

Advanced MODELLER example

<http://salilab.org/>


Dr. Benjamin Webb

Sali Lab

University of California San Francisco

Summary

- This example demonstrates
 - using an external server (BLAST) to generate our target-template alignment, rather than MODELLER itself
 - adding bound ligands to the model
 - building models of multi-chain proteins

 **BLAST**

Basic Local Alignment Search Tool


[Home](#) [Recent Results](#) [Saved Strategies](#) [Help](#)

NCBI BLAST/ blastp suite


[blastn](#) [blastp](#) [blastx](#) [tblastn](#) [tblastx](#)

Enter Query Sequence

BLASTP programs search protein databases using a protein query. [more...](#)


Enter accession number, gi, or FASTA sequence 


[Clear](#)


Query subrange 

>seq
TLSSCKKKKKYKFDKNCERTKCIYKLSFQWTSFELDTKTLSDGMVNLPGQVKTITTD
GLKKISSMGLYKDDSTVGGSTETFEKLEETDNPWYV
.....

From
To


Or, upload file [Browse...](#) 



Job Title
Enter a descriptive title for your BLAST search 

☐ Align two or more sequences 


Choose Search Set

Database

Protein Data Bank proteins(pdb) 


Organism Optional ☐ Exclude 
Enter organism common name, binomial, or tax id. Only 20 top taxa will be shown. 

Exclude Optional ☐ Models (XM/XP) ☐ Uncultured/environmental sample sequences

Entrez Query Optional
Enter an Entrez query to limit search 

Program Selection

Algorithm

☒ blastp (protein-protein BLAST)
☐ PSI-BLAST (Position-Specific Iterated BLAST)
☐ PHI-BLAST (Pattern Hit Initiated BLAST)
Choose a BLAST algorithm 


BLAST

Search database Protein Data Bank proteins(pdb) using Blastp (protein-protein BLAST)
☐ Show results in a new window

[Algorithm parameters](#) Note: Parameter values that differ from the default are highlighted in yellow and marked with + sign

Monday, June 21, 2010


Excerpt of BLAST output

>[pdb|1MFW|A](#)  Chain A, Structure Of N-Terminal Doublecortin Domain From Dclk:
Selenomethionine Labeled Protein
Length=107

Score = 187 bits (474), Expect = 1e-48, Method: Compositional matrix adjust.
Identities = 94/105 (89%), Positives = 100/105 (95%), Gaps = 3/105 (2%)

Query	1	TLSSDKKRKKVRFYRNGERYFKGIVWALSPDRFRSFEAL--DITRTLSDGNVNLPQGVRT	58
		TLSS+KK KKVRFYRNG+RYFKGIV+A+SPDRFRSFEAL D+TRTLSD NVNLPQGVRT	
Sbjct	2	TLSSEKKAKKVRFYRNGDRYFKGIVYAISPDRFRSFEALLADLTRTLSD-NVNLPQGVRT	60
Query	59	IYTIDGLKKISSMDQLVEGDSYVCGSIEPFKKLEYTKNVNPNWSV	103
		IYTIDGLKKISS DQLVEG+SYVCGSIEPFKKLEYTKNVNPNWSV	
Sbjct	61	IYTIDGLKKISSXDQLVEGESYVCGSIEPFKKLEYTKNVNPNWSV	105

Convert to Modeller format

>[pdb|1MFW|A](#)  Chain A, Structure Of N-Terminal Doublecortin Domain From Dclk:
Selenomethionine Labeled Protein
Length=107

Score = 187 bits (474), Expect = 1e-48, Method: Compositional matrix adjust.
Identities = 94/105 (89%), Positives = 100/105 (95%), Gaps = 3/105 (2%)

