

Comparative Protein Structure Prediction

MODELLER tutorial

```
$>mod9v5 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 (9v5) 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 :: :: :: :: :: ::
```

```
VDAFCATWKLTD SQNFDEYMKALGVGFATRQVGNVTKPTV IISQEGGKVVIRTQCTFKNTEINFQLGEEFEETSIDDRNCKSVV  
RLDGDKLIHVQKWDGKETNCTREIKDGKVVTLTFGDIVAVRCYEKA*
```

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 `mod9v5 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 `mod9v5 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 `mod9v5 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 `mod9v5 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 IISQEGGKVVIRTQCTFKNTEINFQLGEEFEETSI
```

```
DDRNCKSVVRLDGDKLIHVQKWDGKETNCTREIKDGKVVTLTFGDIVAVRCYEKA*
```

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 IISQEGGKVVIRTQCTFKNTEINFQLGEEFEETSI
```

```
DDRNCKSVVRLDGDKLIHVQKWDGKETNCTREIKDGKVVTLTFGDIVAVRCYEKA*
```

Modeling of BLBP

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

Output

```

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

aln.p        70      80      90      100     110     120     130
1hms         EFDETTADDRKVKSI VTL DGGKLVHLQKWDGQETTLVRELIDGKLILTLTHGTAVCTRITYEKE
blbp         EFEETSIDDRNCKSVV RLDGDKLIHVQKWDGKETNCTREIKDGKMMVTLTFGDIVAVRCYEKA
