Chapter 3: MembStruk GUI version 4.10

Introduction

Membstruk is an automated method for the development of protein structures for G-Protein Coupled Receptors (GPCRS). This manual will describe the current procedures and GUI environment for running this method. The future methods in testing are not yet validated will be mentioned and show their current place in the methodology. However, these additional scripts will not be made available until they are out of the beta testing phase.

The MembStruk method is separated into two main parts: 1) TM2ndS – Prediction of the transmembrane (TM) region, and 2) Optimization of the template TM structures and final build of the 3D structure. The prediction of the TM regions is critical for the method, since validations of Bovine Rhodopsin (crystal structure 1B, 1HZX) have show that the final structure's resolution is dependent on the initial TM prediction. The better the TM prediction the better quality structure produced from the method.

This manual will not only try to cover the basics to running the scripts and producing a valid structure for the target GPCR, it will also cover the chemistry looked at in the designing of each step in the method and possible pitfalls and key items to look for in a good protein structure. The number one idea to keep in mind is that the TM region is located in a lipid medium with a low dielectric, so the protein will want to avoid having highly charged residues interacting with the lipid.

Each chapter will be devoted to the development of one particular script or step in the method for understanding and utilization. Each chapter will also show an example of how that step was use on Bovine Rhodopsin and the problems and successes involved.

Credit must be given to all the members of the Goddard BioGroup, who have each contributed to the progress and development of these methods.

Installation

In order to install this program you must have the following programs already installed on you machines: Biograf with alias to the current directory executables that are in the format so that a background macro (addH.macro) will run with the following command: “bgver 330; /biograf/bio_msc batbio dreidii addH.macro”. Biograf will use one of your Cerius2 tokens to run. You will need clustalw version 1.74. Blast is also used and the version given by the contents of the VERSION file by compile date are:

The executable files were built on Tue Apr 30 12:59:07 EDT 2002

The version number of each individual application may be found in the appropriate documentation files in ./ncbi/doc/

uname -a ouput is: IRIX64 cruncher 6.2 03131016 IP25

You will also need to install scwrl 3.0 (Linux version) and place the executable in the $membstruk/bin/ directory. You will also need to create a link called mpsim in the $membstruk/bin/ directory and have it point to the current running mpsim. Mpsim will also need access to the linear algebra library (liblapack, liblabast). You may need to set symbolic links to the linear algebra library to use version 2.0 since that is what mpsim was compiled with.

You will need to have the MembStruk directories seen in the same placement on all Linux and SGI machines that will run the GUI. This is important for setting up Linux clusters to use the MembStruk method. The cluster must be able to ssh onto the SGI machine (using NAT is one way to do this). You can set up an alias to replace ssh with

rsh for older machines that do not support ssh. However, the Linux cluster must understand PBS scripts and be run using the qsub command. Again, you can bypass this using an alias to replace qsub with another command. Also by default to run the PBS script through qsub, you type borg as the Linux machine to use. Just alias the name "borg" to whatever the full machine name is of the master node of the cluster to be used.

Every person who is going to use the program should have the following in their .cshrc file:

Set in your path the directory containing the TMPred2ndS executable's directory.

Set the following environmental variables:

BLAST_DATA - sequence archives

BLAST_DIR - blast exe

membstruk – the directory where the executables are located. No "/" located on the end of this environmental variable.

setenv MPSIM_INFO $membstruk/FF/

setenv FF_INFO $membstruk/FF/

setenv MPSIM_DATA $membstruk/bin/mpsimdata/

To run the Modeller script set the following variables for 6v2:

MODINSTALL6v2 – Set this environmental variable to the directory that contains the modeler executables.

EXECUTABLE_TYPE6v2 - Set this environmental variable to the machine version.

LIBS_LIB6v2 - Set this environmental variable to the lib file. Ex)

$MODINSTALL6v2/modlib/libs.lib

KEY_MODELLER6v2 - Set this environmental variable to the key value used.

alias mod mod6v2

set path = ($path $MODINSTALL6v2/bin)

If you are using a different version of Modeller set up the environmental variables accordingly, but change the mod alias to reflect the new command.

memtmp1 – Set this environmental variable to a temp directory on the working Linux machine. No "/" on the end. You choose this variable in the GUI under Linux temp 1.

memtmp2 – Set this environmental variable to a temp directory on the working Linux machine. No "/" on the end. You choose this variable in the GUI under Linux temp 2.

memtmp3 – Set this environmental variable to a temp directory on the working Linux machine. No "/" on the end. You choose this variable in the GUI under Linux temp 3.

memsgi1 – Set this environmental variable to a temp directory on the working SGI machine. No "/" on the end. You choose this variable in the GUI under SGI temp 1.

memsgi2 – Set this environmental variable to a temp directory on the working SGI machine. No "/" on the end. You choose this variable in the GUI under SGI temp 2.

memsgi3 – Set this environmental variable to a temp directory on the working SGI machine. No "/" on the end. You choose this variable in the GUI under SGI temp 3.

And for WAG group members to run the MoleculeGL have the following in your .cshrc:

setenv MGL /project/Biogroup/Software/moleculeGL

For some of the perl scripts they use the first line directory of:

/usr/sbin/perl

or

/usr/local/bin/perl

so set up symbolic links in these directories if perl is not set up in these directories.

NOTE: All the needed variables are set up in the C-Shell source script $membstruk/membstruk.sh. Just source this file in your .cshrc file to have all variables set up correctly.

Errors

1) Make sure that all *.ctl files in your directory have the right directory for the forcefields used. See the example), Section 1.1, Chapter 1.

Version History

The credits window appears at the start of the program and shows the latest changes and the date that those changes were implemented. Below is the current history of the developmental changes to the MembStruk methods.

Version 1.00 (1999) includes:
- Used on S ORs (Wely)
- Initial trial of Membstruk method

Version 1.50 (5/1/2001) includes:
- Used on preM-I7 and preR-I7 (Spencer)
- First build of Linux scripts to run Methods
- Update to FF and .ctl file on Neimo and RBMD

Version 2.00 (12/1/2001) includes:
- Used on M-I7 and M2R-I7 (Spencer)
- First inclusion of Rotmin to the methodology
- Initial build of GUI for Membstruk Methods

Version 3.00 (10/14/2002) includes:
- Used on initial D2 (Yashar)
- Addition of Phobic rotation to the template file

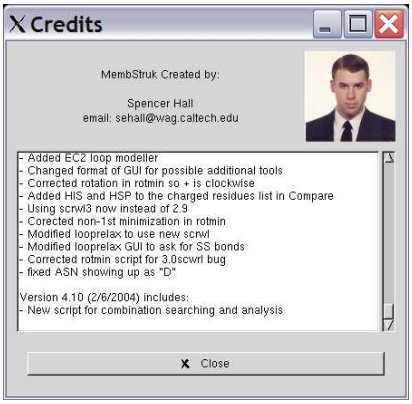


Figure 1 – MembStruk Credits Window

- Use of bisector angle of three neighbors for rotation phobic
- Use of TMPred reccomended to build template

Version 3.50 includes:
- Addition of Translation by HPM centers
- Use of Center of protein for hydrophobic rotation
- Updated methods used on Rhodopsin (Rene 1st structure)

Version 3.51 includes:
- New Householders and QR eigenvector method
- Access to setup files for Neimo and Neimo MD buttons
- New decimal translation on Hydro-Centers for Template rotation
- Translation removed from BGF rotation button
- New Credits window on startup

Version 3.52 includes:
- New Remote Machine entry
- Updated Credits window
- All Scripts runnable on Borg
- Fixed random horizontal translation bug
- Fixed MD Neimo snap2bgf bug

Version 3.53 includes:
- Fixes bug when running non-borg scripts
- Fixed output window that only shows initial rotation pictiure.
- Fixed bug that left HPM initial copy file.
- Created membstruk.version file
- Fixes bug with neimo_Dyn.ctl file copy

Version 3.54 includes:
- Fixes fixhelix bug that recenters after Neimo
- Restores older Neimo output files
- New radio buttons added to Neimo windows to choose temp drive
- Fixed bug in Template Rotation 3.0 script

Version 3.55 (6/13/2003) includes:
- Allows multiple Prefixes running in the same directory
- Fixes another bug (wrong directory) in MD Neimo script
- Fixes RBMD bug with the golbal $2 to ${protein} error
- Fixes MembComp running error
- Fixes error in Rotmin (template file misnamed)

Version 3.56 (6/24/2003) includes:
- Updated Credits window
- Corrected error message on Neimo
- Fixed filename length error in nacl-append

Version 3.57 (7/25/2003) includes:
- Changed Neimo window to include neutral qeq
- Added qeq neutral charges to the Neimo scripts

Version 3.58 (8/15/2003) includes:
- Added functionality to the Individual helices button
- Created new windows to run scripts for MD/Neimo Helix
- Created window to run Helix merger script
- Fixed a few perpetrated typos
- Fixed open MD/Neimo multiple times bug
- Fixed Setup files button to copy {prefix}.ctl
- Extended sidechain builder fixed (MD Neimo set as default)
- Updated compare to hold 45 residue names in scale.dat file
- Updated compare to print out initial inputs

Version 3.59 (10/17/2003) includes:
- Set Fixhelix default to use Cartesian MD
- Updated hcenterTR scripts to ask for Hydro. Scale
- Updated GUI to ask for Hydro. Scale (White scale included in /ul/sehall/MEMBSTRUK/hcenterTR/)
- Fixed Neimo/MD bug (now the default truely is MD) (window reset the mtd variable back to 0)
- Fixed small bug in rotmin with - file not crashing right

Version 4.00 (11/17/2003) includes:
- Added buttons for loop scripts
- included functionality for accessing modeller for loops
- New script to print out predictions for Whatif
- New script to rotate helix in bundle and produce a graph
- Fixed rotmin.IH script to include Lipid analysis for Helix rotation
- New script to minimize new loop positions and possibly scwrl
- Fixed ions not being fixed in the rotmin script
- Added Hydrophobic vector rotation to rotation code
- Set Default rotation for after MD/Neimo to Hydrophobic Rotation
- Set Default rotation to use exact HPMCenters plane
- Fixed lipid file not correct bug
- Fixed Rotmin to have fixed ions again
- Changed calling procedure for modeller
- Fixed Bug with translations from exact alignment
- Rotation based on C-alpha mom. of inert. axis

- Fixed bug with rotmin.IndHel not liking pos. energies
- Added directory FF to hold links to the most current Files for Biograf and MpSim
- Fixed conversion problem with Indvidual helix Fixhelix for Neutral QeQ atoms
- Fixed FF problems with Modeller using Biograf
- Fixed Modeller NT/CT charge problem after scwrl
- Fixed Awk ${2} problem in RBMDL script
- Fixed rotation bug again
- Corrected non-copy of snapshots from fixhelix.IndvHel
- Corrected non-match of lipid before scwrl
- Fixed the matching of lipid in lipid rotational analysis
- Added TM2ndS and EC2 functions
- Used on Rhodopsin in BioPhys paper (Rene)

Version 4.05 (1/1/2004) includes:
- Added correct script for moleculeGL and an {prefix}.eng file
- MoleculeGL now used to view helical snapshots
- TMpred is now a part of MembStruk GUI
- Added EC2 loop modeller
- Changed format of GUI for possible additional tools
- Corrected rotation in rotmin so + is clockwise
- Added HIS and HSP to the charged residues list in Compare
- Using scrwl3 now instead of 2.9
- Corected non-1st minimization in rotmin
- Modified looprelax to use new scrwl
- Modified looprelax GUI to ask for SS bonds
- Corrected rotmin script for 3.0scwrl bug
- fixed ASN showing up as "D"

Version 4.10 (2/6/2004) includes:
- New script for combination searching and analysis
- Updated rotminL.script to use env variable membstruk, /bin directory (4/22/2004)
- Updated all scripts to use environmental variables to find installed directory
- Updated GUI to give use of environmental variables
- Fixed rotmin translation errors and now back in the Default Method.

Chapter 0 - Main GUI interface

The main program window of MembStruk is set up to put the main steps of the method into sequentially ordered buttons. The buttons closest to the top-left are first and moving down and then towards the right. The figure below shows the numbering of how the buttons should be used in a default manner.

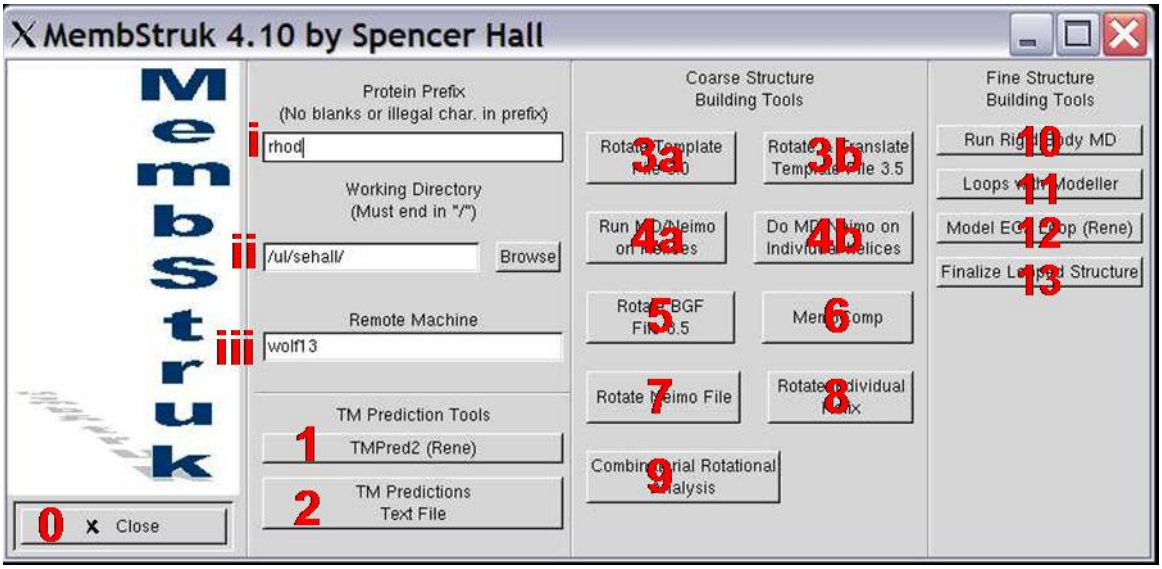


Figure 2 – Main MembStruk GUI window.

Text Entry Boxes:

i) Type in here the prefix name you want all the structure files to start with. This works best if no longer than 8 characters in length to avoid long file names. This will be refered to as {prefix}. As long as two proteins don't have the same prefix name they can be run in the same directory.

ii) Type in here the directory name you want to have all files from the finished run deposited into. You can use the Browse button to the right of this box to open up a file manager window to find the directory. This directory must end with a "/" symbol.

iii) Enter in here the name of the remote linux machine to rsh the script onto and run in one of the temp directories. If you enter "borg" it will rsh onto the borg cluster and run the script in the queue.

Buttons:

0) This button when pressed will shut down the entire program. Closing the window by any other means without pressing this button will on some machines leave the program running in the background.

1) This button will bring up the TMPred2ndS program interface to find the TM predicted regions from a fasta formatted sequence file. See Chapter 1.

2) This button will run a simple awk script to reformat the TM predictions held in the hel1 – hel7 files into an easier to read format. See Chapter 2.

3) These buttons run a script to rotate the template helices by hydrophobicity and translate by hydrophobic moments.
    a) This button runs an older version of the template rotation and contains no code to translate the helices. This will not be discussed in this manual and is included to recreate old protein structures if necessary.
    b) This is the current button to use. It will bring up a window to run a rotational and translational script. See chapter 3.

4) These buttons will handle the individual dynamics run on each helix.
    a) This button will open up a window to run the dynamics/neimo simulation on the entire template. See chapter 4.
    b) This button will open up a window to run specialized simulations on a individual helix and replace the old helix structure in the starting structure with a new one. This is to be used in the event of problems with button 4a. See chapter 5.

5) This will bring up a window that will allow the hydrophobic rotation of a bgf file. See chapter 6.

6) This will start the MembComp program that can be used for a more detailed analysis of the protein structure and alignment. Not covered in this manual, but it is recommended that this be used to check the rotations and salt bridges of structures.

7) This button will bring up the Rotmin window that will rotate the structure through a series of rotations on each helix and keep the lowest energy structure after each series of rotations. See chapter 7.

8) Rotate an individual helix and produce a graph of the energy scores. This should be use either before RBMD or after depending on the time allowed. See chapter 8.

9) Produce a series of minimized structures that are combinations of the possible rotations allowed by the energy scan produced by button 8. See chapter 9.

10) This will open the RBMD window to optimize the packing of the helices. See chapter 10.

11) This will open up a window to run Pete's modeler (all loops at once) script. The current method uses whatif to add the loops. However, this method is supported in chapter 11.

12) Model the folding of the EC2 loop using this button to access Rene's program ***BETA Testing***. See chapter 12.

13) This final script will fully minimize the structure in lipid and allow the option of scwrling the entire protein or just the loops. See chapter 13.

Section 0.1 - MembStruk Methodology Outline

A) Running TMPred2ndS

1) Click Button 1 and follow the outline presented in section 1.1.
2) Click Button 2 and create the TM text file to use to check the accuracy of the TM predictions.

B) Running MembStruk

1) Rotate and Translate the Structure.
    a) Press Button 3b to translate and rotate the pdb template file.

2) Run Fixhelix on the structure.
    a) Press Button 4a
    b) Check your structure visually. For large bends or deviations in a helix goto step c, if the whole structure is bad redo step a) with different defaults. If everything is fine goto step 3).
    c) Press 4b and run the scripts for every bad helix
    d) Visually check your structure. If still bad back to a) and try different parameters.

3) Run Coarse Rotation
    a) Press Button 5 (can do this 2-3 times if you want convergence, slight error increase though in translation).

4) Check your Structure in MembComp (***Beta only***)
    a) Press Button 6 and check the overall positions of residues of importance.

5) Rotate according to Energy properties
    a) Press Button 7 to run the rotmin script.

Note: Currently, there are two ways of running the order of steps 6, 7, and 8. You can either do them as 6, 7, 8 or 8, 6, 7 which takes 4 days longer but is slightly better in terms of accuracy (my preference).

6) Rotational Energy Scan for Individual Helices (**Business optional, Beta mandatory)**
    a) Press Button 8 for every desired helix (I prefer to do all), and use the graphs to determine possible combinations. If this is being done after step 8 use the final bgf file and then make sure to check yes on lipids.

7) Combinatorial Structure Design (**Beta only**)

a) Press Button 9 and enter in all combinations. Make sure to check yes for lipids if running this after having done step 8.
b) Examine the structures and choose the best (or several best).

8) Rigid Body Molecular Dynamics
a) Press Button 10.

9) Add Loops
Do one of the following:
a) Press Button 11 (**Business preferred**)
b) Use Whatif and relax the structure afterwards ($membstruck/loops/pdb2CYXbgf.script or Quanta)
c) Use SwissPDB Viewer
e) Use your own method

10) EC Loop 2 Simulation of Folding (**Beta, Unreliable**)
a) Press Button 12 and follow directions (take notes!).

11) Relax Final Structure for Docking
a) Press Button 13.

Chapter 1 – TMPred2ndS (TM Predictions)

Section 1.1 – A Quick Start Guide to Running TMPred2ndS

1) This protocol predicts the structure of a GPCR from sequence alone. The sequence which you are interested in should be in FASTA format in a file. The FASTA format is as follows:

```
>Here is the pain receptor (mrgA1) sequence.
MGESSTCAGFLALNTSASPTAPTTTNPMDNTIPGGINITILIPNLMIII
FGLVGLTGNGIVFWLLGFCLHRNAFSVYILNLALADFFFLLGHIIDSILLLLNVFYPITF
LLCFYTIMMVLYIAGLSMLSAISTERCLSVLCPIWYHCHRPEHTSTVMCAVIWVLSLLIC
ILNSYFCGFLNTQYKNENGCLALNFFTAAYLMFLFVVLCLSSLALVARLFCGTGQIKLTR
LYVTIILSILVFLLCGLPFGIHWFLLFKIKDDFHVFDLGFYLASVVLTAINSCANPIIYF
FVGSFRHRLKHQTLKMVLQNALQDTPETAKIMVEMSRSKSEP
```

Note: Often the > {name} line is removed during the read in fasta file protocol so before you run clustal 6) c) open blastseq.txt and make sure the first sequence has a > {name} line to it. Otherwise, clustal will fail.