Query	1	TLSSDKKRKKVRFYRNGERYFKGIVWALSPDRFRSFEAL--DITRTLSDGNVNLPQGVRT	58
		TLSS+KK KKVRFYRNG+RYFKGIV+A+SPDRFRSFEAL D+TRTLSD NVNLPQGVRT	
Sbjct	2	TLSSEKKAKKVRFYRNGDRYFKGIVYAISPDRFRSFEALLADLTRTLSD-NVNLPQGVRT	60
Query	59	IYTIDGLKKISSMDQLVEGDSYVCGSIEPFKKLEYTKNVNPNWSV	103
		IYTIDGLKKISS DQLVEG+SYVCGSIEPFKKLEYTKNVNPNWSV	
Sbjct	61	IYTIDGLKKISSXDQLVEGESYVCGSIEPFKKLEYTKNVNPNWSV	105

Convert to Modeller format

```
>pdb|1MFW|A S Chain A, Structure Of N-Terminal Doublecortin Domain From Dclk:  
Selenomethionine Labeled Protein  
Length=107
```

```
Score = 187 bits (474), Expect = 1e-48, Method: Compositional matrix adjust.  
Identities = 94/105 (89%), Positives = 100/105 (95%), Gaps = 3/105 (2%)
```

Query	1	TLSSDKKRKKVRFYRNGERYFKGIVWALSPDRFRSFEAL--DITRTLSDGNVNLPGVVRT	58
		TLSS+KK KKVRFYRNG+RYFKGIV+A+SPDRFRSFEAL D+TRTLSD NVNLPQGVRT	
Sbjct	2	TLSSEKKAKKVRFYRNGDRYFKGIVYAISPDRFRSFEALLADLTRTLSD-NVNLPQGVRT	60
Query	59	IYTIDGLKKISSMDQLVEGDSYVCGSIEPFKKLEYTKNVNPNWSV	103
		IYTIDGLKKISS DQLVEG+SYVCGSIEPFKKLEYTKNVNPNWSV	
Sbjct	61	IYTIDGLKKISSXDQLVEGESYVCGSIEPFKKLEYTKNVNPNWSV	105