_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 `mod9v5 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 `mod9v5 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 `mod9v5 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

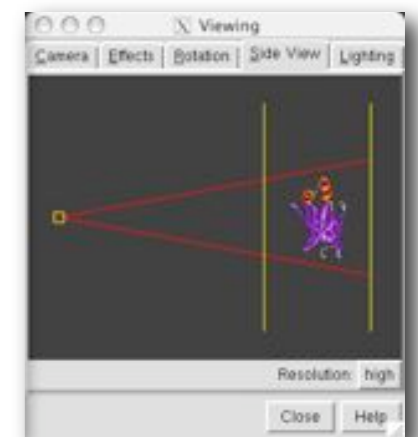
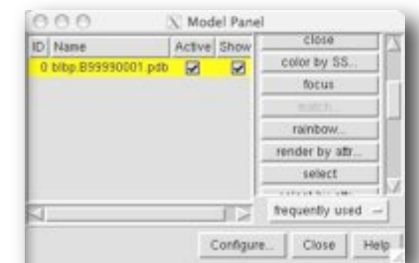
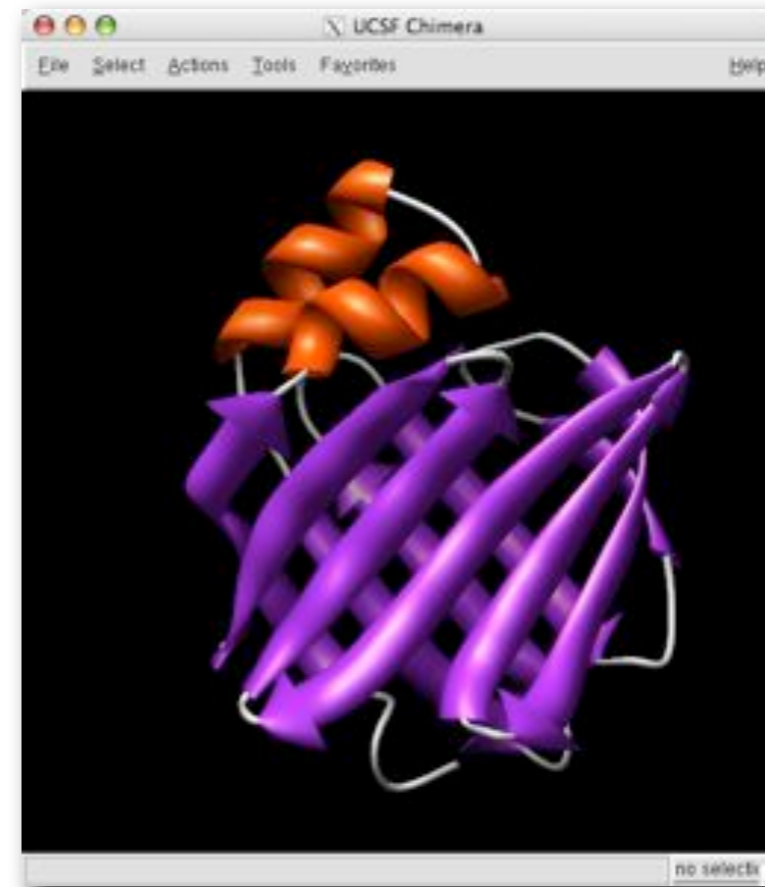
<http://www.cgl.ucsf.edu/chimera/>

Rasmol

<http://www.openrasmol.org>

PyMol

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



Model file →

blbp.B99990001.pdb

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

Modeller
Program for Comparative Protein Structure Modelling by Satisfaction of Spatial Restraints

Tutorial

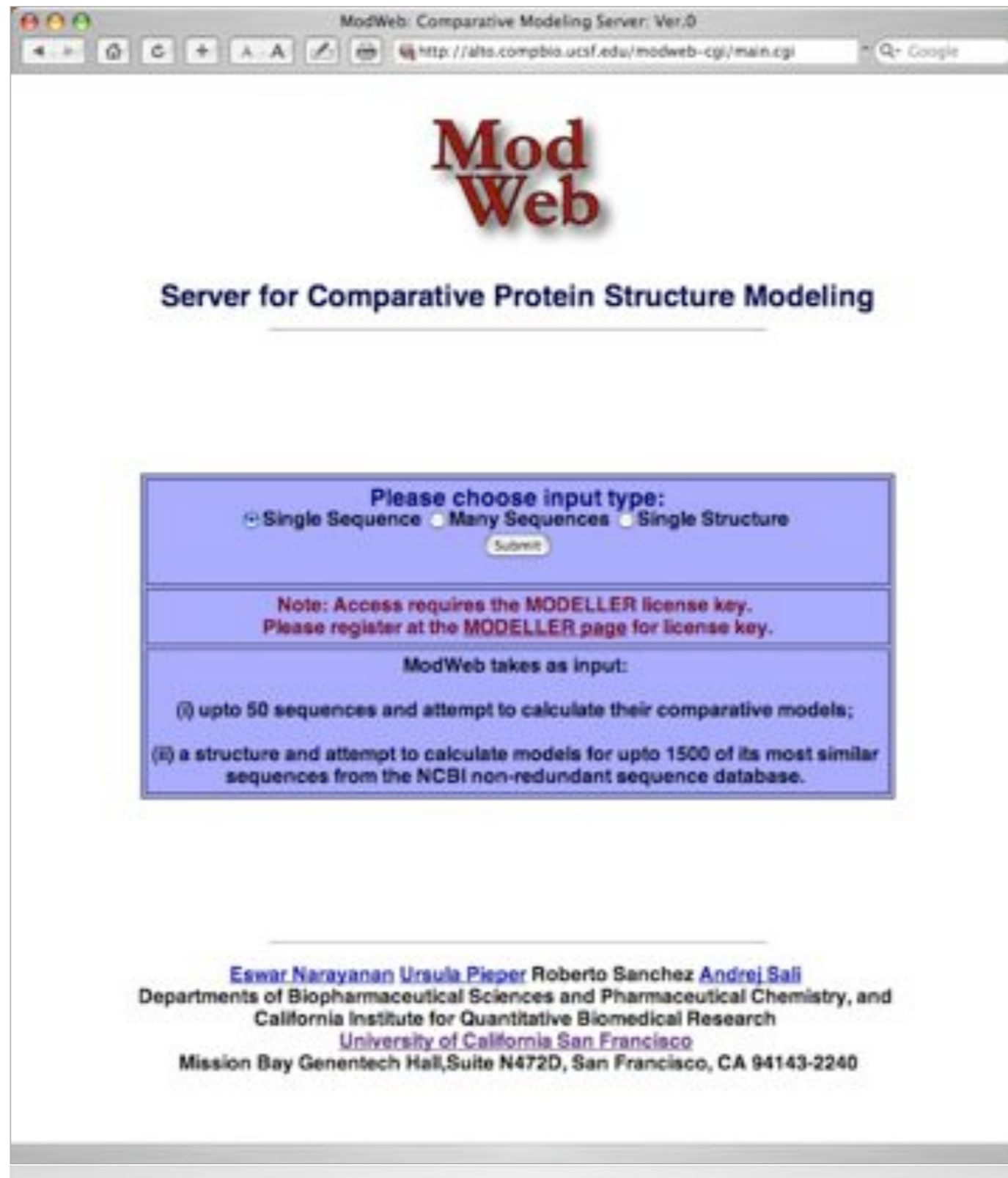
MODELLER is used for homology or comparative modeling of protein three-dimensional structures (1,2,3). The user provides an alignment of a sequence to be modeled with known related structures and MODELLER automatically calculates a model containing all non-hydrogen atoms.

This web site presents a tutorial for the use of MODELLER 6v0 (for older versions of MODELLER, use the [old MODELLER 7v7 tutorial](#)). There are 4 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 and ligands in the process of model building with MODELLER.