2) To run the GUI, you need to copy the following lines into your .cshrc file:

```
#To run Memb GUI
if ( -d /exec/blast/TMPred_MembStruk_GUI ) set
path=(/exec/blast/TMPred_MembStruk_GUI $path)

if      ( $HOSTTYPE == iris )   then
```

```
        if ( -d /exec/dock-4.0.1/bin ) set path=( /exec/dock-4.0.1/bin
$path )
else if ( $HOSTTYPE == i386-linux ) then
        if ( -d /source/DOCK-4.0/Dock-4.0.1/LinuxDock/Dock.v4/bin ) set
path=( /source/DOCK-4.0/Dock-4.0.1/LinuxDock/Dock.v4/bin $path)
endif
```

Note: The lines that are underlined have no carriage return between them and are all on one line.

3) Then from an empty directory, type TMPred_MembStruk and Enter (then OK to proceed and Cancel to exit). See Section 1.2.

4) Enter the directory in which you are in (full path preferable, i.e., /net/hulk....). See Section 1.3.

5) Enter a project name which will be appended to all outputted filenames and will be used in the MembStruk optimization GUI. See Section 1.3

6) Click on Predict TM Regions

a) Open you file with the sequence of interest. Section 1.4

b) "Perform Blast Search"- will search for homologues to your sequence from the SWISS-PROT or NR database. You should try the swiss-prot search first, as these sequences are of higher quality. Section 1.4.1.

Outputted files: test.out - has Blast output
blastseq.txt - has the sequences kept from the Blast search (sequences with non-zero e values and bit scores above 200). This file may be modified in order to have only sequences of the same subtype or sequences from the same organism

c) "Perform Multiple Sequence Alignment"- will run clustalw to obtain a pairwise alignment of the sequences in blastseq.txt. Section 1.4.1.

Outputted files: blastseq.pir,blastseq.aln - are two files with clustalw output in two formats
temppirfile - is equal to blastseq.pir and it is inputted into TM2ndS TM prediction program

d) Run TMPred (Section 1.4.2)

i) Show coarse results-finds the best window size and baseline for initial coarse predictions which are finely adjusted in the fine results step, should yield output of "found seven helices" if not consult rtraban@ucal.edu. See section 1.4.2.1 for an example.

1) Run default
2) Run improved - should run this updated version
3) Run interactive

Outputted files: which_baseline_used.txt – shows which baseline should be used,WIN12_plot.ps-- plot of hydrophobic profile

ii) Show fine results -- applies capping module to inital predict obatined coarsely. See section 1.4.2.2 for an example.

1)Run default

2) Run improved (again, you should run this)

Outputted files: predictions.txt -- has the ranges and sequence for each helical prediction (the section above has the initial predictions before capping)
data_w*.txt -- have hydrophobic profile data for window sizes indicated,
*get_centers_output -- has output from get_centers programs which finds the hydrophobic center of the helices
HPMCenter.txt -- has the hydrophobic center values
*WIN12_plot.ps -- again,a plot of hydrophobic profile with baseline plotted as well

7) Create model (Section 1.5)

a) Create template --will build the canonical helices with the extended library for the side chains and minimize the potential energy of each separately.

Outputted files: *-Final.pdb -- the bundle of helices ready for optimization

b) skip the optional step

c) Begin MembStruk optimization will begin MembStruk optimization, the next step. See Chapter 2.

Section 1.1.1 – Running TMPred with manually chosen TM's

If the standard method is not producing a valid TM prediction then you can manually run your own predictions.

1) Run through the default method with your best alignment get (blastseq.txt)

2) Look to see if there are data??_[no]basechange.txt files in your directory (should be about 40). If these files are not in the directory then run the interactive TMPred coarse version from the GUI and answer "Y" to the lonest question that asks about plotting data.

3) Now plot these data??_[no]basechange.txt files (only the even numbered ones) against the sequence.txt file to see the alignment. Now I like to sum all the sequences in MS

Excel. I have an awkscript in TMPred/sehall/ datamerge.script and awkdatamerge that will printout a file ready for MS Excel. Copy both files to your working directory and type: datamerge.script.

4) Analyze the graph and determine where the TM regions are located and write down the start and end numbers of each TM region. I like to try and correlate the numbers with the default predictions I got whenever possible.

5) Run TMPred2auto_mod and enter in your numbers - 1 since the program starts it's arrays at zero. You will have to enter you numbers in twice.

6) Run get_centers and enter in the numbers provided in predictions.txt.
a) You will probably want to edit this since the alignment is not the best and that is why the default program couldn't get the correct TM predictions. I like to find the windows that have a leap (~3 difference in the two numbers) and then choose the section that is either the largest (as long as it is not higher than 20, and lower than 9) or the closest to the center (trusting in your analysis rather than the program's).

7) Edit the hel? files for any "-" and delete these. If these appear in the beginning remember that they might affect the HPMCenter predictions from get_centers.

8) Use the GUI and run the create template, starting at step seven in the outline above this one.

Section 1.2 – Opening the TMPred2ndS GUI Window

This GUI program was developed by Rene for the determination of the TransMembrane regions of the membrane bound program. The start window allows the start or end of the GUI.

Start TMPred Window

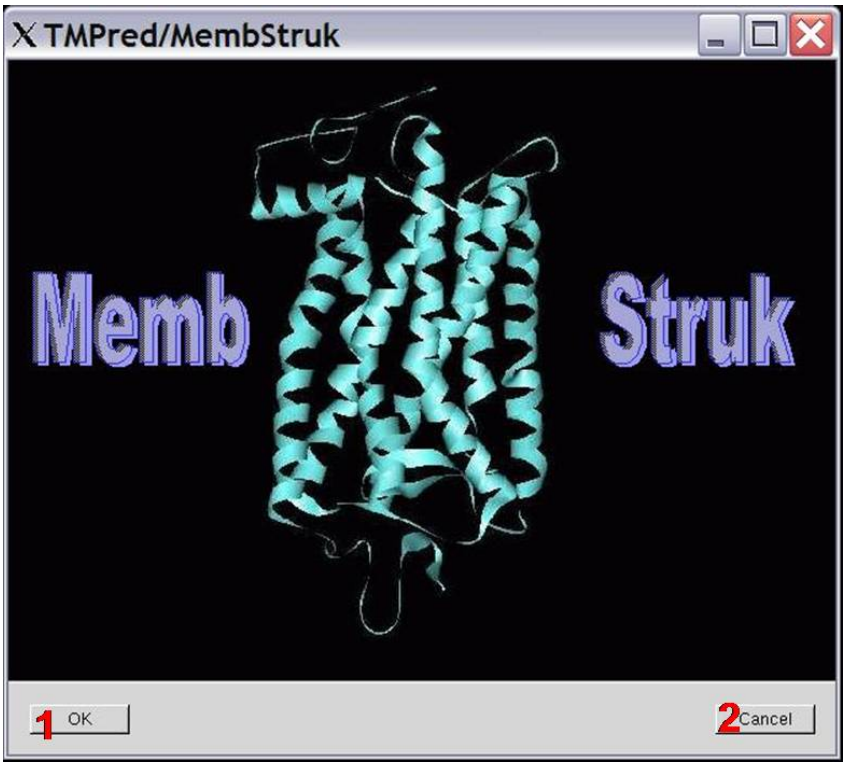


Figure 3 – TMPred Main Start Window

Buttons:

1) Click to begin using the TM prediction protocol. This button will open up the Main TMPred window and the TMPred Output windows. (Section 1.3)

2) Click this button at any time to close the GUI.

Section 1.3 – Main and Output TMPred Windows

These two windows set up the basic script variables and provide a place for notes on new versions of the Gui. The output window will contain information about script changes and Gui updates.

Main TMPred Window

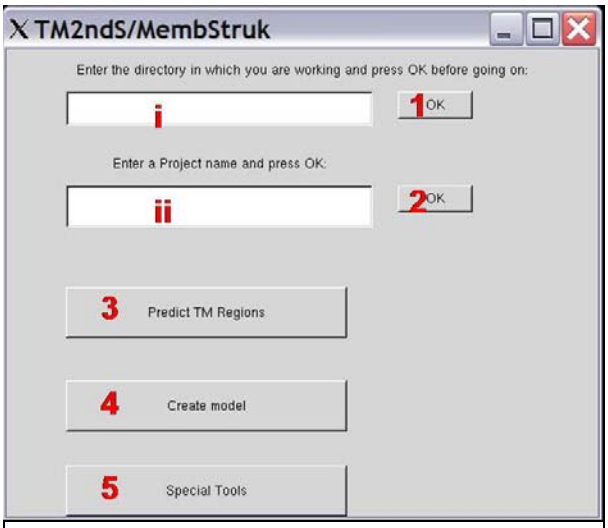


Figure 5 – Main TMPred Input Window

Output TMPred Window

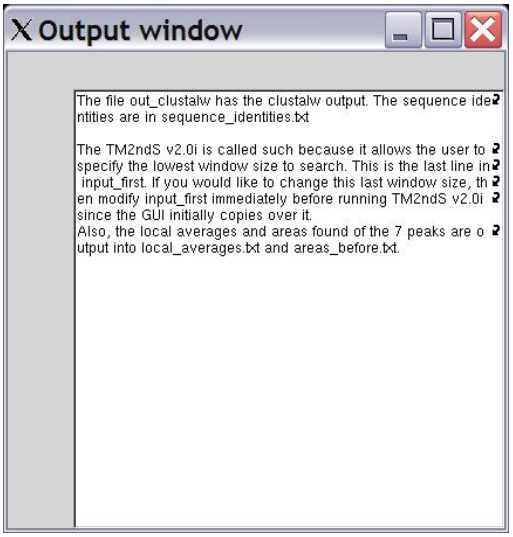


Figure 4 – Main TMPred Output Window

Text Entry Boxes:

i) Type in this box the directory that you want all the working files to be written to.

ii) Type in a prefix name that you want to appear in front of each file produced. It is best to keep this to 8 characters or less.

Note: The Business Edition contains two more text boxes, one to enter in the remote SGI macine to use (origin2 is used for the beta version), and one to enter in the working directory on the SGI machine (for machine that don't have mounted home directories common to all machines).

Buttons:

1) This button must be pressed in order to set the directory variable to what was typed in the text entry box i).

2) This button must be pressed to set the prefix variable to what was type in text entry box ii).

3) This button will bring up the buttons used for the TM predictions. See section 1.4.

4) This button will bring up the buttons used to build the pdb structure. See section 1.5.

5) This button is to be used for access to the individual TM analysis. See section 1.6.

Output Files from Program

Note: it is advisable to run the TM prediction in it's own separate directory away from the main membstruk working directory.

1.xpm	- Picture file for the GUI
PREFIX_temp.txt	- This contains the prefix entered
delH.macro*	- This is a biograf macro for removing hydrogens
dreidii322-quanta.cnv*	- This is a biograf forcefield conversion file
dreidii322-quanta.par*	- This is a biograf forcefield parameters file
ingnu*	- This is a gnuplot macro file for the sequence window plot
ingnu2*	- This is another gnuplot macro more detailed for the sequence window plot
ingnu_lastline	- This is the end of the gnuplot macros
input_first	- This is an input file for a C+ program
inputalign*	- This is an input file for Clustal
macromin.ctl*	- This is a macro file for an mpsim minimization ***(this will need to be edited for systems running in a different directory setup to find the correct forcefield file)***
premac.macro*	- This is a biograf macro file for starting helix building
rhodopsin.fta	- This is my Fasta formatted file containing the rhodopsin sequence
PREFIX.txt	– This file contains the prefix I just entered and is also used with the MembStruk GUI to determine the prefix to use.

Section 1.4 – Predicting TM Regions

This window contains all the buttons needed to do a first basic TM prediction for the desired membrane protein.

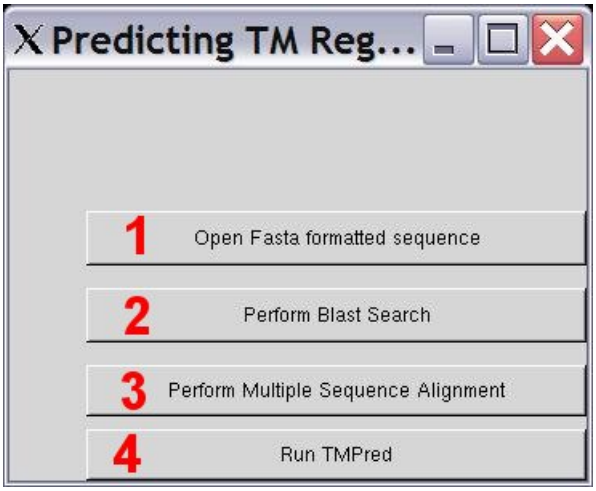


Figure 6 – Predicting TM Regions Window

Buttons:

1) Click this button to choose which fasta formatted file to use for the Blast search. **Note:** This button opens a fileselection window for Beta testers, however in the Business Edition this opens a window were all you do is enter the filename that ***must be in the working directory!***

2) Click this button to perform a Blast Search on the fasta sequence. See section 1.4.1.

3) Click this button to run the Clustal program to perform an alignment.

4) This button will take the alignment and predict TM regions for your protein. See section 1.4.2.

Section 1.4.1 Sequence Alignment

This section will run through the development of an alignment that will be used for the prediction of the TM Regions. The Blast Search will search through the installed library to find all sequences that are within 30% sequence homology of the opened Fasta formatted file.

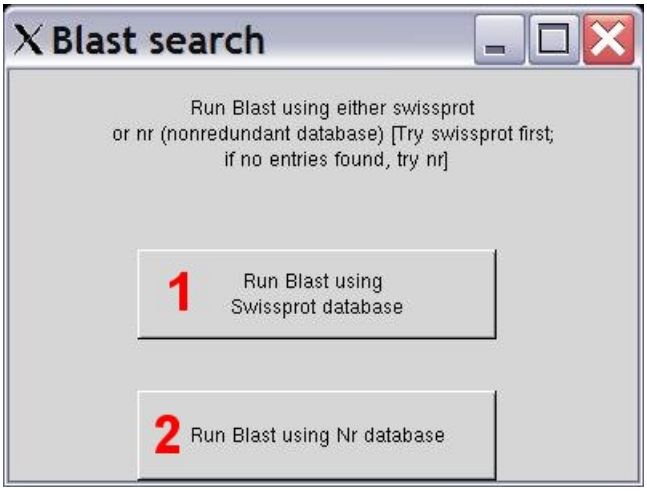


Figure 7 – TMPred Blast Search Window

Buttons:

1) Click to run a blast against the Swissprot database. (Try this one first, if there are less than 20 sequences, then try button 2.)

2) Use this button to run a blast against the Non-Redundant database. (Careful with this database, it has many fragments that are not useful for full protein TM prediction. These fragments are fine to use with specific TM region prediction, see section 1.6)

Output Files from the Program

test.out	- This contains the output from the Blast search
test.txt	- This file contains the fasta sequence used in the Blast search
temp	- The Blast scores and sequence names found are located here
exitit	- This file was empty
temper	- This is a temp file with the full fasta sequences found.
blastseq.txt	- This contains all the sequences found in fasta format ready for Clustalw
y	- This file contains numerical output about the sequences
out_clustalw	- This is the direct output from running clustalw
blastseq.pir	- This is a .pir file containing the alignment of all the sequences
blastseq.aln	- This is an .aln file containing the same alignment information as the .pit file, just in a different format
temppirfile	- This is the temp file created to form the blastseq.pir file, this is needed for running TM2ndS
sequence_identities.txt	- This contains the sequence homologies of the blast sequences. (Use this file to determine which sequences to keep in the blastseq.txt file to have a good distribution of sequences used for alignment.)

Section 1.4.2 TM Prediction (TM2ndS)

This window will run several versions of the TMPred program that will look at the alignments and build a hydrophobicity graph along the alignment to determine where the TM regions are located. This is based on the fact that the helices are more hydrophobic than the rest of the protein since they are immersed in a lipid membrane.

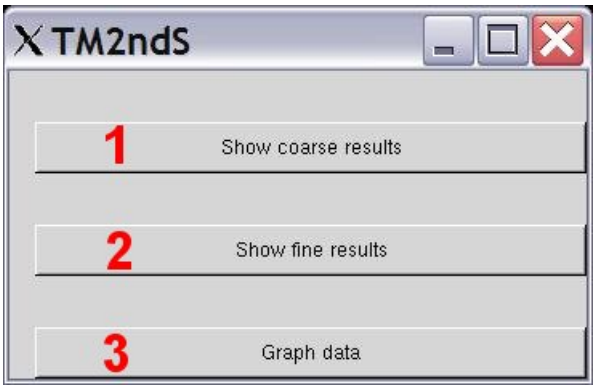


Figure 8 – TM2ndS Window

Buttons:

1) This will run through the TM prediction coarse method to make sure that there are 7 TM regions located in the alignment. See section 1.4.2.1.

2) This will add capping rules to the predictions and prepare the files for model construction. See section 1.4.2.2.

3) This button will display a graph of the hydrophobicity plot using gnuplot.

Section 1.4.2.1 *TM2ndS options and methods*

This window allows for the user to choose which method to use in order to get the best TM prediction. The current method uses buttons 3 and 4 below.

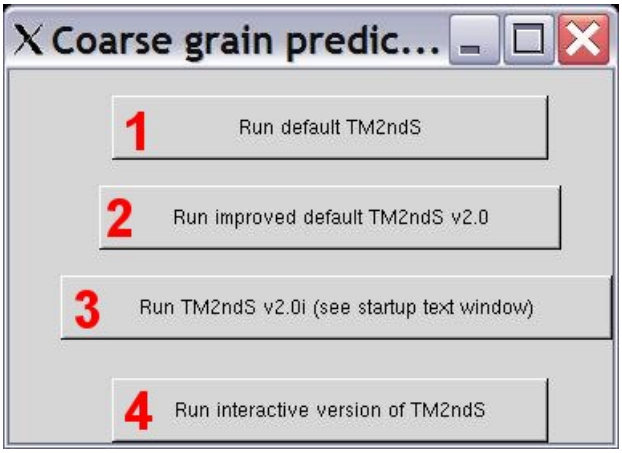


Figure 9 – TMPred Coarse Grain Prediction Window

Buttons:

1) Run the original TMpred program. No longer supported.

2) Run the improved TM2ndS program. Not used in favor of v2.0i.

3) Run the latest version of TM2ndS. This is the program to run for TM predictions on a coarse level.

4) This will run an interactive version that will allow the user to manually obtain all the nessesarry files for the Bipeak analysis program without running a Fine grain TM2ndS.

Output Files from Program

Note: It is not necessary to run TM2ndS Interactive in order to get the files needed for the BiPeak analysis, since running Fine grain TM2ndS will create them. However, the method is left here as an option to those who wish another coarse way of finding initial TM predictions. It is perfectly valid to only run the TM2nd2 v2.0i at this point as described below as part of this example. The example here will look the same except that you do not have to type anything to run the program, and both sets of output files will appear after running TM2ndS v2.0i.

Answer "N" <enter> to all of then until you get to the longest question:

```
"Would you like to have the option of graphing the
hydrophobic profile at each window size (the profiles will
be saves into data12_basechange.txt for window 12 with no
base modification for example; whichever profile you want
to graph needs to be renamed data.txt before using the
graph plot button)?(Pres Y or N)"
```

Type "y" <enter> then type "N" <enter> until the program finishes.

sequence.txt	- This file contains the sequence alignment of all the sequences
data.txt	- This file contains the hydrophobic data from the sequence alignment
predictions.txt	- The predictions from TMPred are located here. (On the 1st try I get only six helices predicted in this file.)
tempresultfile	- Another temp file used in the TMPrediction program.
data##_basechange.txt	- These files contain the graph information for each window size
data##_nobasechange.txt	- These files contain the graph information for each window size

```
foreach i (P28682 P51472)
awk '{if (substr($0,1,1) == ">") {flg = 0}; if
(index($0,"sp|"spnm"|")!=0) {flg = 1}; if (flg == 1) {print
$0}}' spnm=${i} blastseq.txt >> blastseq.tmp
end
```

The above tcsh awk script is all one line (underlined here) and you replace the highlighted region with the sp numbers of the sequences to place in the >> file, in order to choose the sequences you want for the next alignment attempt.

