**MODDELING DMI3 PROTEIN**

**Database for DMI3 peptide sequence**

<https://plants.ensembl.org/Glycine_max/Info/Index?db=core>

<ftp://ftp.ensemblgenomes.org/pub/plants/release-43/fasta/glycine_max/dna/>

<https://phytozome.jgi.doe.gov/pz/portal.html#!gene?search=1&detail=1&method=4433&searchText=transcriptid:30497334>

**Scripts 1 a and 1 b**

1. **build\_profile.py and result**

**Script**

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='DMI.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')

**2. compare.py and the result**

**Script**

from modeller import \*

env = environ()

aln = alignment(env)

for (pdb, chain) in (('1ywr', 'A'), ('1gz8', 'A'), ('2bfx', 'A'),

('1blx', 'A'), ('1uu3', 'A'), ('1bjf', '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)