3. **Iterative Modeling.** Increase the accuracy of the modeling exercise by iterating the 4 step process. This exercise introduces the concept of MOULDING to improve the accuracy of comparative models.
4. **Difficult Modeling.** Model a sequence based on a low identity to a template. This exercise uses resources external to MODELLER in order to select a template for a difficult case of protein structure prediction.

MODWEB

<http://salilab.org/modweb>



The screenshot shows a web browser window with the title "ModWeb: Comparative Modeling Server: Ver.0". The address bar contains the URL "http://alts.compbio.ucsf.edu/modweb/cgi/main.cgi". The page features the "Mod Web" logo in a stylized red font. Below the logo is the text "Server for Comparative Protein Structure Modeling". A central blue box contains the following information:

Please choose input type:
 Single Sequence Many Sequences Single Structure

Note: Access requires the MODELLER license key.
Please register at the [MODELLER page](#) for license key.

ModWeb takes as input:

- (i) upto 50 sequences and attempt to calculate their comparative models;
- (ii) a structure and attempt to calculate models for upto 1500 of its most similar sequences from the NCBI non-redundant sequence database.

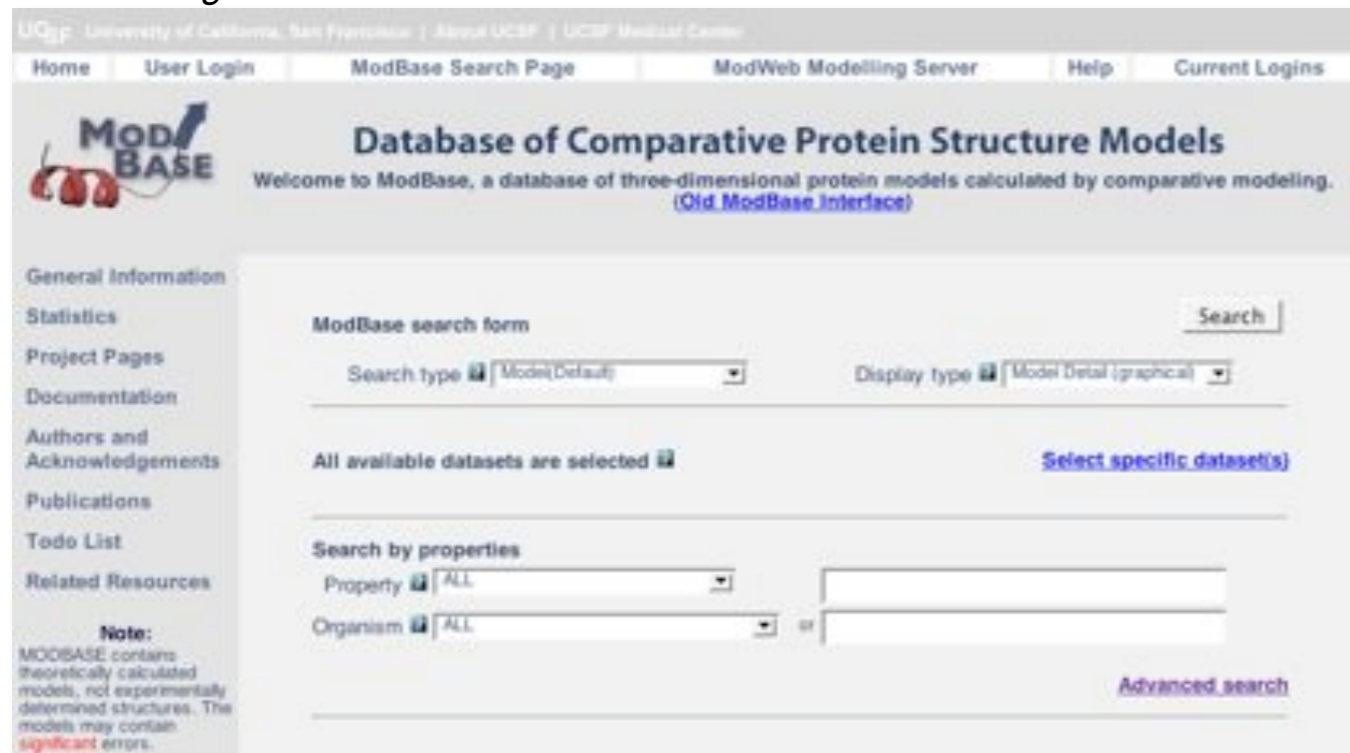
At the bottom of the page, the following text is displayed:

[Eswar Narayanan](#) [Ursula Pieper](#) [Roberto Sanchez](#) [Andrej Sali](#)
Departments of Biopharmaceutical Sciences and Pharmaceutical Chemistry, and
California Institute for Quantitative Biomedical Research
[University of California San Francisco](#)
Mission Bay Genentech Hall, Suite N472D, San Francisco, CA 94143-2240

MODBASE

<http://salilab.org/modbase>

Search Page



Model Details



Sequence Overview

SeqId	Fold	MScore	Accession	Description	Organism	Count
			Q8G8A6	hypothetical protein	<i>Pseudomonas aeruginosa</i>	3738
			Q8G9W1	hypothetical protein	<i>Escherichia coli</i>	1140
			Q8CY62	hypothetical protein spr1965	<i>Streptococcus pneumoniae, Streptococcus pneumoniae R6</i>	1038

Model Overview

Model	Accession	Description	Organism	SeqId	Fold	MScore	SeqId	Fold	MScore	Template PDB Code	Template Region
	Q8G8C7	hypothetical protein	<i>Pseudomonas aeruginosa</i>	4996	2089-2158	70	37.00	7e-14	1.00	1dnyA	8-78
	Q8G8C7	hypothetical protein	<i>Pseudomonas aeruginosa</i>	4996	492-1017	526	36.00	1e-82	1.00	1amuA	19-529
	Q8G9W1	hypothetical protein	<i>Escherichia coli</i>	1140	349-1135	787	35.00	0	1.00	1r9dA	6-783

“take home” message