Note: ***You can always run through both programs (interactive and manual) to see if one version picks up something that the other one missed, however run the successful program last.*** Often the sign of a successful alignment is a large (15+) amount of gaps before and after the original sequence with not too many in between. The files produced from pressing button 3 (TMPred v2.0i) in the Coarse grain predictions window are:

areas_before.txt	- Lists the area under the curve for each interval
local_average.txt	- Gives the local average used to get the helix interval
info_pred.txt	- input file for the predictions program
data1.txt	- Same as the data.txt file with the number of lines missing
which_baseline_used.txt	- Gives the baselines used to determine the helical intervals
temp_base_plot	- GnuPlot setup script information
inTMPred2	- Information needed to run the TM2ndS program

Section 1.4.2.2 - *TM2ndS Fine Prediction*

Now that we have obtained a coarse grain TM prediction that has seven helical regions defined, we will apply capping rules and gap penalties to get our final 1st set of TM predictions.

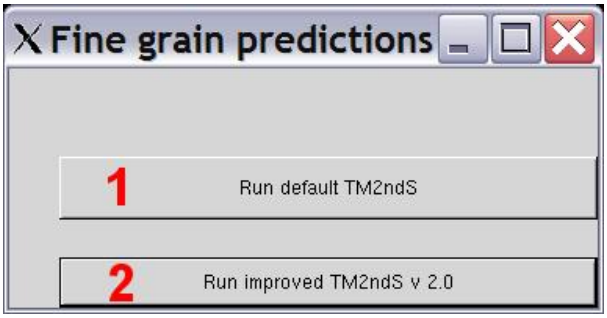


Figure 10 – TMPred Fine Grain Predictions Window

Buttons:

1) This will run the older default TM2ndS program that lacks several features.

2) This button is the one to use to add hydrophobic centers and capping with the latest version of TM2ndS.

Input Files: inTMPred2, temppirfile

Output Files from the Program

The following files are generated along with the data files:

hel[1-7]	- These files contain the predictions per helix and are used to build the final pdb structure.
prediction_numbers.txt	- This contains the input needed for the getcenters program
HPMCenter.txt	- This file contains the HPM centers used for translation
get_centers_output	- This is the output from the getcenters program
ingnu3	- Input file for GnuPlot
ingnu4	- Input file for GnuPlot
data_hyd_cent.txt	- Data output file from the getcenters program
output_TM_core.txt	- Contains the middle 15 residues from the predictions
output_TM_hel_extended.txt	- Gives the TM predictions with the regions being extended by 20 residues
output_TM_hel_extended_size.txt	- Gives the size of the extended regions.
input_interactive	- Gives the input for the interactive program to get all the data files.
data_before_cap.txt	- Contains numbers for the capping rules.
data_after_cap.txt	- Contains the new numbers after capping.
WIN12_plot.ps	- PostScript plot of the Hydrophobicity at Window 12
{Prefix}_predictions.txt	- Same as the predictions.txt saved under the prefix name
{Prefix}_WIN12_plot.ps	- Same as the WIN12_plot.ps file saved under the Prefix name
{Prefix}_get_centers_output	- Same as the get_centers_output file just saved under the Prefix name
{Prefix}_inTMPred2	- The input TMPred file used with this prefix

Section 1.4.3 Interactive Non-GUI TMPred2ndS (When all else fails)

When the automatic version of TM2ndS fails to find seven helical regions or finds seven incorrect regions, then the user must analyze the graph themselves as seen in the supplementary Appendix A. At this point you must have the temppirfile file in your directory to run, the baselines from which_baseline_used.txt, and the seven regions to use for helical determination from either the plot or the predictions.txt file. Now you are ready to start these programs.

This first program will run through a coarse grain prediction of the alignment sequences. If you already have your regions then you can skip this first program. This program is for those who need to try several different baselines in order to get a good coarse grain prediction.

Section 1.4.3.1 - *TMPred1_interactive*

1) Run TMPred1_interactive from the directory with the temppirfile alignment file.

2) Answer No to all questions except, "Would you like to use your own baseline?" Enter the local TM base.

Note: If you liked most of the regions found in the Fine Grain predictions, then use the baseline found in the file inTMPred2. Rene has the following to say about choosing the baseline value to use: "The baseline which should be chosen first is the TM local base because it is the most physically sound one (it is essentially the local average of the TM region, excluding the N and C terminus). If that one is not appropriate, there are two other baseline called the valley base or the average base (average of the valley and local TM base). The baseline is chosen from Fine Grain TM Predictions if 7 helices are obtained with that respective base." This is the sixth line in the inTMPred2 file found in your directory.

3) At the last window size, there are three sections outputted with "The sorted 0 interval is ....". The three will be identical if you choose the baseline. Otherwise they will have the coarse predictions with the local baseline and the global baseline of all the sequence in the last two sections.

Write down the sorted intervals for the coarse predictions (with area greater than 1 except when combining peaks).

Section 1.4.3.2 - *TMPred2auto_mod*

To obtain Fine Grain Predictions with the capping rules applied you must run the TMPred2auto_mod program.

1) Run TMPred2auto_mod from your temppirfile directory.

2) `What is the pir file name:` temppirfile

3) `Would you like to specify a window? (Y or N)` Y – These next inputs are not really important since you will specify your own regions.

4) `what is the window size number?` 12

5) `Give a description:` 1

6) `DO you want to double the base (1), define the base by local averaging (2), or apply no base change (3)?` 2

7) `Do you have your own predictions(Y)?` Y

At this point enter in the regions that you decided to use from the TMPred1_interactive or from analysis of the graphs of the plots. ***Remember to subtract 1 from these numbers since the arrays start counting at 0 in this program.***

8) Do you have your own predictions(Y)? N – Unless you want the helices to be ended at any gap found inside the predicted region.

Section 1.4.3.3 – *get_centers*

Now that we have a final fine grain prediction for our protein, we still need to find the hydrophobic centers for translation. The program get_centers will do this for the user automatically. In order to use this program you need to have the data##.nobasechange.txt files in your directory (obtained from answering Y to the graphing question in the interactive version, or having run Fine Grain Predictions from the GUI in section 1.4.2.2). Also, for this program you do not have to subtract 1 from your region prediction numbers. Other than this information the questions are fairly straightforward. Do expect

to have another console window open to look for the information requested in other files, such as the size of the sequence file (how many lines it contains). Once you have entered all the nobasechange.txt files, you will want to save the output given to another file since this information is used for bipeak analysis.

Now to understand the output you have:

```
[10 10 10 14 13 15 14 15 3 18
 11 10 9 12 11 15 16 17 16 16
 16 15 16 17 17 12 11 12 13 11
 14 13 12 11 16 14 14 15 16 17
 10 11 12 10 15 15 13 17 16 15
 16 17 17 16 13 14 13 15 16 17
 9 13 11 12 12 11 10 8 8 7]
```

This section has the hydrophobic centers calculated at each window size 12 to 30 for every helix. The helices are in rows and windows in columns. The program will choose those window sizes where the center prediction is stable (not much variation) for all helices as compared to values at window size 20.

Stable means the value does not deviate by 5 or more from the value at window size 20. For example, the value of 17 for helix 3 at window size 20 is 5 away from the value at window size 22.

```
[The flags are :0110100000]
```

This tells you which ranges of window sizes were found to be stable. Typically it is nice to have an island about window size 20 like 0001111000, but as in the case above, the helix 3 was too unstable to allow this choice.

In cases like this, the user must look at the raw data and ascertain which regions

are stable excluding a certain helix and average those values at those window sizes to obtain the hydrophobic center which should be entered in HPMCenter.txt.

In all cases, the calculation will be done automatically, but the user can look at the output in *get_centers_output* to figure out if the center predictions was stable. Center predictions are generally better if one uses more sequences in the alignment.

Section 1.4.4 – BiPeak Analysis for Helical Translation (Experimental, not supported)

This section will describe how to use a bipeak analysis program to get an increased accuracy in your HPM centers prediction that was obtained through get_centers (section 1.4.3.3) or from Fine Grain Predictions (section 1.4.2.2). The program can be found as a Linux executable in the:

$membstruk/bin/

directory under the name: BiPeakanalysis.exe.

When running this program you will want to look for the peak that is closest to the center and near to the original get_centers prediction.

1) On a Linux machine run the BiPeakanalysis.exe program in your final predictions directory.

2) `Print the input data file name ("END" to stop)` Enter in the data##.nobasechange.txt file names that you want to use for analysis. Default is the windows chosen by the get_centers program. Type the filename <enter> then another filename <enter> until all files have been entered then type “END” <enter>.

3) `Enter in the ranges to use (according to seq. align) Example > 24 46 (0 0 to stop)` Enter the ranges here from the predictions.txt file, again no need to subtract 1 from the range numbers. After each range is entered 1 to 2 peaks will be given and this information should be saved for later reference.

4) Enter 0 0 to stop the program after you have entered in all 7 ranges. Now you need to choose which peaks to use from each region.

5) Replace the HPMCenters.txt contents with the new centers obtained from the BiPeak analysis program. You are now ready to continue building the structure.

Section 1.5 – Creating the PDB Structure

This window allows for the creation of a pdb structure containing the TM regions of your sequence. This is the final step in the TMPred Method.

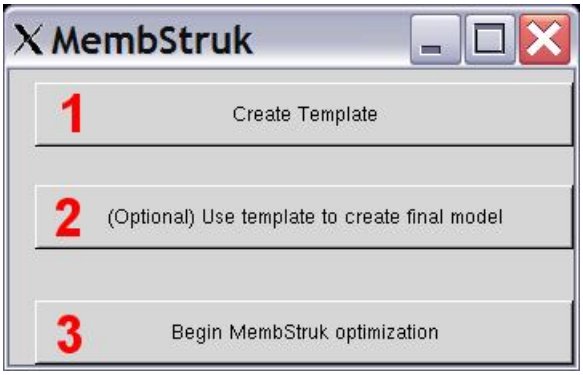


Figure 11 - Create Template Window

Buttons:
1) Create Template – This button will look for the files hel1 through hel7 and use those files to create 7 pdb helices in Biograf. These files are built with an extended amino acid library (in Peter Fred's directory), and then minimized in mpsim. These files are then matched onto a bundle alignment derived from frog rhodopsin.

Input files needed:
hel1 – 7

Output files:

hel?.pdb	- These are the pdb helix structure files created from Biograf. These files are created during the Create Template button 1 in the Create Template Window (section 1.5).
temphelix.bgf	- This is a temp file to save the bgf conversion of the helix pdb file. This file is part of button 1 in the Create Template Window (section 1.5).
logfile.macro	- This is a file created by Biograf that contains a macro of all commands used in the last execution of Biograf in the current directory.
tmp2.bgf	- A temp file for holding a helix structure during conversion from pdb to bgf in the create template function. See section 1.5.
min.out	- This contains the output from mpsim of the last minimization run on the last .bgf helix file. Created from button 1 in the Create Template window (section 1.5).
temp2.out	- This is the output generated from biograf during the conversion process. Created from button 1 in the Create Template window (section 1.5).

hel?-bgf.pdb	- Saved files after the bgf conversion to pdb. Created from button 1 in the Create Template window (section 1.5).
hel?-final.pdb	- These are the final minimized helix structure files that are used in the alignment of the bundle. Created from button 1 in the Create Template window (section 1.5).
original.pdb	- This is a structure pdb file used in the creation of the final structure bundle file. Created from button 1 in the Create Template window (section 1.5).
template.pdb	- This is a structure pdb file used in the creation of the final structure bundle file. Created from button 1 in the Create Template window (section 1.5).
final.pdb	- This is a structure pdb file used in the creation of the final structure bundle file. Created from button 1 in the Create Template window (section 1.5).
template-final.pdb	- This is a structure pdb file used in the creation of the final structure bundle file. Created from button 1 in the Create Template window (section 1.5).
bundle.pdb	- This is a structure pdb file used in the creation of the final structure bundle file. Created from button 1 in the Create Template window (section 1.5).
{Prefix}-Final.pdb	- This file is the final pdb bundle structure that will be used for the rest of the MembStruk protocol. Created from button 1 in the Create Template window (section 1.5).

2) This button will use user defined helix files to align the final minimized TMPred helix files to in order to get the helix orientations according to the template supplied. This feature is currently being tested.

3) This button will open the MemStruck main window as seen in Chapter 0. This button is not necessary to press if you have started the TMPred GUI through the MembStruk main Window by pressing button. All you need to do is press the Cancel button located in the TMPred main window (Chapter 1) and all the TMPred windows will close.

Section 1.6 – Special Tools

For the advanced user, there are buttons for tools commonly used when doing an in-depth analysis of the TM prediction. Most of these features are unsupported and in an experimental testing phase. They are listed here for reference and an example of how to use the most commonly requested features. All the special tool buttons are located in window affectionately called by its creator “Window 9”.

Figure 12 - Special Tools window

Buttons:

1) This will extract sequences from your blastseq.txt file. This will not run on Borg, and will require that the files blastseq.txt and your fasta file be present. After looking at your blastseq.txt file it will ask:
`Enter the name of the single sequence file used as BLAST input:` Enter the name of the fasta formatted file used in the blast search.
This program will then create a blastseq.txt that contains all sequences that were generated from blast. This program is good to use if you need to enrich your data set and haven't obtained enough sequences from the Swissprot and NR databases.

2) Perform a special alignment with high local gap penalties. The program gives the following information:

```
This will perform an alignment with high gap penalties for
specified regions. The file profile_template has been
copied to this directory. Also the sequence should be
changed to the TM helix extended sequence [TMhelix
prediction +-10] as in output_TM_hel_extended.txt. You need
to change the TM core [sequence which would have high gap
penalties, +- 10 from the hydrophobic center) range numbers
(the core seqs are in output_TM_core.txt). The sequence in
the profile (be it one or multiple extended TM helix
```

regions) will be aligned to the sequences in the blastseq.txt file in this directory. Typically, the sequences in blastseq.txt would be homologues in the TM core of the query sequence.
The following new files are created along with temppirfile, sequence_identities.txt, and out_clustalw:

profile_template	- This file contains your sequence and can be modified as described in the header of this file to force the high gap alignment program to extend TM regions so specified. This file is used exclusively for the Special Tools Button 2 program. See section 1.6.
inputalign2	- This input file is used to run a special alignment with high gap penalties. This file is used exclusively for the Special Tools Button 2 program. See section 1.6.
blastseq_TMcore.dnd	- This file is the output of a special high gaps penalty program of the sequences alignment in dnd format. This file is used exclusively for the Special Tools Button 2 program. See section 1.6.
blastseq_TMcore.ala	- This file is the output of a special high gaps penalty program of the sequences alignment in ala format. This file is used exclusively for the Special Tools Button 2 program. See section 1.6.
blastseq_TMcore.pir	- This file is the output of a special high gaps penalty program of the sequences alignment in pir format. This file is used exclusively for the Special Tools Button 2 program. See section 1.6.

3) Obtain certain selected sequences from the Blast search. This program is the same as that used in button 1 of the Special Tools window, except that is asks for the user input: What is the minimum bit score value: 200 [this is the default used in the blast search on button 1 or 2 of section 1.4.1. You set this to a lower value to include more sequences and a higher value to exclude sequences]
Blastseq.txt is created with the new sequences from your bit score choice.

4) Align you helices to your own template. This will take your files after template creation and align the hel?.pdb files to the files named ol?.pdb so that you can set in your directory any 7 ol?.pdb files and create a template based on those seven pdb files. This is useful for creating templates from other predicted models and new crystal structures.

5) This will perform an iterative TM prediction on each of the seven regions found in the fine grain predictions (section 1.4.2.2). This is described in more detail in section 1.6.1 and in the examples 1 and 2 found below.

6) TM core gap checking and gap capping. This program will check for gaps in the TM core regions and output a file called to_exclude.txt with those sequences with gaps. You

may later want to delete those sequences causing the gaps and re-align. In addition, this program will check for gaps beyond the TM core region and give an alternate capping of these helices which may be compared to the TM2ndS capping. The file max_ranges.txt would have these values, and 0 for cases where no gaps were found beyond 20 of the hydrophobic centers.

`Would you like to check on all helices or just one. Type All if you would like all, or type the helix number you would like to be analyzed:` All

This program is useful to determine out of a large number of sequences which ones are causing problems in the TM prediction. This program can only be run after a Fine Grain prediction has been successful, or create the needed output_TM_core.txt (see section 1.6.1) and then run this to further refine your blast sequences.

New files created along with sequence.txt and data.txt:

to_exclude.txt	- This contains a list of sequences to exclude from your blastseq.txt file in order to get rid of gaps occurring in the core TM regions. This file is exclusive to the Special Tools button 6 program. Section 1.6.
max_ranges.txt	- This file lists the limits on the TM predicted regions from gaps in the sequence, should high gap penalty be used. This file is exclusive to the Special Tools button 6 program. Section 1.6.

Section 1.6.1 – Iterative TM Prediction

This section describes in detail how to use the Iterative TM Prediction program that is currently in a beta testing state. Once button 5 of the Special Tools Window is pressed the following appears:

`1) What is the name of the query sequence file?` Enter the name of the Fasta formatted sequence file that was used for the blast search and opened in section 1.4.1 button 1.

`2) Make sure you have a file output_TM_core.txt with the core regions from the first round of TM helical predictions. If yes type Y, if no, type N and the program will exit so that you may prepare this file:` Y - If at this point you do not have the needed file, in the case of example 1 since we used the interactive version without the GUI we need to create one. The easiest way to do this is to open up the predictions.txt file and copy each of the sequence lines into this file, starting with TM region 1. Once this is done you need to limit the sequence to 20, so find your hydrophobic center number and make that residue the middle residue of the 20. If there are not enough sequences to make the hydrophobic center in the middle get it as close as

possible. The only time you should add residues to the sequence that were not found in the original prediction is when the prediction is less than 20, in that case add residues equally on both ends until the needed twenty has been reached.

3) Would you like to use the swissprot (swissprot) or non-redundant (nr) database for this iterative search on TM0? Choose the database that you used for the initial TM predictions.

4) Would you like to filter low complexity regions (T or F with T recommended first)? T - Usually you will want to filter unless you believe that there is not enough diversity in the database you have chosen. However, it is still recommended that you run through the program once choosing T first and then compare it to the output obtained from running through the program choosing F.

The program at this point will run a blast search on the first TM region sequence as found in the output_TM_core.txt file. Then it will run a clustalW alignment.

5) On what Linux computer do you want to run TM2ndS? Enter in the name of an available linux computer to run the program.

Now the computer will run TM2ndS and predict the TM regions for all seven regions, however the only region of interest is the one from region that is currently being run.

6) Now steps 3, 4, and 5 will be repeated for all seven regions before the program finishes.

The following files were created along with all the normal files from a fine grain prediction (***However, these files only reflect the final region's prediction. It would be better to run the non-GUI TM2ndS with the regions taken from each individual TM prediction***):

output_TM_core.txt	- This was altered to tell which core sequences had gaps
output_TM_core1.txt	- This is the backup copy of the original output_TM_core.txt file after being run through the iterative TM prediction. This file is created after running Special Tools Button 5, section 1.6.1.
HPMCenter?.txt	- These files represent the get_centers that were run for each regions prediction. These files are created after running Special Tools Button 5, section 1.6.1.
blastseq?.txt	- These files contain the blast sequences used for each region. These files are created after running Special Tools Button 5, section 1.6.1.
get_centers_output?	- These files contain the get_centers output for each region. These files are created after running Special Tools Button 5, section 1.6.1.

out_clustalw?	- ClustalW output for each region. These files are created after running Special Tools Button 5, Section 1.6.1.
predictions?.txt	- These files contain the predictions found for each region tested. These are the file to use for the final TM prediction. These files are created after running Special Tools Button 5, Section 1.6.1.
sequence_identities?.txt	- These files contain the sequence homology information obtained from clustalW for each region. These files are created after running Special Tools Button 5, Section 1.6.1.
{prefix}.txt	- This contains the fasta sequence of the original protein to be studied. This file is created after running the Special Tools Button 5, Section 1.6.1.
{prefix}?.out	- These files contain the output from running the complete TM Prediction cycle on the ? region. These files are created after running Special Tools Button 5, Section 1.6.1.

