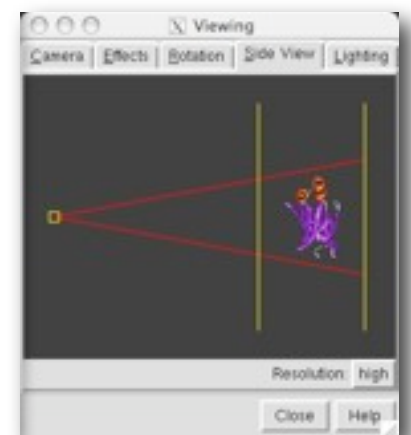
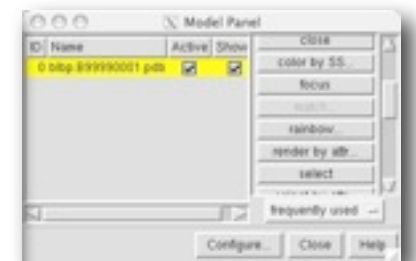
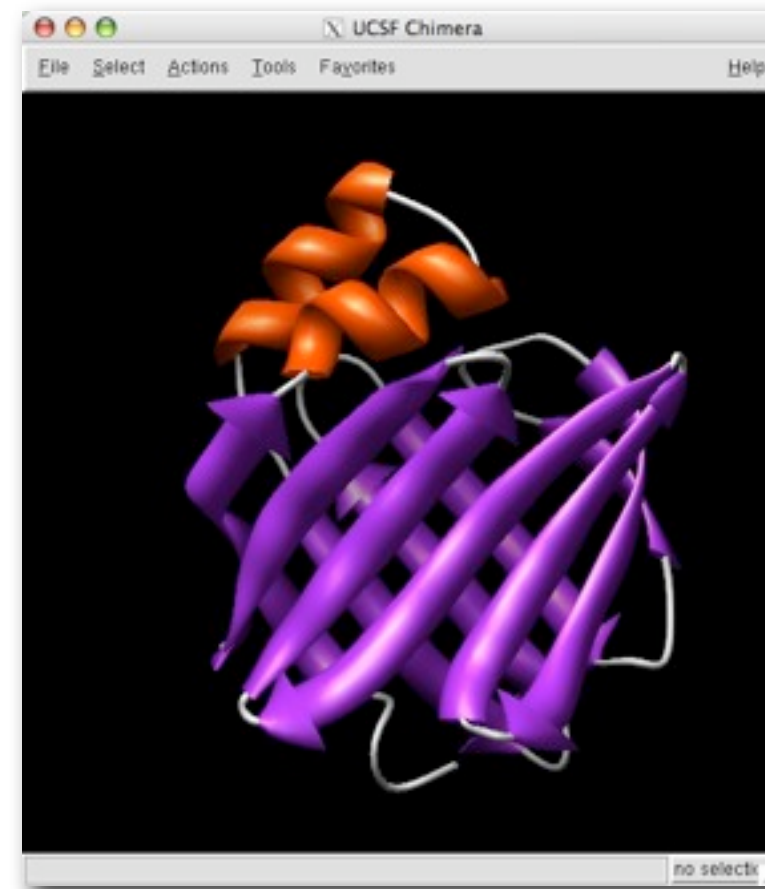
<http://www.cgl.ucsf.edu/chimera/>

Rasmol

<http://www.openrasmol.org>

PyMol

<http://pymol.sourceforge.net/>



Model file →

blbp.B99990001.pdb

<http://www.salilab.org/modeller/tutorial/>

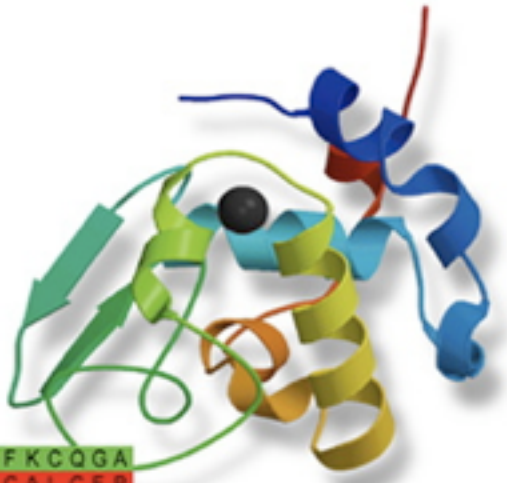
Tutorial

[http://salilab.org/modeller/tutorial/](#)

To main Sali lab pages

Modeller

Program for Comparative Protein Structure Modelling by Satisfaction of Spatial Restraints



```

A I L V G S M P R R D G M E R K D L L K A N V K I F K C Q G A
V E V C P V D C F Y E G P N F L V I H P D E C I D C A L C E P
G A C K P E C P V N I I Q G S - - I Y A I D A D S C I D C G S
C - - I A C G A C K P E C P V N I I Q G S - - I Y A I D A D S
  
```

[About MODELLER](#)
[MODELLER News](#)
[Download & Installation](#)
[Release Notes](#)
[Data file downloads](#)
[Registration](#)
[Accelrys licensing](#)
[Discussion Forum](#)
[Subscribe](#)
[Browse archives](#)
[Search archives](#)
[Documentation](#)
[FAQ](#)
[Tutorial](#)
[Online manual](#)

Tutorial

MODELLER is used for homology or comparative modeling of protein three-dimensional structures. The user provides an alignment of a sequence to be modeled with known related structures and MODELLER automatically calculates a model containing all non-hydrogen atoms.