1. Remove headers, prefix/suffix, and match information

Convert to Modeller format

TLSSDKKRKKVRFYRNGERYFKGIVWALSPDRFRSFEAL--DITRTLSDGNVNLPGVVRT

TLSSEKKAKKVRFYRNGDRYFKGIVYAISPDRFRSFEALLADLTRTLSD-NVNLPQGVRT

IYTIDGLKKISSMDQLVEGDSYVCGSIEPFKKLEYTKNVNPNWSV

IYTIDGLKKISSXDQLVEGESYVCGSIEPFKKLEYTKNVNPNWSV

1. Remove headers, prefix/suffix, and match information

Convert to Modeller format

TLSSDKKRKKVRFYRNGERYFKGIVWALSPDRFRSFEAL--DITRTLSDGNVNLPOGVRT

TLSSEKKAKKVRFYRNGDRYFKGIVYAISPDRFRSFEALLADLTRTLSD-NVNLPQGVRT

IYTIDGLKKISSMDQLVEGDSYVCGSIEPFKKLEYTKNVNPNWSV

IYTIDGLKKISSXDQLVEGESYVCGSIEPFKKLEYTKNVNPNWSV

1. Remove headers, prefix/suffix, and match information
2. Gather target and template lines

Convert to Modeller format

TLSSDKKRKKVRFYRNGERYFKGIVWALSPDRFRSFEAL--DITRTLSDGNVNLPOGVRT

TLSSEKKAKKVRFYRNGDRYFKGIVYAISPDRFRSFEALLADLTRTLSD-NVNLPQGVRT

IYTIDGLKKISSMDQLVEGDSYVCGSIEPFKKLEYTKNVNPNWSV

IYTIDGLKKISSXDQLVEGESYVCGSIEPFKKLEYTKNVNPNWSV

TLSSDKKRKKVRFYRNGERYFKGIVWALSPDRFRSFEAL--DITRTLSDGNVNLPOGVRT
IYTIDGLKKISSMDQLVEGDSYVCGSIEPFKKLEYTKNVNPNWSV

TLSSEKKAKKVRFYRNGDRYFKGIVYAISPDRFRSFEALLADLTRTLSD-NVNLPQGVRT
IYTIDGLKKISSXDQLVEGESYVCGSIEPFKKLEYTKNVNPNWSV

1. Remove headers, prefix/suffix, and match information
2. Gather target and template lines

Convert to Modeller format

**TLSSDKKRKKVRFYRNGERYFKGIVWALSPDRFRSFEAL--DITRTLSDGNVNLPOGVRT
IYTIDGLKKISSMDQLVEGDSYVCGSIEPFKKLEYTKNVNPNWSV**

**TLSEKKAKKVRFYRNGDRYFKGIVYAISPDRFRSFEALLADLTRLSD-NVNLPOGVRT
IYTIDGLKKISSXDQLVEGESYVCGSIEPFKKLEYTKNVNPNWSV**

1. Remove headers, prefix/suffix, and match information
2. Gather target and template lines

Convert to Modeller format

**TLSSDKKRKKVRFYRNGERYFKGIVWALSPDRFRSFEAL--DITRTLSDGNVNLPOGVRT
IYTIDGLKKISSMDQLVEGDSYVCGSIEPFKKLEYTKNVNPNWSV**

**TLSEKKAKKVRFYRNGDRYFKGIVYAI SPDRFRSFEALLADLTRLSD-NVNLPOGVRT
IYTIDGLKKISSXDQLVEGESYVCGSIEPFKKLEYTKNVNPNWSV**

1. Remove headers, prefix/suffix, and match information
2. Gather target and template lines
3. Add PIR headers and '*' terminator

Convert to Modeller format

```
>P1;sequence
sequence::::::::::
TLSSDKKRKKVRFYRNGERYFKGIVWALSPDRFRSFEAL--DITRTLSDGNVNLPOGVRT
IYTIDGLKKISSMDQLVEGDSYVCGSIEPFKKLEYTKNVNPNWSV*
```

```
>P1;1mfwA
structureX:1mfw:1:A:104:A::::
TLSSEKKAKKVRFYRNGDRYFKGIVYAI SPDRFRSFEALLADLTRTLSD-NVNLPQGVRT
IYTIDGLKKISSXDQLVEGESYVCGSIEPFKKLEYTKNVNPNWSV*
```

1. Remove headers, prefix/suffix, and match information
2. Gather target and template lines
3. Add PIR headers and '*' terminator

Proceed to modeling...

- align1.ali:

```
>P1;sequence
sequence:::::::::
TLSSDKKRRKKVRFYRNGERYFKGIVWALSPDRFRSFEAL--DITRTLSDGNVNLPOGVRT
IYTIDGLKKISSMDQLVEGDSYVCGSIEPFKKLEYTKNVNPNWSV*

>P1;lmfwA
structureX:lmfw:1:A:104:A::::
TLSSEKKAKKKVRFYRNGDRYFKGIVYAISPDRFRSFEALLADLTRTLSD--NVNLPQGVRT
IYTIDGLKKISSXDQLVEGESYVCGSIEPFKKLEYTKNVNPNWSV*
```

- model1.py:

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

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

a = automodel(env, alnfile='align1.ali',
              knowns='lmfwA', sequence='sequence')
a.starting_model = 1
a.ending_model = 5
a.make()
```

Proceed to modeling...

- align1.ali:

```
>P1;sequence
sequence:::::::::
TLSSDKKRKKVRFYRNGERYFKGIVWALSPDRFRSFEAL--DITRTLSDGNVNLPGVVRT
IYTIDGLKKISSMDQLVEGDSYVCGSIEPFKKLEYTKNVNPNWSV*

>P1;lmfwA
structureX:lmfw:1:A:104:A::::
TLSSEKKAKKVRFYRNGDRYFKGIVYAI SPDRFRSFEALLADLTRTLSD~NVNLPQGVRT
IYTIDGLKKISSXDQLVEGESYVCGSIEPFKKLEYTKNVNPNWSV*
```

- model1.py:

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

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

a = automodel(env, alnfile='align1.ali',
              knowns='lmfwA', sequence='sequence')
a.starting_model = 1
a.ending_model = 5
a.make()
```

Run in the usual way...