Chapter 2 – TM Predictions text file

This window allows the user to create a more readable form of the predictions.txt file that they can use to view the TM predicted regions. When button 2 of the Main MembStruk window is pressed (Chapter 0) this window appears:

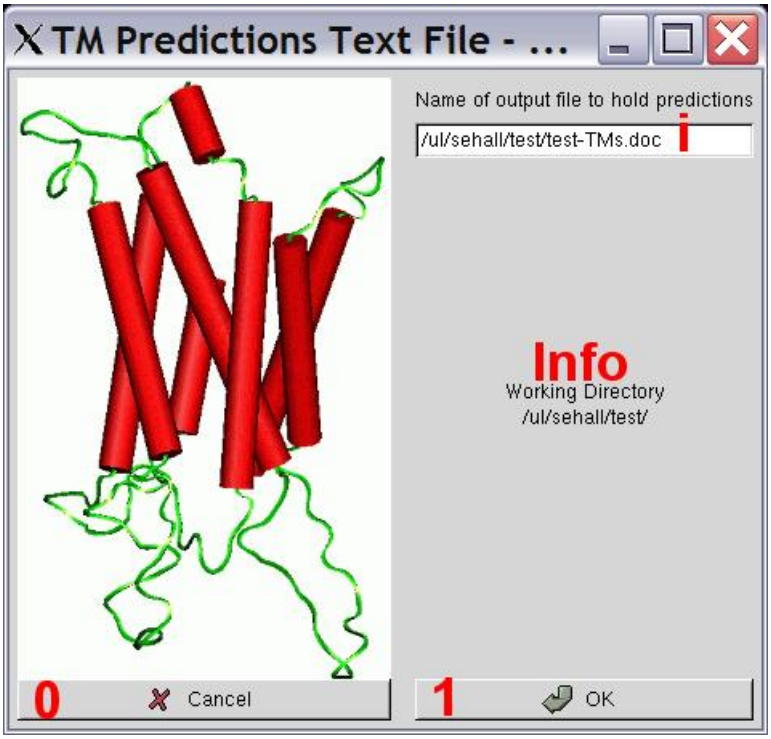


Figure 13 - TM Predictions Text Window

Text Boxes:

i) Enter in here the name that you want the TM prediction information to be saved into. This will overwrite an existing file. The default is: {Prefix}-TMs.doc

Information Section:

The information section contains the current working directory as defined in the Main MembStruk GUI window. This is the directory that needs to contain the input files: hel1-7 and blastseq.txt.

Buttons:

0) Press this button to cancel out of the window and return to the MembStruk Main GUI window.

1) Press this button to run the TM predictions text program. See section 2.1 for details.

Section 2.1 – TM Predictions Text program

This is a simple awk program that will create a file that will contain the TM predictions in an understandable format. This program is run completely from the GUI as a system call to the awk program. It uses the file:

$membstruk/loops/whatif/awkwhatif

The system call looks something like this:

```
awk –f $membstruk/loops/whatif/awkwhatif hel1 hel2 hel3
hel4 hel5 hel6 hel7 blastseq.txt > {file name chosen}
```

The above is all on one line and created the file chosen to contain the prediction information.

This small program was originally created to help organize the information needed in creating loops using whatif.com (see chapter 11). This turned out to be popular in viewing the initial TM predictions and was created as part of the main GUI. An in depth look at the output can be seen in example 1 below.

The output from this program gives us a very good picture of how the TM predictions look. Some things to check for are:

1) All loops must be 6 residues or longer. The TM2ndS will always check that this is the case, but often it will count one gap as being a residue and thus reduce the number to 5. This is the case in loop 1 (or IC 1), so we need to keep an eye on this loop and we can assume that the predictions of TM1 and TM2 might be too long at the ends connecting LP 1.

2) Look for TM regions that are smaller than 20 residues and larger than 36 residues. This size restriction is due to the general thickness of the cell membrane and that the tilt of the helix is generally close to being perpendicular to the membrane. Helices 1, 2, 4, 5, and 6 all tend to be in the mid. twenties, which suggests that helix 1 and 2 might be too long. Also since helix 3 is often tilted more it tends to be in the low thirties. The last exception is helix 7 which tends to be small around twenty residues (and also tends to be bent more than the other helices.)

3) Conserved CYS bridges can be seen in this format. There are two CYS bridges that seem to be conserved across many GPCR's: 1) The CYS bridge between loop 4 (EC 2) and the extracellular tip of TM 3. (This bridge seems to control the movement of the EC 2 loop in the active and in-active protein forms.), 2) The CYS bridge between the beginning of loop 4 and the end of loop 4. (This controls the overall shape of the EC 2 loop.) In our structure there are only two cystines in loop 4 making the second cys bridge unavailable. But the 1st type of conserved bridge is available and we will have to do an alignment of our sequence to find which of the two cystines is forming the bridge with TM 3. The cystine residues shaded in the above output are identified as forming a cystine bridge.

Input Files needed:

hel1 – hel7	- These files must be in the working directory and contain no gaps "-".
blastseq.txt	- This file needs to contain only the original fasta sequence file in the first lines of the file.

Output Files:

{Prefix}-TMs.doc	- This is the default name (any can be chosen) to contain the information about the TM prediction.
PREFIX.txt	- This file is written with the current working prefix once the ok button is pressed.

Running Time for TM Predictions Text program:

One minute.

Chapter 3 – Rotation and Translation of the Template

This window will access the hcenterTR.script C shell script that will translate the helices by their hydrophobic centers and then rotate the helical files by hydrophobic moments.

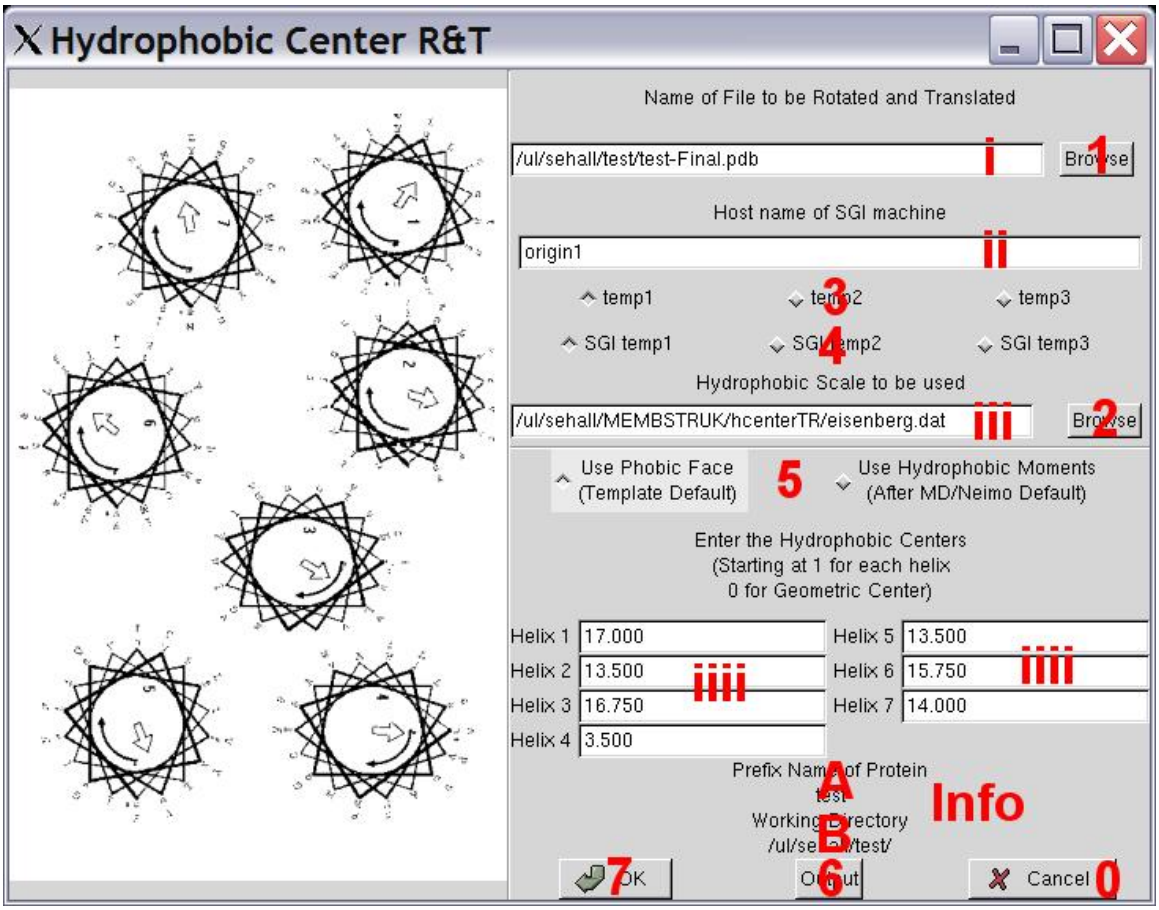


Figure 14 - Template Rotation and Translation Window

Text Boxes:

i) Enter in this textbox the name of the final pdb structure file obtained from the TM2ndS method. The default file name entered is {Prefix}-Final.pdb.

ii) This textbox should contain the name of the SGI machine that will run Biograf remotely. You must be able to ssh onto this machine via a C-Shell (you must have this machine located in your home directory .ssh/ in the known_hosts file). It is best if you try to ssh into this computer before you run this script.

iii) Enter here the text file that contains the hydrophobic index that you wish to use for determination of the hydrophobic moments. The default is the eisenberg index located in the file:

```
$membstruk/hcenterTR/eisenberg.dat
```

another file provides the White scale:

```
$membstruk/hcenterTR/White.dat
```

both give a average error of ~50 degrees per helix when tested on perfect rhodopsin helices taken from the crystal structure 1b.pdb. However, only the eisenberg index has been tested on the 1b.pdb structure with the improved BiPeak hydrophobic centers (see example 3). Using the improved centers the eisenberg index gives an average error of ~30 degrees on the rotation of the helices. The format of this file is as follows:

```
REMARK average of 5 scales, by Eisenberg SCALE
REMARK pg. 637 Pred. of Prot. Struc. ....
ALA a      0.62
ARG r     -2.53
ASN n     -0.78
ASP d     -0.90
CYS c      0.29
GLN q     -0.85
GLU e     -0.74
GLY g      0.48
HIS h     -0.40
ILE i      1.38
LEU l      1.06
LYS k     -1.50
MET m      0.64
PHE f      1.19
PRO p      0.12
SER s     -0.18
THR t     -0.05
TRP w      0.81
TYR y      0.26
VAL v      1.08
END
```

To change this file the first and last columns must be edited. The comparison program only reads the first and last columns to get the information needed to calculate the hydrophobic moment. Currently, the reader for this file supports 44 different residue names, so you can specify up to 44 different residue names to create your unique hydrophobic scale. The first column in this file is matched with the 4th column in the BGF file to assign hydrophobicity values and the reader is looking in the second column a double real variable.

iiii) These seven textboxes are used to contain the hydrophobic centers assigned in the TM2ndS program. The default is to look for the file HPMCenter.txt and read the values in the second column in order 1 to 7 into these textboxes. If you are going to use the geometric centers of the helices (MembStruk method 3.0 or earlier) then enter 0.0 into each textbox. These centers will be sent to the translational program:

transhelix-res.exe - This is a Fortran 77 program that will take a seven helical BGF file and translate each helix from it's geometric center to a specified residue center. This is used in the hcenterTR.script to translate the template onto it's hydrophobic centers.

Information Section:

A) This will contain the prefix that you entered in the textbox in the main gui window.

B) The information section contains the current working directory as defined in the Main MembStruk GUI window. This is the directory that will have the final output files.

Buttons:

1) This button will open up a file selection window to find the final pdb file to use for rotation and translation.

2) This button will open a file selection window to select a data file to contain the hydrophobic scale to use in the comparison program.

3) These three radio buttons are used to determine which environmental variable to use to define the Linux temp directory. Choosing temp 1 will use $memtmp1 up to temp 3 will use $memtmp3. Only one can be chosen and the default is temp 1. This variable will be used to create a temporary directory ($memtmp?/{Prefix}{unique}) to store files while in use for the hcenterTR script.

4) These three radio buttons are used to determine which environmental variable to use to define the SGI temp directory. Choosing SGI temp 1 will use $memsgi1 up to temp 3 will use $memsgi3. Only one can be chosen and the default is SGI temp 1. This variable will be used to create a temporary directory on the remote SGI machine ($memsgi?/{Prefix}{unique}) to store files while in use for the hcenterTR script.

5) These two radio button decide which method to use in finding the hydrophobic moment to rotate the helices by.

If you choose "Use Phobic Face" then the helices will be treated as canonical and have all C-alpha carbon positions (the middle 15 residues around the hydrophobic center) projected onto the plane that intersects the hydrophobic center and aligns the smallest moment of inertial along the Z-axis (effectively placing the main length of the helix along the Z-axis). Then all projected C-alpha positions will be assigned their hydrophobic

scalar according to the hydrophobicity scale chosen and then the program will determine the largest gap of the helix that does not face other helices and look the the largest number (summing the C-alpha numbers over that gap). The middle of the gap containing the largest sum of all C-alphas will become the hydrophobic moment and the helix will be rotated to move this position to 180 degrees from the center of the protein.

If you choose "Use Hydrophobic Moments", then each helix has it's hydrophobic moment centered at it's hydrophobic center and is calculated the normal way through vector addition. The calculation of the hydrophobic moment uses only the middle 15 residues around the hydrophobic center of each helix. Then these hydrophobic moments are projected onto the plane of intersection of the hydrophobic centers (see MembComp manual). The degree of rotation needed to point the end of hydrophobic vector 180 degrees away from the center of the protein is used for the individual rotation of the helices.

Since the template helices are using extended side chains and are perfectly canonical, the default is set to "Use Phobic Face".

6) This button will bring up the Output Window. See section 3.2 for more details.

7) This will start the hcenterTR.script running. See section 3.1 for more details.

Section 3.1 – Template Translation and Rotation Program

This is the first of many C-shells that make up the MembStruk methodology. This script will take the pdb template file and translate it according to the hydrophobic centers and then rotate it according to the hydrophobicity scale chosen. This script can run indepentantly of the MembStruk GUI. The main script run from the GUI is called: $membstruk/hcenterTR/hcenterTR.script

This script run using the following command:

```
hcenter.script {PDB template file} {protein prefix} {Hydo.
Scale} {0 = PFace, 1 = Hydro} {SGI remote machine name for
Biograf} { Linux environmental variable [1/2/3]} {SGI temp
enrivonmental variable [1/2/3]}
```

The pdb template file has a specific format that is required and should be created from the TMPred method. The SGI remote machine name should have Biograf installed and be able to handle SSH logins from a C-shell without needing the user's password. The

Linux and SGI environmental variable are just numbers 1-3 to pick the corresponding environmental variable to use.

This script will take the PDB template file and convert it into a BGF file for the translation, comparison, and rotational programs. It will be translated, analyzed, rotated, analyzed again, and then finally converted back into a usable PDB template file. This script is used to get the starting alignments of each helix according to hydrophobicity (that the main vector should point towards the membrane) and hydrophobic centers (the idea that hydrophobicity acts like an object floating in a liquid with the center at the middle of the bilipid layer).

Dependant Files for hcenterTR.script:

FF/Biograf.par	compare/compareL.exe
FF/Biograf.cnv	compare/HsprotL.exe
bin/transhelix-res.exe	

Input Files for hcenterTR:

{Prefix}-Final.pdb	- This is the template pdb file obtained from TM2ndS.
HPMCenter.txt	- This file contains the hydrophobic centers defined from TM2ndS.
eisenberg.dat	- This file or the once in it's place contains the hydrophobic data used for rotation.

Output Files from hcenterTR:

{Prefix}-hcenterTRv4.10	- This file is a PBS script used to run hcenterTR on a linux cluster.
out-{Prefix}.rot	- This file contains all the output from running the hcenterTR script.
{Prefix}-rot.pdb	- This file contains the new template that has been translated and rotated.
{Prefix}.beforeR.txt	- This file contains the analysis of the pdb file after translation and before rotation. See examples below for more details.
{Prefix}.beforeR.ps	- This PostScript file contains the visual plot of the data contained in {Prefix}.beforeR.txt.
{Prefix}.afterR.txt	- This file contains the analysis of the pdb template file after translation and rotation.

{Prefix}.afterR.ps	- This PostScript file contains the visual plot of the data contained in the {Prefix}.afterR.txt file.
PREFIX.TXT	- This file is rewritten with the prefix used for running hcenterTR.script.

Running Time of the hcenter.script:
5 to 10 minutes.

Section 3.2 – The Output Window for hcenterTR and hcenterTR-BGF

This GUI window is used for both hcenterTR (chapter 3) and for hcenterTR-BGF (chapter 6). This is a convenient way to view the visual data from the scripts and make analysis of how to proceed with the methods.

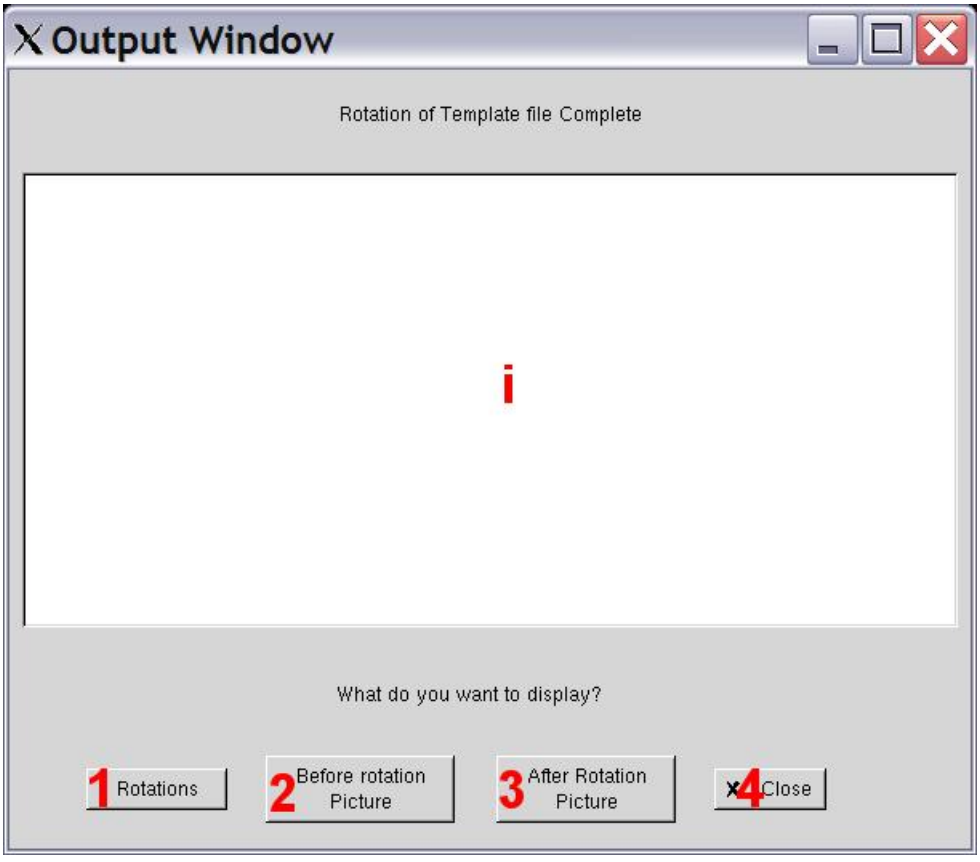


Figure 15 - HcenterTR Output Window

Text Boxes:
i) This is a non-inteactive text box that is use for displaying of text from the output files.