This web site presents a tutorial for the use of MODELLER 9v2 or newer (for older versions of MODELLER, use the [old MODELLER 7v7 tutorial](#)). There are 5 modeling examples that the user can follow:

1. [Basic Modeling](#). Model a sequence with high identity to a template.
This exercise introduces the use of MODELLER in a simple case where the template selection and target-template alignments are not a problem.
2. [Advanced Modeling](#). Model a sequence based on multiple templates and bound to a ligand.
This exercise introduces the use of multiple templates, ligands and loop refinement in the process of model building with MODELLER.

MODWEB

<http://salilab.org/modweb>

The screenshot shows a web browser window titled "ModWeb: A Server For Protein Structure Modeling". The address bar shows the URL "http://modbase.compbio.ucsf.edu/ModWeb20-html/modweb.html". The page features the MODWEB logo and the subtitle "A Server for Protein Structure Modeling". On the left, there is a sidebar with links for Resources (ModWeb, ModBase, Modeller, Sali Lab, Help, Current ModWeb queue), Developed By (Eswar Narayanan), and Acknowledgements (David Eramian, Mallur S. Madhusudhan, Marc A. Marti-Renom, Ursula Pieper, Min-Yi Shen, Ben Webb, Andrej Sali). The main content area is divided into sections: General Information (with input fields for e-mail address, run name, and MODELLER access key), Input Data (with a text area for pasting sequences and a file upload option), Select Models By (with checkboxes for "Best scoring model" and "Longest well scoring model"), and Other Options (with a search speed dropdown set to "Very Fast" and a checkbox for "Upload models to ModBase"). At the bottom, there are buttons for "CALCULATE MODELS" and "RESET FORM".

ModWeb: A Server For Protein Structure Modeling

http://modbase.compbio.ucsf.edu/ModWeb20-html/modweb.html

Google

MODWEB

A Server for Protein Structure Modeling

Resources:
[ModWeb](#)
[ModBase](#)
[Modeller](#)
[Sali Lab](#)
[Help](#)
[Current ModWeb queue](#)

Developed By:
Eswar Narayanan

Acknowledgements:
David Eramian
Mallur S. Madhusudhan
Marc A. Marti-Renom
Ursula Pieper
Min-Yi Shen
Ben Webb
Andrej Sali

Please address enquiries to:
modweb@salilab.org

ModWeb version SVN.r1182

General Information

Your e-mail address

A name for your run (optional)

MODELLER access key

Input Data

Paste your sequence(s) in the window:

OR Upload a file containing your sequences (FASTA only) **Tip:** If you want to submit several sequences, you should submit them combined in one FASTA file. It greatly reduces the processing time.

no file selected

Select Models By

☒ Best scoring model ☒ Longest well scoring model

Other Options

Search speed ☒ Upload models to ModBase

MODBASE

<http://salilab.org/modbase>

Search Page

UCSF University of California, San Francisco | About UCSF | UCSF Medical Center

Home User Login ModBase Search Page ModWeb Modelling Server Help Current Logins

MODBASE
Database of Comparative Protein Structure Models

Welcome to ModBase, a database of three-dimensional protein models calculated by comparative modeling.
([Old ModBase Interface](#))

General Information
Statistics
Project Pages
Documentation
Authors and Acknowledgements
Publications
Todo List
Related Resources

Note:
MODBASE contains theoretically calculated models, not experimentally determined structures. The models may contain significant errors.

ModBase search form

Search type Display type

Search

All available datasets are selected [Select specific dataset\(s\)](#)

Search by properties

Property

Organism or

[Advanced search](#)

Model Details

UCSF University of California, San Francisco | About UCSF | UCSF Medical Center

Home User Login ModBase Search Page ModWeb Modelling Server Help Current Logins

MODBASE

Sequence Information

Primary Database Link [P43632 \(KI2S4_HUMAN\)](#)

Organism [Homo sapiens](#)

Annotation killer cell immunoglobulin-like receptor 2d34 precursor (mhc class ide nk cell receptor) (natural killer associated transcript 8) (nk4t-8)ide (p58 natural killer cell receptor clone cl-39) (p58 nk)

Sequence Length 304

Model Information

Perform action on this model

Sequence Model Coverage 

Sequence Identity 89.00%

E-Value 2e-43

Model Score 1.00

Target Region 27-221

Protein Length 304

Template PDB Code [1nkz](#)

Template Region 6-200

Dataset snp-human2

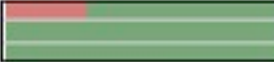
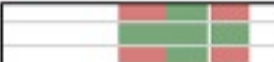


Filtered models for current sequence ([Show all models](#))



Cross-references

Sequence Overview

	☐ Q8G8A6	hypothetical protein	Pseudomonas aeruginosa	3738
	☐ Q8G9W1	hypothetical protein	Escherichia coli	1140
	☐ Q8CY62	hypothetical protein spr1965	Streptococcus pneumoniae , Streptococcus pneumoniae R6	1038

Model Overview

	☐ Q8G8C7	hypothetical protein	Pseudomonas aeruginosa	4996	2089-2158	70	37.00	7e-14	1.00	1dnyA	8-78
	☐ Q8G8C7	hypothetical protein	Pseudomonas aeruginosa	4996	492-1017	526	36.00	1e-82	1.00	1amuA	19-529
	☐ Q8G9W1	hypothetical protein	Escherichia coli	1140	349-1135	787	35.00	0	1.00	1r9dA	6-783

“take home” message