It doesn't work!

```
rdpdb__303E> No atoms were read from the specified input PDB file, since the
               starting residue number and/or chain id in MODEL_SEGMENT (or
               the alignment file header) was not found;
               requested starting position: residue number " 1", chain " A";
               atom file name:  atom_files/pdb1mfw.ent
```

- We told MODELLER to read residues 1 through 104 in chain A, but it couldn't find them
- So let's look in the actual PDB file for the ATOM records:

```
ATOM      1  N   THR A  49         18.633  -2.703  42.182   1.00  58.62           N
...
ATOM     855  CG2 VAL A 154         9.314   6.191  42.676   1.00  25.16           C
```

- So, we need to change the alignment header

>P1 ; sequence

sequence :::::::::::

TLSSDKKRKKVRFYRNGERYFKGIVWALSPDRFRSFEAL--DITRTLSDGNVNLPGVVRT
IYTIDGLKKISSMDQLVEGDSYVCGSIEPFKKLEYTKNVNPNWSV*

>P1 ; 1mfwA

structureX:1mfw:1:A:104:A::::

TLSSEKKAKKVRFYRNGDRYFKGIVYAISPDRFRSFEALLADLTRLSD-NVNLPQVVRT
IYTIDGLKKISSXDQLVEGESYVCGSIEPFKKLEYTKNVNPNWSV*



>P1 ; sequence

sequence :::::::::::

TLSSDKKRKKVRFYRNGERYFKGIVWALSPDRFRSFEAL--DITRTLSDGNVNLPGVVRT
IYTIDGLKKISSMDQLVEGDSYVCGSIEPFKKLEYTKNVNPNWSV*

>P1 ; 1mfwA

structureX:1mfw:49:A:154:A::::

TLSSEKKAKKVRFYRNGDRYFKGIVYAISPDRFRSFEALLADLTRLSD-NVNLPQVVRT
IYTIDGLKKISSXDQLVEGESYVCGSIEPFKKLEYTKNVNPNWSV*

It still doesn't work!

```
read_te_291E> Sequence difference between alignment and pdb :  
                x (mismatch at alignment position      72)  
Alignment      KKISSXDQLVEGESYVCGSIEPFKKLEYTKNVNPNWSV  
PDB            KKISSMDQLVEGESYVCGSIEPFKKLEYTKNVNPNWSVNV  
Match          *****
```

- Again, by inspection of the PDB file, we can see that
 - the mismatched residue is an MSE; MODELLER treats MSE as regular MET (M), but BLAST evidently thinks it is UNK (X)
 - the full PDB chain contains two extra residues that are not in our original sequence
- Fix by
 - changing X to M
 - telling MODELLER to stop reading the PDB at residue 152, not 154

>P1 ; sequence

sequence :::::::::::

TLSSDKKRKKVRFYRNGERYFKGIVWALSPDRFRSFEAL--DITRTLSDGNVNLPGVVRT
IYTIDGLKKISSMDQLVEGDSYVCGSIEPFKKLEYTKNVNPNWSV*

>P1 ; 1mfwA

structureX:1mfw:49:A:154:A::::

TLSSEKKAKKVRFYRNGDRYFKGIVYAISPDRFRSFEALLADLTRLSD-NVNLPQVVRT
IYTIDGLKKISSXDQLVEGESYVCGSIEPFKKLEYTKNVNPNWSV*



>P1 ; sequence

sequence :::::::::::