Buttons:
1) This button will display in text box i the contents from the out-{Prefix}.rot or out-{Prefix}.Rot for the hcenterTR-bgf script.

2) This button will open up xv and display the postscript file after translations and before rotations, plus display in the text box the contents of the analysis data file.

3) This button will open up xv and display the postscript file after translations and after rotations, then display the data from the analysis into the text box.

4) This button will close this window and return you to the previous window.

Chapter 4 – Helical Dynamics

This window will allow the user to set up the template file into a bgf file with helices that have been relaxed with dynamics simulations.

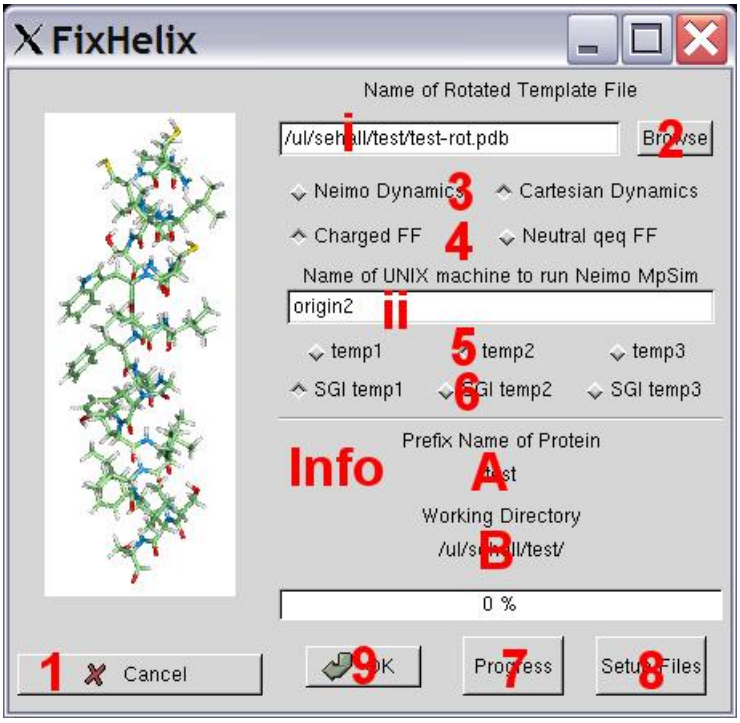


Figure 16 - Fixhelix Window

Text Boxes:
i) Enter here the pdb file that has been rotated and translated. The default is {Prefix}-rot.pdb.

ii) This textbox should contain the name of the SGI machine that will run Biograf remotely. You must be able to ssh onto this machine via a C-Shell (you must have this machine located in your home directory .ssh/ in the known_hosts file). It is best if you try to ssh into this computer before you run this script.

Information Section:
A) This will contain the prefix that you entered in the textbox in the main gui window.

B) The information section contains the current working directory as defined in the Main MembStruk GUI window. This is the directory that will have the final output files.

Buttons:
1) Press this button to close this window and return to the previous window.

2) This button will bring up a file selection window to choose the file to place in text box i.

3) These two radio buttons allow you to choose what kind of dynamics simulation to run on the helices. The Neimo dynamics button will run the Neimo simulation method on the helices (this is owned by NASA and not supported in the Business Edition). The default is to have the Dynamics simulation selected which will run through a dynamics simulation at a temp of 300 Kelvin.

4) These two radio check buttons are used to determine what kind of charges you will use on the charged residues in your helices. By default the charged FF is selected that will set all positive residues to a +1.00 net charge and all negative residues to a net -1.00 charge. The neutral qeq FF button when selected will convert all charged residues to a neutral net charge as predetermined by QeQ.

5) These three radio buttons are used to determine which environmental variable to use to define the Linux temp directory. Choosing temp 1 will use $memtmp1 up to temp 3 will use $memtmp3. Only one can be chosen and the default is temp 1. This variable will be used to create a temporary directory ($memtmp?/{Prefix}{unique}) to store files while in use for the script.

6) These three radio buttons are used to determine which environmental variable to use to define the SGI temp directory. Choosing SGI temp 1 will use $memsgi1 up to temp 3 will use $memsgi3. Only one can be chosen and the default is SGI temp 1. This variable will be used to create a temporary directory on the remote SGI machine ($memsgi?/{Prefix}{unique}) to store files while in use for the script.

7) This button is no longer supported. It will still display the progress of the fixhelix script as long as the working directory is set to the unique working temp directory of the running job that progress is to be checked.

8) This button will place in your working directory the ctl file that will be used for running the dynamics simulation. This way you can edit it to your needs and fixhelix will use this file instead of the default one. This file is called {Prefix}.ctl.

9) Click this button to run the C-shell script fixhelix. See section 4.1 for more details.

Section 4.1 – The Fixhelix Program

This program called fixhelix will take a template pdb file and first break it up into seven helical pdb files. Then each file will be correctly converted into a bgf file with Dreiding 322 charges and have counter ions (Na and Cl) placed in the correct positions to offset the charged residues that are not forming salt bridges. These counter ions are inflated to represent the size of these ions as if surrounded by water. Then each helix is fully minimized and has a dynamics simulation run on them for 100000 steps. After dynamics, the structure with the lowest energy after 50000 steps (going in steps of 1000) is minimized again. Then all seven structures are bundled together by matching the structures back onto the original positions and run through the scwrl program. After scrwl, the bundle is minimized one last time and outputted.

This script is unique in that it is really four scripts and the script to use is determined by your choices of radio buttons 3 and 4. The four scripts run the same but use different ctl files and forcefields for the main dynamics simulations:

Neimo, Charged:	fixhelixL.script
Neimo, QeQ:	fixhelixL.NQEQ
Dynamics, Charged [default]:	fixhelixL.Dynamics
Dynamics, QeQ:	fixhelixL.DYNqeq

The command to run these scripts is the same [fixhelixL.script used as an example]:

```
fixhelixL.script {template.pdb} {protein prefix} {SGI
machine to run Biograf} {Linux environmental variable temp
# [1/2/3]} {SGI environmental variable temp # [1/2/3]}
```

Dependant files for the Program:

FF/Biograf-qeq.cnv FF/Biograf.cnv

FF/Biograf.par
FF/Mpsim.par
bin/bgf2fasta.pl
bin/bgfscwrl
bin/change_ntct_crg.pl
bin/helixcm.exe
bin/mpsim
bin/nacl-appendL
bin/scwrl
bin/sparse_pdbL
fixhelix/NEIMOfnd.awk
fixhelix/neimo_Dyn.ctl
fixhelix/neimo_Dynqeq.ctl
fixhelix/neimo_NQEQ.ctl
fixhelix/neimo_orig.ctl

Input Files Needed:

{Prefix}-rot.pdb - This is the output of the hcenterTR program.

Output Files from the Program:

{Prefix}.ctl	- This is the ctl file that will run the main MpSim dynamics simulation.
{Prefix}.fix	- This is a fixed version of the {Prefix}.ctl file that has the correct $membstruk directory for the FF.
out-{Prefix}.fix	- This is the output file for a non-borg job for fixhelix.
{Prefix}-bundle332.bgf	- This file contains the first bundling of the new relaxed helices in fixhelix.
{Prefix}-bundle-ion.bgf	- This is the next step in the bundling with counter ions added in fixhelix.
{Prefix}-bundle-min.fin.bgf	- This is the bundle file after a full minimization in fixhelix.
{Prefix}-helix?.bgf	- BGF conversion of the unminimized pdf helix file that has been recentered for fixhelix.
{Prefix}-helix?-cm.bgf	- The helix file after running helixcm.exe for fixhelix.
{Prefix}-helix?-ion.bgf	- The centered helix file with counterions before minimization for fixhelix.
{Prefix}-helix?-min-A.bgf	- This is the minimized helical file before Dynamics for fixhelix.
{Prefix}-helix?-neimo.fin.bgf	- This if the final structure file after the dynamics simulation for fixhelix.
{Prefix}-helix?-neimo.NM.bgf	- This is the chosen snapshot file according to energy that has been converted to a BGF file for fixhelix.
{Prefix}-helix?-neimo-rmion.bgf	- Final dynamics helix file with counterions removes and is ready to bundle for fixhelix.
{Prefix}-helix?NM-ion.bgf	- This is the chosen Dynamics snapshot with counterions added for fixhelix.
{Prefix}-helix?NM-min.fin.bgf	- This is the Dynamics snapshot after minimization in MpSim for fixhelix.

{Prefix}-helix?-final.bgf	- This is the final helical file that has been matched back to it's original position and is ready for bundling in fixhelix. This is the file to check first to see if the unusual bends in the final bundle file is due to dynamics or due to scrwl.
{Prefix}-scw.scwrl.bgf	- This is the bundle file right after scwrl for fixhelix.
{Prefix}.fixhelix.bgf	- This is the final fixhelix file ready to be rotated.

Running Time for the Program:
12 to 18 hours.

Chapter 5 – Individual Helix Optimization

The main window for this program sets up variables needed to run the fixhelix simulation on an individual helix. Below is the main window that appears:

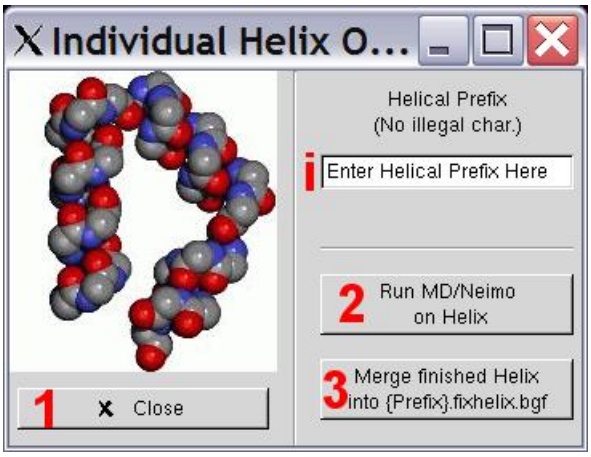


Figure 17 - Individual Helix Optimization Main Window

Text Boxes:
i) Enter here a new Prefix to be used just for these scripts. It is best to place the number of the helix being replaced in this prefix. A common default is to use {Prefix}-H# to reference the structure being used and the helix number being fixed. This prefix will be saved in the file named: HPREFIX.txt

Buttons:
1) This button will close this window and make the main window active again.

2) This button will open up the fixhelix.IndHel GUI window to run the dynamics simulation on an individual helix. See Section 5.1

3) Pressing this button will open up a GUI window to merge an individual helix file back into the fixhelix structure. This is used often to replace the {Prefix}-helix#-final.bgf file back into the final structure when scrwl has forced an artificial bend into the structure. See section 5.2.

Section 5.1 – The Individual Fixhelix GUI

This window will run the fixhelix.IndHel C-shell script on the remote machine chosen in the main window.

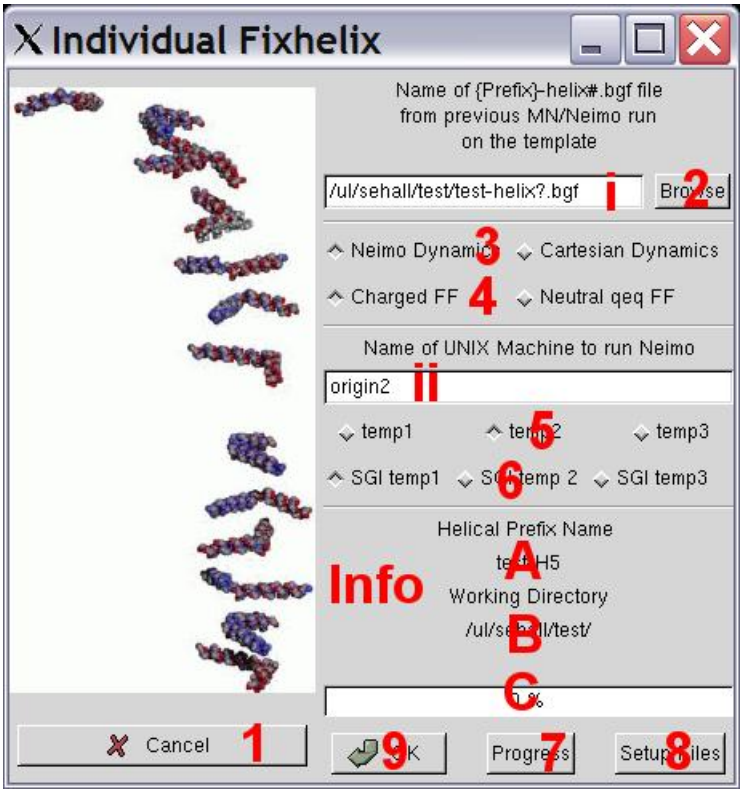


Figure 18 - Individual Helix Dynamics Window

Text Boxes:

i) Enter in here the name of the helix file from fixhelix that needs to be rerun. The file to use is called {Prefix}-helix#.bgf.

ii) This textbox should contain the name of the SGI machine that will run Biograf remotely. You must be able to ssh onto this machine via a C-Shell (you must have this machine located in your home directory .ssh/ in the known_hosts file). It is best if you try to ssh into this computer before you run this script.

Information Section:

A) This will contain the prefix that you entered in the textbox in the main Individual Fixhelix gui window.

B) The information section contains the current working directory as defined in the Main MembStruk GUI window. This is the directory that will have the final output files.

C) This is the progress bar to tell you how far along you are in the simulation. It is no longer supported since there was no easy way of keeping track of what machine was used to run the remote script.

Buttons:

1) This button will close this window.

2) Click this button to bring up a file selection window to choose the file to use in this script.

3) These two radio buttons allow you to choose what kind of dynamics simulation to run on the helices. The Neimo dynamics button will run the Neimo simulation method on the helices (this is owned by NASA and not supported in the Business Edition). The default is to have the Neimo simulation selected.

4) These two radio check buttons are used to determine what kind of charges you will use on the charged residues in your helices. By default the charged FF is selected that will set all positive residues to a +1.00 net charge and all negative residues to a net -1.00 charge. The neutral qeq FF button when selected will convert all charged residues to a neutral net charge as predetermined by QeQ.

5) These three radio buttons are used to determine which environmental variable to use to define the Linux temp directory. Choosing temp 1 will use $memtmp1 up to temp 3 will use $memtmp3. Only one can be chosen and the default is temp 1. This variable will be used to create a temporary directory ($memtmp?/{Prefix}{unique}) to store files while in use for the script.

6) These three radio buttons are used to determine which environmental variable to use to define the SGI temp directory. Choosing SGI temp 1 will use $memsgi1 up to temp 3 will use $memsgi3. Only one can be chosen and the default is SGI temp 1. This variable will be used to create a temporary directory on the remote SGI machine ($memsgi?/{Prefix}{unique}) to store files while in use for the script.

7) This button is no longer supported, but if pressed will look in your current working directory to see if the files from the fixhelix.IndvHel script are in there and update your progress bar accordingly.

8) This will place the file {HPrefix}.ctl in your working directory for you to edit. It will choose the appropriate file according to your choices in buttons 3 and 4. ***Note: If you***

edit this file and then change the method used in 3 and 4 this file will not be updated unless you press this button again.

9) Run the C-shell script fixhelix.IndvHel. See section 5.1.1 for more details.

Section 5.1.1 – Running the Fixhelix.IndvHel C-Shell script

This script is actually a merger of the 4 fixhelix scripts discussed in Chapter 4. This script will perform the same functions as the first half of those scripts on one single helix instead of all seven. This script will take your {Prefix}-helix#.bgf file that is converted and add counter ions to it. Then it will run a small minimization and then perform the 100,000 steps of dynamics selected in the radio buttons. I will convert snapshots of the structure during the simulation at every 1000 steps after 25,000 steps and place these back into the working directory. Each one of these structures will have been minimized. In this fashion, the user can obtain several structures of the same helix along the simulation and choose the best based on energy and structure.

To run this script you must use the following command:

```
fixhelix.IndvHel {PREFIX-helix?.bgf} {Helix protein prefix}
{SGI remote machine} {Linux temp drive variable #} {FF
0=ions 1=neutral} {Method 1=MD 0=NEIMO} {SGI Drive variable
#}
```

Dependant files for the Program:

fixhelix/neimo_Dyn.ctl	bin/helixcm.exe
fixhelix/neimo_Dynqeq.ctl	bin/snap2bgf
fixhelix/neimo_NQEQ.ctl	bin/snap2bgfL
fixhelix/neimo_orig.ctl	bin/nacl-appendL
FF/Biograf.cnv	bin/bgf2fasta.pl
FF/Biograf.par	bin/bgfscwrl
FF/Mpsim.par	bin/mpsim
FF/Biograf-qeq.cnv	bin/scwrl

Input Files Needed:

{Prefix}-helix?.bgf	- BGF conversion of the unminimized pdf helix file that has been recentered for fixhelix.

Output Files from the Program:

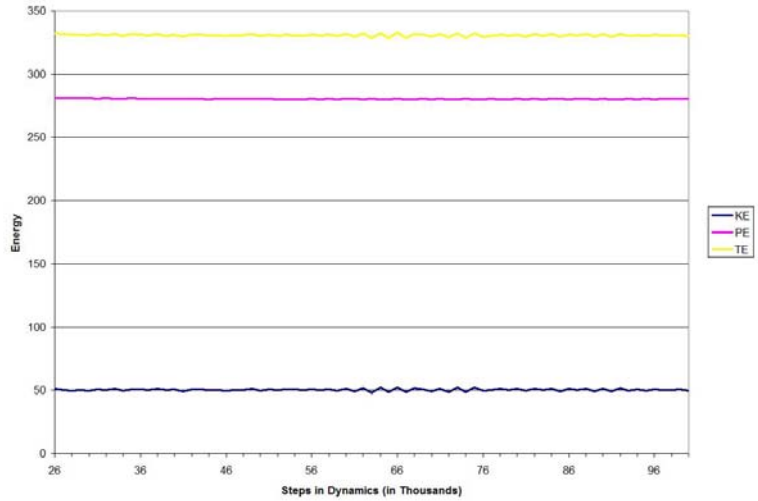
HPREFIX.TXT	- This file holds the Prefix used for the Fixhelix.IndvHel and Fixhelix.IndvPut C-shell scripts.
{HPrefix}-fixhelixLv4.10	- This file holds the qsub batch script if a linux cluster is going to be used.
{HPrefix}-snap[25K-99K].bgf	- These files hold the BGF files of the dynamics run at every 1,000 steps for you to choose the best structure.
{HPrefix}.ctl	- This file contains the choosen dynamics ctl file to use for individual helix runs with the HPrefix. Once this file is in your directory, you will need to press the setup files button to change it to the correct file if you plan on using the same HPrefix with a different method.
{HPrefix}.dat	- This file contains the list of the files produced in a script that is usable for MolecularGL
{HPrefix}.eng	- This file contains all the energies of each snapshot BGF file for the user to choose the best structure.
{HPrefix}.fix	- This is a ctl file used for MpSim after the $memstruk variable has been used.
out-{HPrefix}.fix	- This file contains the output from the script fixhelix.IndvHel if run outside a cluster.

Running Time for the Program:

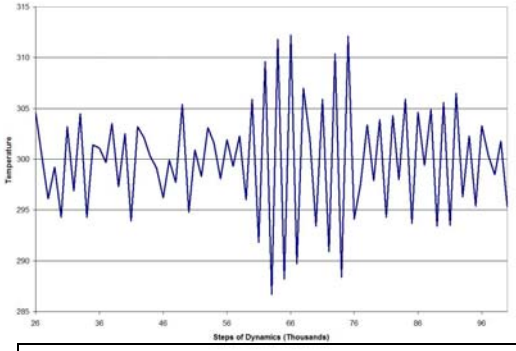
12 to 18 hours depending on the simulation method chosen.

Example 2 *– Running Fixhelix.IndvHel(not done on rhodopsin)*

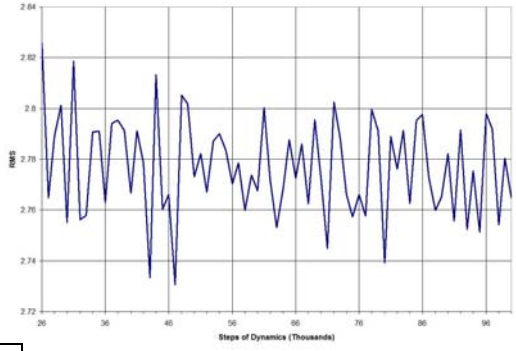
Even with all 3 examples looking fine after fixhelix, example 2 was used to test the Fixhelix.IndvHel script in the FixInd directory. This test was run on helix 5 and then the data in the {HPrefix}.eng was plotted to give the graph below.



Graph 1 - Graph of energies for Helix 5



Graph 2 - Plot of Helix 5's Temperature



Graph 3 - Plot of Helix 5's RMS

Since the Energies are so close to each other, we need to look at each file in a structure viewing program to decide what to do. We will do this in section 5.2.1.

Section 5.2 – Merging a finished helix structure into a {Prefix}.fixhelix.bgf file

Once button 3 is pressed on the window in the start of Chapter 5, it will bring up a GUI window as shown below.

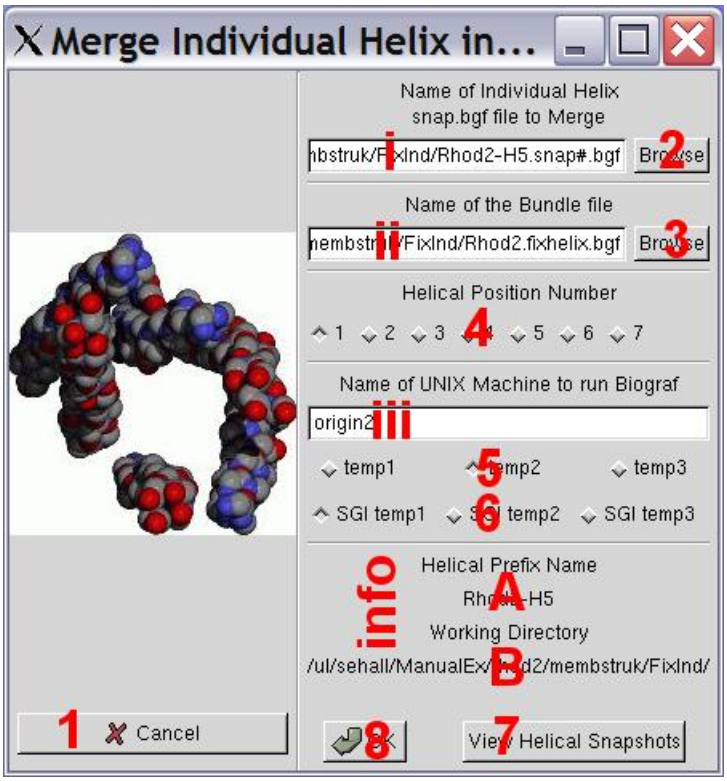


Figure 19 - Merge Individual Helix Window

Text Boxes:
i) Enter here the name of the final bgf file to merge back into the original {Prefix}.fixhelix.bgf file.

ii) Enter here the name of the bundle file ({Prefix}.fixhelix.bgf) to take out the old helix and insert the new one. ***Note: This script will make a backup of the original file, but if you do more than one helix then this back up will be overwritten. It is recommended that you make a backup of your own.***

iii) This textbox should contain the name of the SGI machine that will run Biograf remotely. You must be able to ssh onto this machine via a C-Shell (you must have this machine located in your home directory .ssh/ in the known_hosts file). It is best if you try to ssh into this computer before you run this script.

Information Section:
A) This will contain the prefix that you entered in the textbox in the main gui window.

B) The information section contains the current working directory as defined in the Main MembStruk GUI window. This is the directory that will have the final output files.

Buttons:
1) Cancel the window and return to the Individual Fixhelix window.

2) Browse for the specified file to place in textbox i.

3) Browse for the file name to place in textbox ii.

4) These seven radio buttons are used to determine which helix to replace. You must click on the helix number that corresponds to the helix you will replace.

5) These three radio buttons are used to determine which environmental variable to use to define the Linux temp directory. Choosing temp 1 will use $memtmp1 up to temp 3 will use $memtmp3. Only one can be chosen and the default is temp 1. This variable will be used to create a temporary directory ($memtmp?/{Prefix}{unique}) to store files while in use for the script.

6) These three radio buttons are used to determine which environmental variable to use to define the SGI temp directory. Choosing SGI temp 1 will use $memsgi1 up to temp 3 will use $memsgi3. Only one can be chosen and the default is SGI temp 1. This variable will be used to create a temporary directory on the remote SGI machine.

7) This will open up an xterm window and a moleculeGL window that you can use to display the snapshots from the previous run. This will only work if membstruk is run in the same directory as the snapshots and the helical prefix is the same. Pressing "-" or "+" will cycle through the snapshots and list their energies in the xterm window. See figures below:

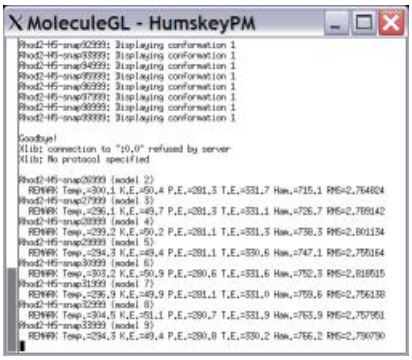


Figure 20 - MoleculeGL text window

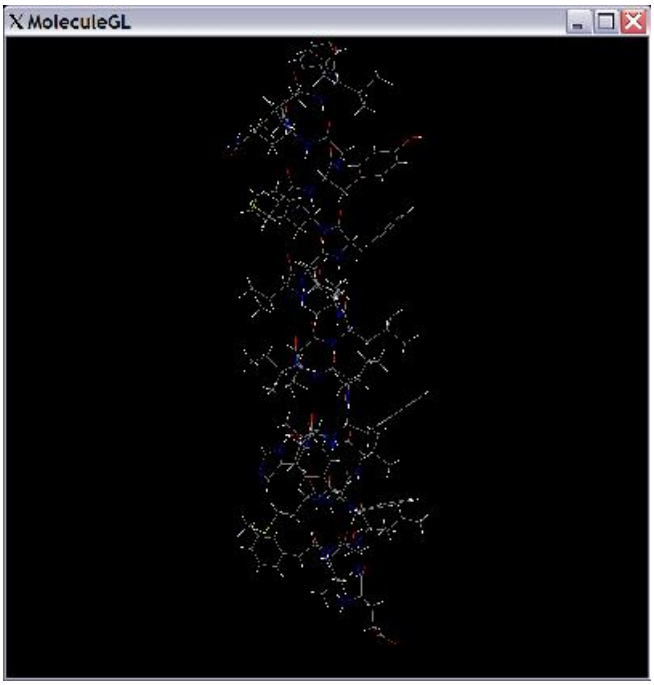


Figure 21 - MoleculeGL Display Window

8) This button will run the C-Shell script fixhelix.IndvPUT. See section 5.2.1 for more information.

Section 5.2.1 – Running the Fixhelix.IndvPUT Script

This script will look in to the bgf file and pull out the helix specified and replace it with the new helix file given. It will perform a match of the two helices in order to get the coordinates correct so the helices must have the same number of atoms. This C-Shell script will copy the old bundle file to bundle filename.bak and then delete out the helix atoms. It will then match the old helix with the new helix and place the new helix atoms into the bundle file. It will then do a scrwl and minimization to get rid of any bad contacts between sidechains.

Command to run the script:

```
fixhelix.IndvPUT {bundle bgf file} {Replacement Helix bgf
file} {Unix machine} {Linux temp #} {helix #} {SGI temp #}
```

Dependant files for the Program:

bin/nacl-appendL	bin/scwrl
bin/bgf2fasta.pl	FF/Biograf.cnv
bin/bgfscwrl	FF/Biograf.par
bin/mpsim	FF/Mpsim.pa

Input Files Needed:

{HPrefix}-snap[25K-99K].bgf	- These files hold the BGF files of the dynamics run at every 1,000 steps for you to choose the best structure.
{Prefix}.fixhelix.bgf	- This is the final fixhelix file ready to be rotated.

Output Files from the Program:

{Prefix}.fixhelix.bgf.bak	- This is the backup file of the original fixhelix file. CAUTION: This file will be overwritten if the fixhelix.IndvPUT script is used more than once to place a helix into the bundle file.
HPREFIX.TXT	- This file is rewritten with the current HPrefix that was used.
{HPrefix}-fixhelixPv4.10	- This is the cluster output file ready for submission.
PREFIX.TXT	- This file is overwritten with the current prefix used.
{Prefix}.fixhelix.bgf	- This is the new bundle file with the new helix coordinates.
out-{HPrefix}.fixP	- This file contains the output from the fixhelix.IndvPUT script.

Running Time for the Program:
8 to 16 hours depending on the differences in the original helix and the new helix structures.

Example 2 – Merging Fixed Helix 5 back into the Bundle (not done for rhodopsin)

Looking at the structures in the MoleculeGL display, there was no real difference between the structures that was discernable to my eye. In this case we default to the final file: Rhod2-H5-snap99999.bgf to replace into the fixhelix bundle. Since this was only an example it will not be used for the final structure since helix 5 appeared fine in the previous structure. However, the merged file and all cooresponding files can be found in the rhod2/membstruck/FixInd/ directory.

Chapter 6 – Hydrophobic Rotation of a BGF Bundle file

This button will activate the same window as seen in Chapter 3. The only difference is that now the default settings are to use Hydrophobic moments instead of the Phobic face method. Since the windows are identical, we will instead focus on the use of this script on bgf files instead of pdb template files. The [OK] button will start the hcenterTR-bgf.script C-shell script that will rotate the bgf file and then perform a scwrl of all sidechains. Once this is finished, the output window will popup and you will be able to see and evaluate the output and graphs produced. The actual running of the script along with several examples will be done in Section 6.1.

Section 6.1 – Running the hcenterTR-bgf Script

This script will take a bgf file with seven distinct starts and ends, and rotate them by their hydrophobic moments until they face outward from the center of the bundle. This step is needed since the overall shape of the bundle can change due to the optimization of the helices. The hydrophobic vectors are used since the helices are no longer perfectly canonical.

Command to run the script:

```
hcenterTR-bgf.script {PDB template file} {protein prefix}
{Hydo. Scale} {0 = PFace, 1 = Hydro} {SGI remote machine
name for Biograf} { Linux environmental variable [1/2/3]}
{SGI temp enrivonmental variable [1/2/3]}
```

Dependant files for the Program:

compare/compareL.exe
bin/HsprotL.exe
bin/bgf2fasta.pl
bin/scwrl
bin/bgfscwrl
FF/Biograf.cnv
FF/Biograf.par

Input Files Needed:

Hydrophobic scale file	- There is the eisenberg.dat file that is the default for this.
{Prefix}.fixhelix.bgf	- This is the default file to rotate, however this script will work on any bgf file that has the same format of seven helices like this file.
HPMCenter.txt	- This file contains the hydrophobic centers to rotate about.

Output Files from the Program:

HPMCenter.txt	- This file contains the centers entered in the GUI window.
{Prefix}-hcenterTRBGFv4.10	- Borg qsub script to run the hcenterTR-bgf.script.
out-{Prefix}.rot	- This contains the output from the hcenterTR-bgf C-Shell script.
{Prefix}.abgfR.ps	- This file is a postscript view of the bgf file after rotation.
{Prefix}.abgfR.txt	- This file contains the text information of the after rotation analysis.
{Prefix}.bRbgf.ps	- This contains the postscript view of bgf file before rotations to use as a reference.
{Prefix}.bRbgf.txt	- This is the text output of the analysis of the bgf file before rotation to use as a reference.
{Prefix}-Rot.bgf	- This is the final output file that has been rotated and scrwled.
PREFIX.TXT	- This file is updated to contain the last used prefix.

Running Time for the Program:
10 minutes maximum, any longer and there is a problem

Chapter 7 – Rotational Analysis by MpSim Energy Scores

The main window for running the Rotmin.script appears as seen below. This window will setup and start a rotmin C-shell script running that will rotate each helix by a defined number of select combinations and select by energy scores the final rotations. This is used mostly for fine grain rotations.

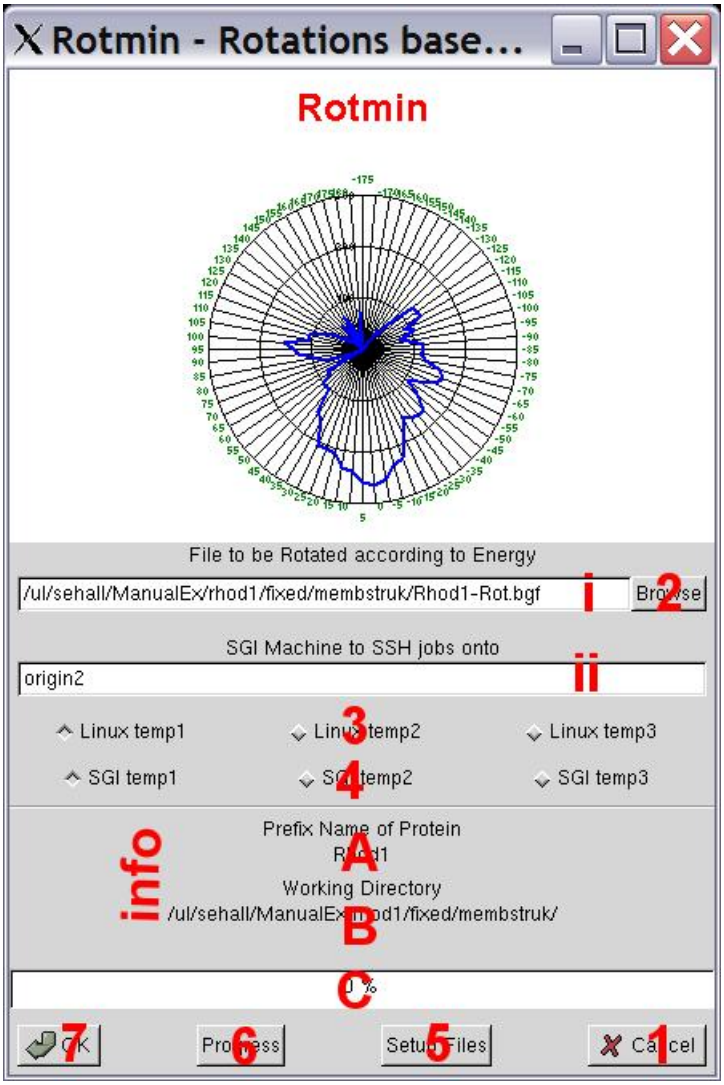


Figure 22 - Rotmin Main GUI Window

Text Boxes:

i) Enter into this textbox the file to be used for rotmin.script.

ii) This textbox should contain the name of the SGI machine that will run Biograf remotely. You must be able to ssh onto this machine via a C-Shell (you must have this machine located in your home directory .ssh/ in the known_hosts file). It is best if you try to ssh into this computer before you run this script.

Information Section:

A) This will contain the prefix that you entered in the textbox in the main GUI window.

B) The information section contains the current working directory as defined in the Main MembStruk GUI window. This is the directory that will have the final output files.

C) This progress bar will show the files that are produced as the script runs. This only works if the GUI is started in the Linux temp directory, and so is not very useful. This is a hold-over function from MembStruk 3.00.

Buttons:
1) This button will close this window and return control back to the Main Window.

2) This will bring up a file selection window to choose which file to place in textbox i.

3) These three radio buttons are used to determine which environmental variable to use to define the Linux temp directory. Choosing temp 1 will use $memtmp1, up to temp 3 will use $memtmp3. Only one can be chosen and the default is temp 1. This variable will be used to create a temporary directory ($memtmp?/{Prefix}{unique}) to store files while in use for the hcenterTR script.

4) These three radio buttons are used to determine which environmental variable to use to define the SGI temp directory. Choosing SGI temp 1 will use $memsgi1, up to temp 3 will use $memsgi3. Only one can be chosen and the default is SGI temp 1. This variable will be used to create a temporary directory on the remote SGI machine

5) This button will place the following file in your working directory for you to edit before you run the rotmin script. These will be explained in more detain in 7.1 under needed files.
degree.dat
degree-IH.dat
helix.dat
min_rotmin.ctl
Warning: These files will be used for all rotminL.scripts that use their directory as the working directory, so if you want the defaults again then delete these after the script has finished.

6) This will activate the progress bar found in Information C. This has many limitations (see information Section C).

7) This will submit a background job to run the rotmin script. See Section 7.1 for more details.

Section 7.1 – Running the RotminL.script

The rotminL.script is a C-shell script that runs through a focused set of rotational combinations to find a nearby energy minimum. It's main goal is to fine tune your rotations that were done with coarser methods looking at hydrophobicity. This script makes the assumption that your helices are in the ballpark of good rotations, otherwise a

little fine tuning is not going to help. This script uses two files for input: degree.dat are degrees to be checked, and helix.dat is the order and helices to check. It will run through 2 loops the top one is degrees and the inner loop is helices, that each degree listed is tried on all helices before a new degree. The defaults have the degrees starting at 5 and going to 25 then back to 5 giving the structure a chance to build up to larger rotations taking into account nearby helices and then fine tune back down. The default order for the helices is: 3 2 1 7 6 5 4, starting with the innermost helix and spiraling outward to the outside helices that are more likely to have a good starting rotation since their hydrophobicity is generally a better predictor.

Once a helix is rotated the whole structure has a sidechain replacement done on it, then it is fully minimized, and then the helix is minimized with the rest fixed. The final energy from the helix movable, rest of the protein fixed is used to determine if the rotation is good enough to keep. If the energy is lower, then it is noted and added to the next round of rotations. This way the final structure is rotationed only once with all the best rotations and this keeps any translational error to a minimum.

Command to run the script:

```
$membstruck/rotminL.script {file to be rotated.bgf}
{Prefix} {SGI machine name} {[1-3] to choose linux temp
variable} {[1-3] to choose SGI temp variable}
```

Dependant files for the Program:

FF/Biograf.par
FF/Biograf.cnv
FF/Mpsim.par
bin/scwrl
bin/bgfscrwl
bin/mpsim
bin/bgf2fasta.pl
bin/nacl-appendL
compare/HsprotL.exe

Input Files Needed:

{Prefix}-Rot.bgf - This is the file after bgf rotation.

degree.dat	- This file contains the order and degrees that will be tested. This can be edited before running the script.
helix.dat	- This file contains the order and helices to run fine rotations on. This can be edited before running the script.
HPMcenter.txt	- This contains the HPM centers used for rotations.
min_rotmin.ctl	- This is the MpSim input file for the second protein fixed, helix movable minimization and can be edited before running the script.
min_rotmin_m2.ctl	- This is the MpSim input file for the first minimization with the whole protein movable, and can be edited before the script runs.

Output Files from the Program:

{Prefix}-rotminv4.10	- This executable file contains the qsub script to run rotminL.script on a linux cluster.
PREFIX.TXT	- This file is updated with the prefix used for the most current operation.
min_rotmin_m2.fix	- This is the min_rotmin_m2.ctl file updated with the $membstruk variable.
min_rotmin.fix	- This is the min_rotmin.ctl file updated with the $membstruk variable.
{Prefix}.rotmin.bgf	- This is the final file after finishing rotminL.script. It contains in it's remarks section the rotations used on it.
{Prefix}.rotmin.out	- This contains all the energy and rotational information used by rotmin.

Running Time for the Program:
8 to 16 hours depending on the condition of the structure.

Chapter 8 – Individual Helical Rotational Energy Scan

This part of the method can be run either after rotmin, or after RBMD. The main difference of running after rotmin versus after RBMD is one of time and accuracy. Running the scans after rotmin resulted in 3 of 7 helices having a definite energy minimum near the correct crystal rotation (1B used), and 5 of 7 helices after running RBMD. However, the gain in rotations costs an additional 3-4 days if running after RBMD. So it is a matter of deciding how much time is available to develop the structure.