TLSSDKKRKKVRFYRNGERYFKGIVWALSPDRFRSFEAL--DITRTLSDGNVNLPGVVRT
IYTIDGLKKISSMDQLVEGDSYVCGSIEPFKKLEYTKNVNPNWSV*

>P1 ; 1mfwA

structureX:1mfw:49:A:152:A::::

TLSSEKKAKKVRFYRNGDRYFKGIVYAISPDRFRSFEALLADLTRLSD-NVNLPQVVRT
IYTIDGLKKISSMDQLVEGESYVCGSIEPFKKLEYTKNVNPNWSV*

Incorporate ligands

- Modeller can add ligands (ions, small molecules, DNA/RNA) to the model as long as they are present in the template
 - Cannot model holo from apo or vice versa
- Typically, we use a “block” ligand (one letter code ‘.’) to copy the ligand rigidly from template to target
- Must tell Modeller to
 - read HETATM records from the PDB in our Python script (usually it only reads ATOM records)
 - modify the alignment to include the template ligand(s)
 - align ligands in the target with appropriate template ligand(s)

>P1 ; sequence

sequence :::::::::::

TLSSDKKRKKVRFYRNGERYFKGIVWALSPDRFRSFEAL--DITRTLSDGNVNLPGVVRT
IYTIDGLKKISSMDQLVEGDSYVCGSIEPFKKLEYTKNVNPNWSV*

>P1 ; 1mfwA

structureX:1mfw:49:A:152:A:::::

TLSSEKKAKKVRFYRNGDRYFKGIVYAISPDRFRSFEALLADLTRLSD-NVNLPQVVRT
IYTIDGLKKISSMDQLVEGESYVCGSIEPFKKLEYTKNVNPNWSV*



>P1 ; seqligand

sequence :::::::::::

TLSSDKKRKKVRFYRNGERYFKGIVWALSPDRFRSFEAL--DITRTLSDGNVNLPGVVRT
IYTIDGLKKISSMDQLVEGDSYVCGSIEPFKKLEYTKNVNPNWSV--/-.*

>P1 ; 1mfwA

structureX:1mfw:49:A:1002:A:::::

TLSSEKKAKKVRFYRNGDRYFKGIVYAISPDRFRSFEALLADLTRLSD-NVNLPQVVRT
IYTIDGLKKISSMDQLVEGESYVCGSIEPFKKLEYTKNVNPNWSVNV/..*

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

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

a = automodel(env, alnfile='align3.ali',
               knowns='lmfwA', sequence='sequence')
a.starting_model = 1
a.ending_model = 5
a.make()
```



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

env = environ()
env.io.atom_files_directory = ['.', 'atom_files']
env.io.hetatm = True

a = automodel(env, alnfile='align-ligand.ali',
               knowns='lmfwA', sequence='seqligand')
a.starting_model = 1
a.ending_model = 5
a.make()
```

Model dimer from a dimer template

- Same as regular modeling, except alignment file lists both chains of the template and target, separated by a chain break (forward slash):

```
>P1 ; sequence
```

```
sequence :: :: :: :: :: :: :: ::
```

```
SSSSSSSSSS/SSSSSSSSSS*
```

```
>P1 ; 1abc
```

```
structureX:1abc:1:A:10:B: :: ::
```

```
AAAAAAAAAA/BBBBBBBBBB*
```

- Note that generally the two chain breaks should be aligned

Model dimer from a monomer template

- Use two copies of your monomer, one for each chain in the target:

```
>P1;sequence
sequence:::::::::
SSSSSSSSSS/SSSSSSSSSS*
```

```
>P1;1abc-1
structureX:1abc:1:A:10:A::::
AAAAAAAAAA/-----*
```

```
>P1;1abc-2
structureX:1abc:1:A:10:A::::
-----/AAAAAAAAAA*
```

- Note that chain-chain interactions need to be added