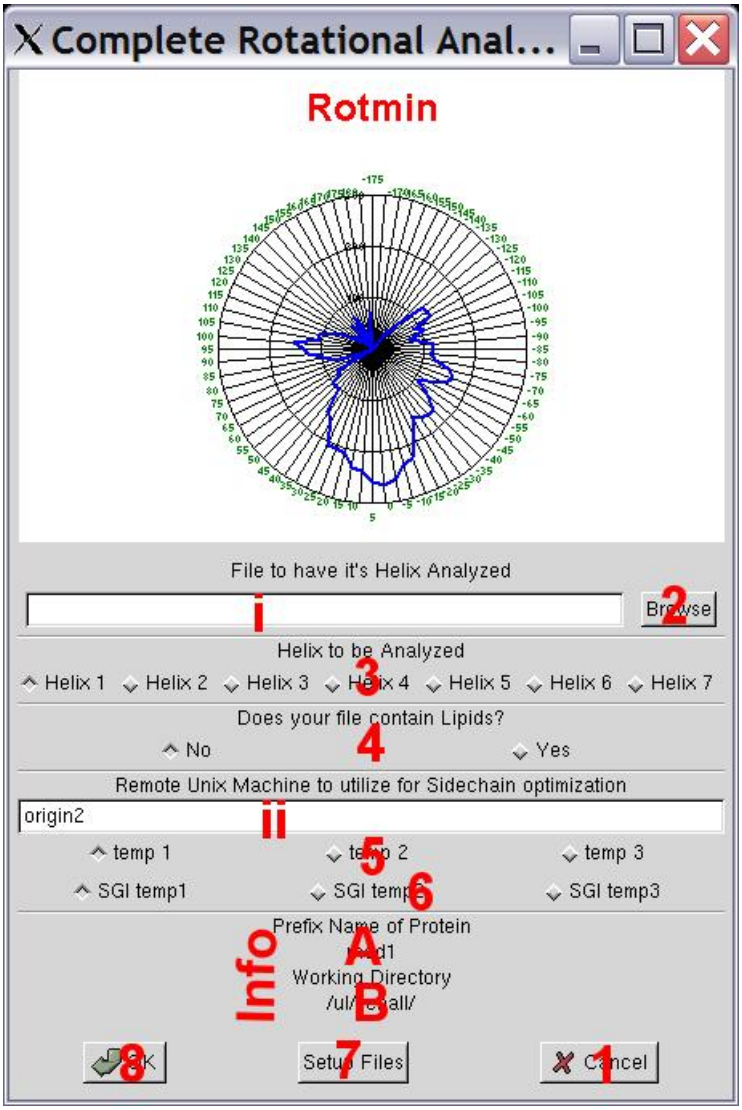


Figure 23 - Main Window for RotminL.IndHel

Text Boxes:

i) Insert the name of the file to have one of it's helices scanned energetically. There must not be any loops in the structure, but lipids are acceptable.

ii) This textbox should contain the name of the SGI machine that will run Biograf remotely. You must be able to ssh onto this machine via a C-Shell (you must have this machine located in your home directory .ssh/ in the known_hosts file). It is best if you try to ssh into this computer before you run this script.

Information Section:

A) This will contain the prefix that you entered in the textbox in the main gui window.

B) The information section contains the current working directory as defined in the Main MembStruk GUI window. This is the directory that will have the final output files.

Buttons:
1) Close the current window and return control to the previous window.

2) Open up a fileselection window in order to select a file to place in textbox i.

3) These seven radio buttons allow the user to choose the helix to run in the helical scan. These buttons assume that the BGF file is set up in the same format as the Fixhelix bundle file with the helices sequentially listed in the file.

4) These two radio button allow you to declare if your file has lipids (Yes) or is a plain bundle file (No).

5) These three radio buttons are used to determine which environmental variable to use to define the Linux temp directory. Choosing temp 1 will use $memtmp1 up to temp 3 will use $memtmp3. Only one can be chosen and the default is temp 1. This variable will be used to create a temporary directory ($memtmp?/{Prefix}{unique}) to store files while in use for the script.

6) These three radio buttons are used to determine which environmental variable to use to define the SGI temp directory. Choosing SGI temp 1 will use $memsgi1 up to temp 3 will use $memsgi3. Only one can be chosen and the default is SGI temp 1. This variable will be used to create a temporary directory on the remote SGI machine ($memsgi?/{Prefix}{unique}) to store files while in use for the script.

7) This will place the following files into your working directory for you to edit: min_IHrotmin.ctl, min_IH_m2.ctl, degree-IH.dat. See appendix for descriptions of these files, and note that once they are in your directory they will always be used for **any** scan job run in that directory.

8) This will submit the rotminL.IndHel script to the remote machine specified. See section 8.1 for more details.

Section 8.1 – Running the rotminL.IndHel C-Shell Script

The rotminL.IndHel C-shell script will take a bundle file (with and w/o lipids) and rotate the helix specified by + and -5 degrees (specified in degreesIH.dat) to 180 degrees covering all possible rotations. For each degree specified in degreesIH.dat, rotminL.IndHel will rotate the structure + and – and then fully minimize the whole protein and then fix all but the helix and minimize and record the final energy. This energy is then plotted in a postscript file that contains the negative energies or the lowest

energies within 300 of each other, and the top 1/3 of those energies. Plus, a file for each rotation is generated and placed in the working directory so that if there is a better structure you can simple choose this structure then recreate it.

Command to run the script:

```
rotmin {File name} {Helix #} {Prefix} {1 = lipid, 0 = no
lipid} {SGI machine name} {Linux temp #} {SGI temp #}
```

Dependant files for the Program:

FF/Mpsim.par	bin/scwrl
rotmin/min_IHrotmin.ctl	bin/bgfscwrl
rotmin/min_IH_m2.ctl	bin/mpsim
rotmin/degree-IH.dat	bin/bgflipidscwrl
rotmin/awkrotminIH	bin/nacl-appendL
rotmin/awkrotminIH-3	bin/bgf2fasta.pl
bin/bgf2fasta.pl	bin/rotate3L.exe

Input Files Needed:

{Bundle BGF file}	- This is the file to run the rotation scan on.
min_IHrotmin.ctl	- Edit this file in your working directory to change the fixed helix minimization.
min_IH_m2.ctl	- Edit this tile to change the full protein minimization parameters.
degree-IH.dat	- Edit this file to change the scope of what is scanned. Every number listed in this file will be scanned + and – from the starting position.
HPMCenter.txt	- This file must be present in the directory for the rotations to be done correctly.

Output Files from the Program:

degree-IH.dat	- This file contains the degrees to rotate the helix. It starts with all degrees from 0 to 360 in 5 increments, but can be modified to restart a script that has failed.
min_IH_m2.ctl	- This file contains the ctl file for MpSim to run the first full protein minimization.
min_IH_m2.fix	- This is an edited form of the min_IH_m2.ctl file that contains corrected directory information.
min_IHrotmin.ctl	- This is a ctl file for MpSim to run the fixed protein helix movable minimization.
min_IHrotmin.fix	This is an edited form of the min_IHrotmin.ctl file with corrected directory information.

{Prefix}-H{helix number}{degree}.IH.bgf - This is a snapshot bgf file of the protein after helix {number} has been rotated {degree} amount, scrwled, and minimized. This file can be chosen to replace the original file if the profile looks better.

{Prefix}-H{Helix #}-rotminIHv4.10 - Lunux qsub script for running Individual Rotation Scan on a cluster.

{Prefix}-H{Helix #}.rotmin.out - Output file that contains all the information from the rotational scan. The 5 degree block of lines, contain zero rotation information as well. Under the energies line of this first block it goes: {energy 0} {energy +5} {energy -5} {saltbridges 0} {saltbridges +5} {saltbridges -5} {HBonds 0} {HBonds +5} {HBonds -5}. For every other block of lines, it looks like this: {energy +deg} {energy -deg} {saltbridges +deg} {saltbridges -deg} {HBonds +deg} {HBonds –deg}

{Prefix}-H{Helix #}.rotmin.ps - This is a postscript file of the information contained in the output file graphed on a scale of the largest negative energy score or a 300 scale if there are no negative numbers.

{Prefix}-H{Helix #}.rotmin.ps-3 - This is a postscript file of the information contained in the output file graphed on a scale of the top 1/3 negative energies, or if no negative energies exist, then a scale of 100.

Running Time for the Program:
8-10 hours w/o lipids to 2-4 days with lipids

Chapter 9 – Helical Rotation Combination Generation

This GUI window will run the rotminL.CombDiv C-Shell script that will create a set of optimized structures containing a combination of specified rotations. This script is to be used after the rotational scans of all the helices has been done. That way the combinations are based on the best energy minima seen in the graphs from the helical scans. Then you can visually inspect the structures along with the energy, saltbridge, and hydrogen bond information that is provided in the output file and choose which structures are best for docking. This will also show structures that might have multiple conformations based on inactive, active, and antagonist docking.

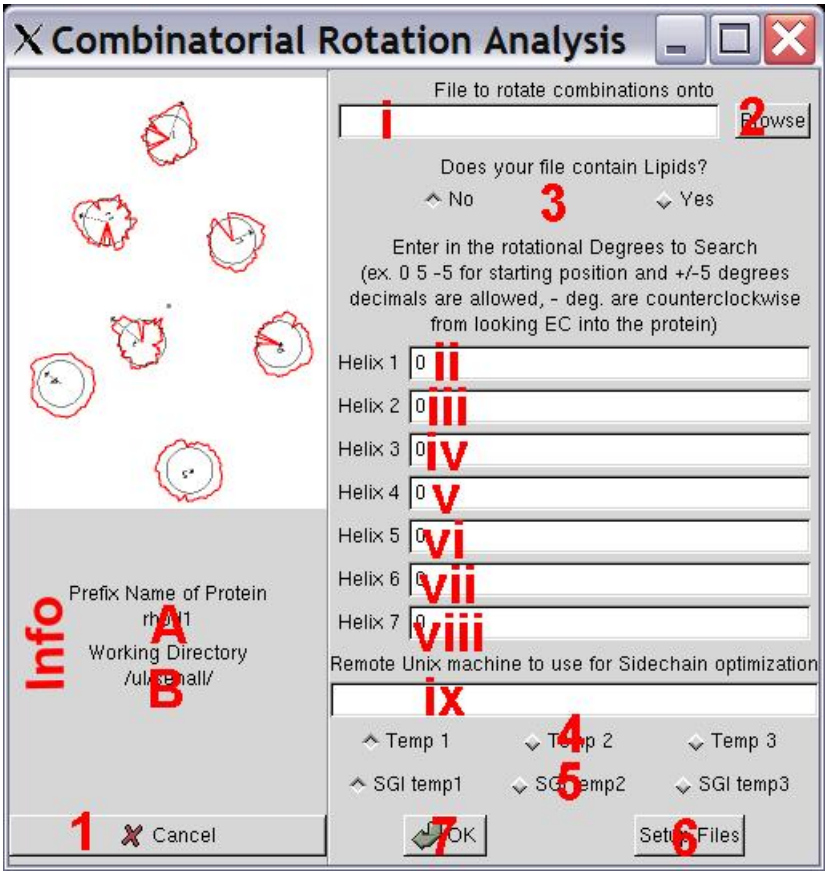


Figure 24 - Main GUI Window for Rotational Combination Generation

Text Boxes:

i) Place in this box the name of the bgf file to generate combinations on.

ii) – viii) Enter in these boxes the rotational angle to consider for generating the combinations. Each angle that you want used should be placed with a space next to the previous one. Example) 0 -5 5 in textbox 1 will generate 3 structures: a default no rotations made structure, a structure with helix 1 rotated +5 degrees, and a structure with helix 1 rotated -5 degrees. If you leave out 0 then the starting position for that helix will not be generated. Remember that combinations multiply so the number of structures generated will grow rapidly and each structure is 1/3 of a Mbyte and takes ~0.5-3 hours to generate depending on how stable the structure is.

ix) This textbox should contain the name of the SGI machine that will run Biograf remotely. You must be able to ssh onto this machine via a C-Shell (you must have this machine located in your home directory .ssh/ in the known_hosts file). It is best if you try to ssh into this computer before you run this script.

Information Section:

A) This will contain the prefix that you entered in the textbox in the main gui window.

B) The information section contains the current working directory as defined in the Main MembStruk GUI window. This is the directory that will have the final output files.

Buttons:

1) This button will close this window and return to the main window for MembStruk.

2) This will open up a browser window to select a filename to place in textbox i).

3) These two radio buttons allow you to specify if your file has lipids (RBMD has been run) or does not.

4) These three radio buttons are used to determine which environmental variable to use to define the Linux temp directory. Choosing temp 1 will use $memtmp1 up to temp 3 will use $memtmp3. Only one can be chosen and the default is temp 1. This variable will be used to create a temporary directory ($memtmp?/{Prefix}{unique}) to store files while in use for the script.

5) These three radio buttons are used to determine which environmental variable to use to define the SGI temp directory. Choosing SGI temp 1 will use $memsgi1 up to temp 3 will use $memsgi3. Only one can be chosen and the default is SGI temp 1. This variable will be used to create a temporary directory on the remote SGI machine ($memsgi?/{Prefix}{unique}) to store files while in use for the script.

6) This will place in your working directory the files: {Prefix}_min_CDrotmin.ctl, {Prefix}_min_CD_m2.ctl to be edited.

7) This will run the C-Shell script rotminL.CombDiv. See section 9.1 for more details.

Section 9.1 – Running the rotminL.CombDiv C-Shell script

This C-Shell script will take a list of rotations for each of the seven helices and run through a series of loops to generate all the possible combinations. Each structure produced will have been rotated, had their sidechains replaced, fully minimize the entire protein, and minimize the helix with all atoms fixed.

Command to run the script:

```
rotminL.CombDiv {Input bgf file - no ions if no lipids are
present} {Prefix} {1 if input file contains lipids, 0 for
no lipids} {SGI remote machine name} {1/2/3 for Linux
```

```
environment variable} {1/2/3 for the SGI environment
variable}
```

Dependant files for the Program:

FF/Mpsim.par
FF/Biograf.par
FF/Biograf.cnv
bin/bgf2fasta.pl
bin/scwrl
bin/bgfscwrl
bin/mpsim
bin/nacl-appendL
bin/bgflipidscwrl

Input Files Needed:

{input file}	- This is the file to generate rotational combinations from. If it lacks lipids then it must not contain ions.
rotmin/min_CDrotmin.ctl	- This file is used for the fixed minimization portion of the script and can be edited in your directory under the name: {Prefix}_ min_CDrotmin.ctl.
rotmin/min_CD_m2.ctl	- This file is used for the protein movable minimization part of the script and can be edited in your working directory under the name: {Prefix}_min_CD_m2.ctl.
rotmin/Default.CD.rot	- This file contains the list of rotations per helix to create combinations from. In your working directory it will be named: {Prefix}.CD.rot.

Output Files from the Program:

Running Time for the Program:

30 minutes to 3 hours per conformation generated

Chapter 10 – Rigid Body Molecular Dynamics

This part of the method is user dependant on time to be spent. The rigid body molecular dynamics (RBMD) can be run after rotmin (Chapter 7) or after rotational combinations (Chapter 9). The benefit of running RBMD after rotmin is that you can then use the lipid file generated for the individual helical scans (Chapter 8) and for combinations making them more effective. However, adding lipids to these scripts increases the time it takes to finish by up to four days. So if time does not permit, then

you can run helical scans and combinations before RBMD with only a small loss in possible accuracy.

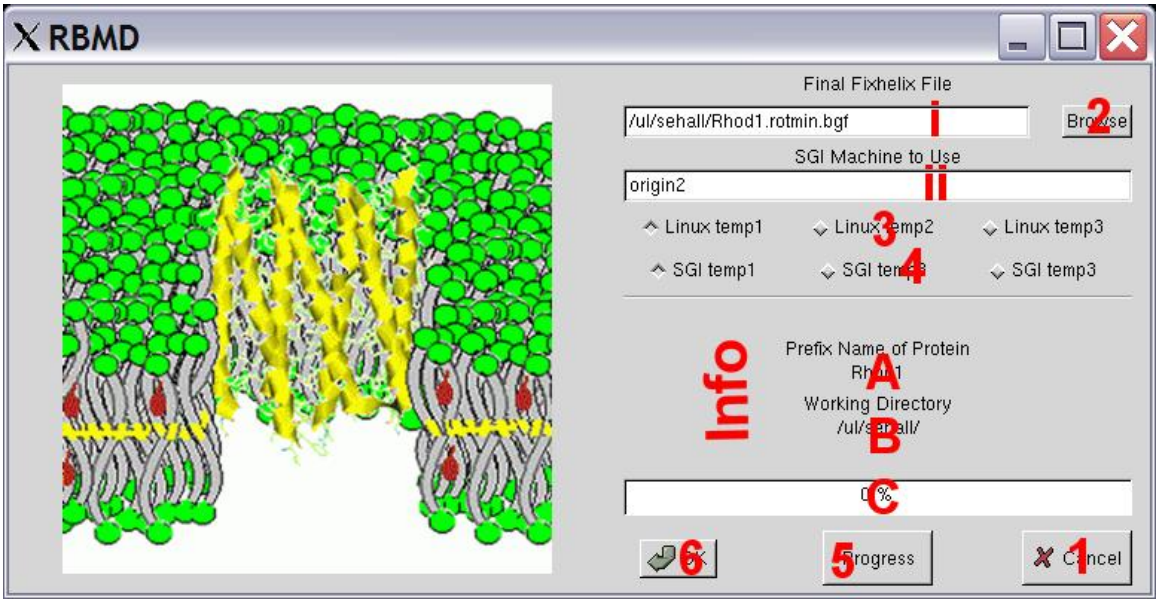


Figure 25 - RBMDL.script Main Window

Text Boxes:

i) This should contain the name of the file to run RBMD on. By default the file {Prefix}.rotmin.bgf is located in this box.

ii) This textbox should contain the name of the SGI machine that will run Biograf remotely. You must be able to ssh onto this machine via a C-Shell (you must have this machine located in your home directory .ssh/ in the known_hosts file). It is best if you try to ssh into this computer before you run this script.

Information Section:

A) This will contain the prefix that you entered in the textbox in the main GUI window.

B) The information section contains the current working directory as defined in the Main MembStruk GUI window. This is the directory that will have the final output files.

C) This is a no longer supported progress bar. It only works if the temp directory and the working directory are the same.

Buttons:

1) This button will close this window and return control over to the main GUI window.

2) This will open up a fileselection window to choose what file to place in textbox i.

3) These three radio buttons are used to determine which environmental variable to use to define the Linux temp directory. Choosing temp 1 will use $memtmp1 up to temp 3 will use $memtmp3. Only one can be chosen and the default is temp 1. This variable will be used to create a temporary directory ($memtmp?/{Prefix}{unique}) to store files while in use for the script.

4) These three radio buttons are used to determine which environmental variable to use to define the SGI temp directory. Choosing SGI temp 1 will use $memsgi1 up to temp 3 will use $memsgi3. Only one can be chosen and the default is SGI temp 1. This variable will be used to create a temporary directory on the remote SGI machine ($memsgi?/{Prefix}{unique}) to store files while in use for the script.

5) This button will look in the working directory and count how many files there are towards finishing the script and update the progess bar. This function is no longer supported and will only work if the working directory and the Linux temp directory being used are the same.

6) This will submit the RBMDL.script on the remote machine specified. See section 10.1 for more details.

Section 10.1 – Running the RBMDL.script

RBMDL.script is a C-Shell script that will take a bundle bgf file and using Biograf add 52 lipid structures around the bundle. Then it will minimized the entire protein, and then run a 50,000 step rigid body dynamics. This pack the lipids around the bundle in the first 25,000 steps and then move and settle the helices with this lipid packing. Once this is done, the sidechains are replaced using the lipids as a bump check. Lastly, the file is converted into a pdb structure ready for loop addition with Whatif (Vriend 1990) or SwissPDB (Guex 2003). Otherwise, use the last bgf file for individual helix scanning.

Command to run the script:

```
RBMDL.script {PREFIX}.rotmin.bgf {PREFIX} {SGI machine}
{Linux temp} {SGI temp}
```

Dependant files for the Program:

FF/Biograf.par	bin/nacl-append
FF/Biograf.cnv	bin/mpsim
FF/Mpsim.par	bin/pdb2fasta.pl
FF/Mpsim-RBMD.par	bin/scwrl
bin/bundlecm.exe	RBMD/lipidbarrelgood-cm.bgf

Input Files Needed:

{Prefix}.rotmin.bgf - This is the BGF bundle file to run the script on.

Output Files from the Program:

out-{Prefeix}.RBD	- This file contains the output from the RBMD script.
{Prefix}-lipid-min.fin.bgf	- A bgf file after merger of the lipid and protein files and a minimization.
{Prefix}-lipid-rbmd.bgf	- The bgf file of only the lipids with no protein after RBMD has finished but before scwrl or the final minimization.
{Prefix}-lipid-rbmd.pdb	- The pdb file conaining just the lipids after RBMD but before the final minimization and scwrl.
{Prefix}-rbmd.fin.bgf	- This is the final file after the long RBMD run finishes, but before scwrl or minimization.
{Prefix}-rbmd-min.fin.bgf	- This is the final file after RBMD and scwrl. This is the file to use for the Modeller scripts.
{Prefix}-rbmd.snap*	- MpSim snapshots of the RBMD simulation
{Prefix}-rbmd.swl.pdb	- Final file from RBMD for use with whatif or SwissPDB Viewer to add loops.
{Prefix}-RBMDv4.10	- Borg qsub script generated for use on a linux cluster to run the RBMD script.

Running Time for the Program:

Three to seven days depending on the processor

Chapter 11 – Loop Addition with Modeller

This GUI window will setup the necessary information to run the makeloops.pl or the makeloops-stepwise.pl script. Since these two Perl scripts will only run on a SGI machine with modeler installed, the entire script is submitted to run on the remote SGI

machine and will have no contact with the submitting Linux machine. These scripts are to facilitate the addition of loops to the bundle bgf file.

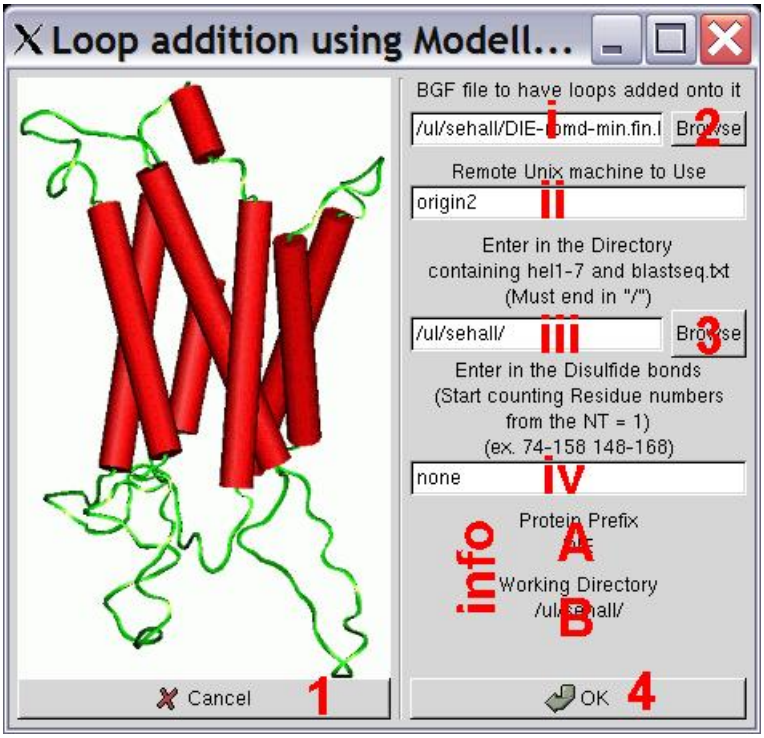


Figure 26 - Main Window for the Makeloops.pl Script

Text Boxes:

i) The name of the file to add loops onto should be placed here. The default name is {Prefix}-rbmd-min.fin.bgf. This script assumes that you have lipids, but it shouldn't matter if your file doesn't.

ii) This textbox should contain the name of the SGI machine that will run Biograf remotely. You must be able to ssh onto this machine via a C-Shell (you must have this machine located in your home directory .ssh/ in the known_hosts file). It is best if you try to ssh into this computer before you run this script. This machine will have sole control over the perl script since there is no need to use the Linux machine.

iii) Enter in here the directory that contains the blastseq.txt, and hel[1-7] files. The hel[1-7] files must not have any gaps in their sequences, and correspond to the TM regions finally chosen for you protein.

iv) Enter in the cystine residues that form cystine bridges. The numbering of the residues is the same as the numbering in the blastseq.txt and not in the bgf file residue column. Enter in pairs connected by a dash ex) 101-178. For the makeloops-stepwise.pl

script these bridges can only be in the same loop or loop to TM region not loop to different loop.

Information Section:

A) This will contain the prefix that you entered in the textbox in the main gui window.

B) The information section contains the current working directory as defined in the Main MembStruk GUI window. This is the directory that will have the final output files.

Buttons:

1) This button closed the window and returns to the main GUI window.

2) This will open up a fileselection window to choose a filename to enter into textbox i).

3) This will open up a fileselection window to choose a filename to enter into textbox iii).

4) Run the perl script: makeloops.pl. If you want to run the script makeloops-stepwise.pl then enter in a fake SGI machine name and then edit the file {Prefix}-modeller.script's last line from makeloops.pl to makeloops-stepwise.pl and then login onto the SGI machine of choice and type: `{Prefix}-modeller.script > & {output file} &` in the working directory. See section 11.1 for more details.

Section 11.1 – Running the makeloops.script

This script will take a bundle file with lipids and strip off the lipids, then run modeler to add loops as specified in the blastseq.txt file in 5 outputs. It will then run scwrl on the loops of each of the 5 structures and then do a series of minimizations and print in the output the energies of the five structures so you can choose the best. In makeloops.pl the loops are added simultaneously, however with some tight structures this causes the loops to be intertwined so you will want to run makeloops-stepwise.pl that adds the loops one at a time. If you want to run the script makeloops-stepwise.pl then enter in a fake SGI machine name and then edit the file {Prefix}-modeller.script's last line from makeloops.pl to makeloops-stepwise.pl and then login onto the SGI machine of

choice and type: `{Prefix}-modeller.script > & {output file} &` in the working directory.

Command to run either script:

```
$membstruk/loops/modeller/makeloops.pl –f {input file name}
–p {prefix} –s ‘{disulfide bridge residues here}’
```

Dependant files for the Program:

/loops/modeller/modeller-template.macro	/loops/modeller/bgf2pdb.pl
All files in FF/	/loops/modeller/mod6v2_iris4d
/loops/modeller/chainA.pl	/loops/modeller/mpsim.pl
/loops/modeller/bgfrenum.pl	/loops/modeller/change_ntct_crg.pl
	/loops/modeller/biovar

Input Files Needed:

{Input bgf file}	- This is the bundle file with lipids that is ready for loops.
blastseq.txt	- This needs to have the protein’s sequence information as the first sequence. The GUI will copy this file to your working directory, otherwise you need to do that.
hel[1-7]	- These files need to be in your working directory. The GUI will copy them for you, or you must do this yourself.

Output Files from the Program:

PREFIX.txt	This contains the prefix chosen in the GUI.
{Prefix}-modeller.script	- This is a C-Shell script that will run the makeloops.pl script on a SGI machine. It can be edited to run the makeloops-stepwise.pl script by simply replacing the makeloops.pl with makeloops-stepwise.pl.
{Prefix}-formod.ali	- This contains the alignment information for modeler to use.
{Prefix}-ptnonly.bgf	- This file contains the input file with no HETATMs.
model-{Prefix}-withloop.top	- This file contains the modeler script for adding the top loops.
{Prefix}-ready.bgf	- This is the cleaned up file ready for modeler.
dreidii322-*	- These are forcefield files used with modeler and biograf.
afterssbonds-{Prefix}-[1-5].pdb	- These files are right after modeler and the cysbridge bonds have been made.
{Prefix}-[1-5]-afterscwrl.pdb	- These files contain the bundle with loops after running scrwl.
{Prefix}-[1-5]-energy.out	- This contains the energy output from mpsim for the looped file.

{Prefix}-[1-5]-finalmin-done.snap* - This is an output file from mpsim.
{Prefix}-[1-5]-finalmin-done.traj1 - Another output file from mpsim.
{Prefix}-[1-5]-finalmin.bgf - This is the looped structure after minimization.
{Prefix}-[1-5]-finalstruct.bgf - This is the final cleaned file that you will use for the next step in the method.
{Prefix}-[1-5]-loopmin-done.snap* - This is an output file from MpSim.
{Prefix}-[1-5]-loopmin.bgf - This is the structure after modeler has added loops and been minimized.
{Prefix}-[1-5]-readymin-done.snap* - This file is an output file from MpSim.
{Prefix}-[1-5]-readymin-done.traj1 - This file is an output file from MpSim.
{Prefix}-[1-5]-readymin.bgf - This file is the structure minimized and ready for modeler.
{Prefix}-mod.out - This is the output file from modeler.
{Prefix}-withloop.* - These are modeler ouput files.

Running Time for the Program:

1 to 3 days depending on the speed of the machine

Chapter 12 – Folding of the EC2 loop (BETA)

This C+ program (EC_LOOP_SIM) is designed to run through a series of annealing dynamics in order to force the EC2 loop to close down on top of the protein barrel. This is a C+ program similar in style to the TMPred2ndS programs. MembStruk will just open a text window with the program running in it, and the rest is following the directions presented on screen. This program is currently in beta testing. The annealing tends to bend the top half of the helices too far away from the bundle.

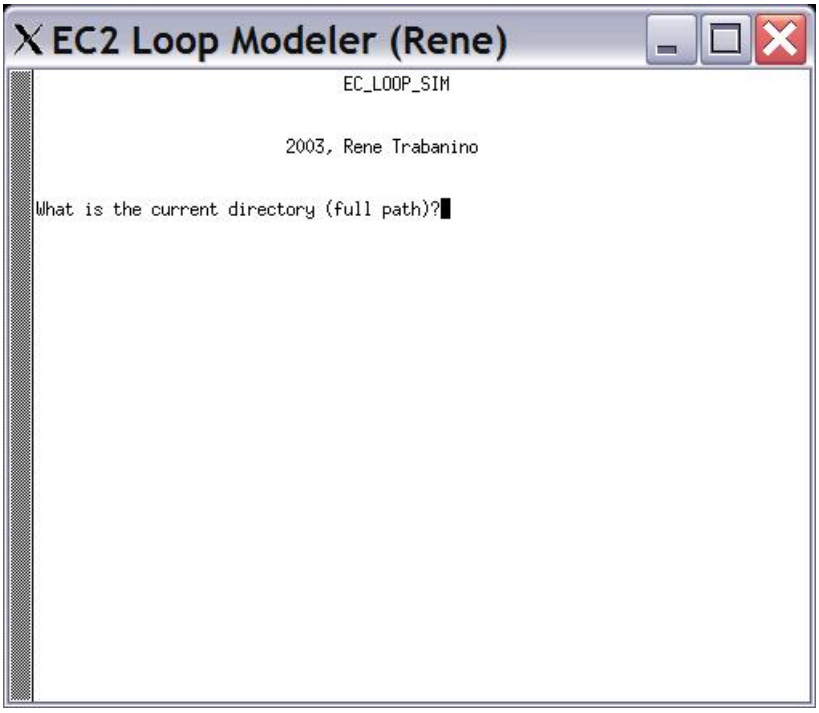


Figure 27 - Text Window for Running EC_LOOP_SIM

Running the program:
1) `What is the current directory (full path)?` {enter the working directory}

2) For these questions you will want to use the {Prefix}-TMs.doc and the input file to get the needed information. Once you do, write the information down somewhere so you don't have to get it again if you need to rerun the program.

```
What is the first atom for EC I ?
What is the last atom for EC I ?
What is the first atom for EC II ?
What is the last atom for EC II ?
What is the first atom for EC III ?
What is the last atom for EC III ?
```

3) You are now give a series of options, that you follow sequentially:

```
Choose 1 to Annealing Dynamics on open loops.
Choose 2 to Form Disulfide with intermediate annealing steps.
Choose 3 to TVN dynamics on ECII.
Choose 4 to run high temp annealin in loop and surrounding residues.
Choose 5 to exit program.
Enter your choice and press return:
```

4) Option 1

a) What is the name of the input bgf file with open loops (non-constrained)? {Enter the name of the input file, this file is bgf with loops and no Cys bridges have been made yet}

b) What is the description from the BGF file: {look in your file in the first few lines it is the word after DESCRP

c) What is the name of the output bgf filename? {enter the output filename}

d) You are now asked for the residue ranges of the loops, once again write this down once you have the information.

What is the first residue for EC I ?
What is the last residue for EC I ?
What is the first residue for EC II ?
What is the last residue for EC II ?
What is the first residue for EC III ?
What is the last residue for EC III ?
What is the last residue number in the protein?

e) On which origin do you want to run this open loop annealing step (origin1 or origin2)? {you should be able to enter any SGI machine that can be ssh'ed into that has biograf installed.}

Output files:

mine2*.macro	- These are macros to start the annealing simulation for the EC loops.
ANNEAL_LOOP.script	- C-Shell script to run the Annealing macros.
out_anneal_open_loops	- Output from option 1 in the EC2 program.

5) Option 2

a) What is the name of the input bgf file with open loops (non-constrained)? {Enter the name of the input file from option 1}

b) What is the description from the BGF file: {look in your file in the first few lines it is the word after DESCRP

c) What is the name of the output bgf filename ? {Enter a filename for output}

d) Once again the next questions you will want to write down the information that you gather.

What is the H atom number on the lower numbered cysteine of the disulfide?

```
What is the H atom number on the higher numbered
cysteine of the disulfide?
What is the S atom number on the lower numbered
cysteine of the disulfide?
What is the S atom number on the higher numbered
cysteine of the disulfide?
```

e) `On which origin do you want to run this open loop annealing step (origin1 or origin2)?` {you should be able to enter any SGI machine that can be ssh'ed into that has biograf installed.}

Output files:

make_*.macro	- These macros are used to create the Cys bridges in option 2 of the EC2 folding program.

6) At this point sometimes the file isn't minimized enough, so I run the output file in the loop relaxation program (see Chapter 13).

7) Option 3

a) Aren't you glad you wrote down that information:

```
What is the first residue for EC II ?
What is the last residue for EC II ?
What is the last residue number in the protein?
```

b) `What is the name of the input bgf filename ?` {Enter here the filename that finished from option 2 and loop relaxation}

c) `What is the distance between these sulfur atoms?` {Enter in the distance between the two sulfer atoms currently going to make a bridge}

d) `What is the name of the output bgf filename ?` {Enter a name for an output file}

Output Files:

out_anneal_dis	- Output file from option 3 of EC2 program.
mine_*.macro	- These macros are used to create the Cys bridges in option 2 of the EC2 folding program.

8) Option 4

a) `What is the description from the BGF file:` {look in your file in the first few lines it is the word after DESCRP

b) `What is the name of the input bgf file with open loops ?` {Enter the name of the input file from Option 3}

c) `What is the name of the output bgf filename ?` {Enter a name for an output file}

d) Again you are glad you wrote down all that information:
`What is the first residue for EC II ?`
`What is the last residue for EC II ?`

Output files:

run_em.script	- This will setup the jobs to run in biograf for EC2 option 4
min2_good_anneal*.macro	- These are macros for biograf from EC2 option 4
out_run_em	- This is the output file from EC2 Option4

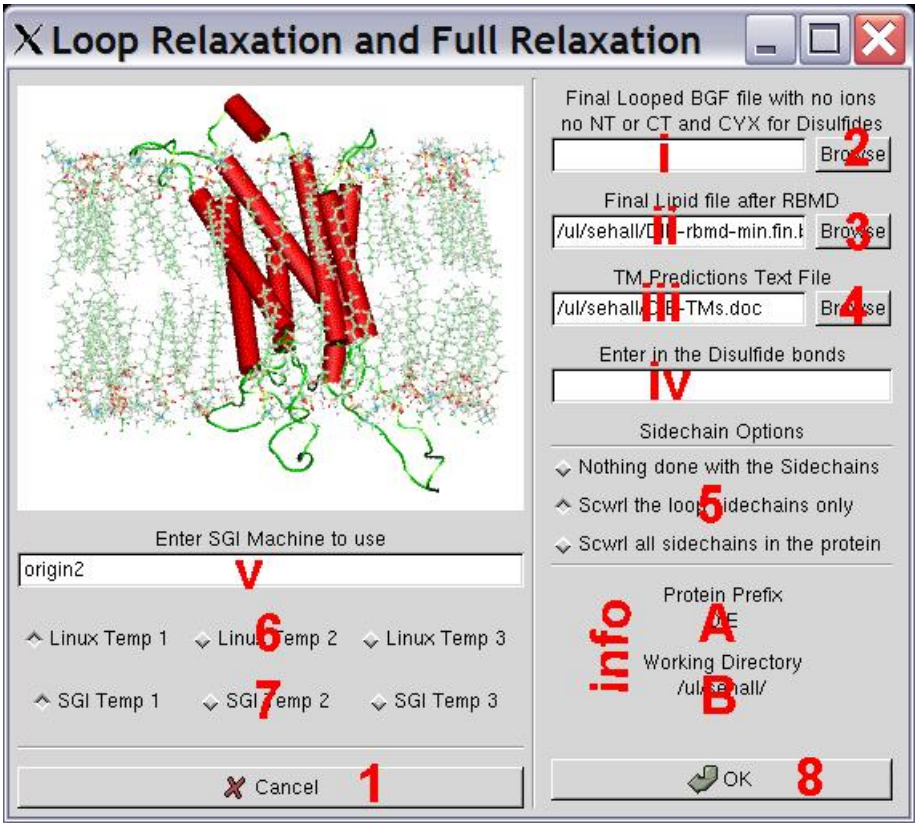
9) Option 5 – Exits the program

10) Fix any charge and forcefield problems. There seems to be a problem with the FF (CYX made S -> S_31 -.07 and CB -> 0.7 where this should be -.08 and .08 and S_3).

11) As a last preparation you will still want to run loop relaxation (Chapter 13) again on the finished file.

Chapter 13 – Loop Relaxation of the Final Structure

This is the final script of the MembStruk methods, and it is used to take the final best structure and make sure it is fully minimized and ready for docking in Cassandra.



Text Boxes:

i) Enter here the final bgf file to be relaxed. With no NT or CT or ions, but with CYX for the cystine residues found in cystine bridges.

ii) This should contain the final lipid file + protein after RBMD (usually called {Prefix}-rbmd-min.fin.bgf).

iii) Enter in here the name and location of the TM Predictions file made in chapter 2.

iv) Enter in here the residues involved in cystine bridges. Each residue number should be numberically ordered with 1 starting as the first residue in TM 1. You enter the numbers in cystine bridge pairs with spaces, so for example 75-148 make a bridge and 138-158 make a bride so you would enter “75 148 138 158”.

v) This textbox should contain the name of the SGI machine that will run Biograf remotely. You must be able to ssh onto this machine via a C-Shell (you must have this machine located in your home directory .ssh/ in the known_hosts file). It is best if you try to ssh into this computer before you run this script.

Information Section:

A) This will contain the prefix that you entered in the textbox in the main GUI window.

B) The information section contains the current working directory as defined in the Main MembStruk GUI window. This is the directory that will have the final output files.

Buttons:

1) Click to close this window.

2) This will open a fileselection window and enter the chosen file into textbox i).

3) This will open a fileselection window and enter the chosen file into textbox ii).

4) This will open a fileselection window and enter the chosen file into textbox iii).

5) Choose the button that corresponds to the desired relaxation method. Scwrl the loop sidechains if running after using whatif after rotational scan, and do a full scwrl after using modeller.

6) These three radio buttons are used to determine which environmental variable to use to define the Linux temp directory. Choosing temp 1 will use $memtmp1 up to temp 3 will use $memtmp3. Only one can be chosen and the default is temp 1. This variable will be used to create a temporary directory ($memtmp?/{Prefix}{unique}) to store files while in use for the script.

7) These three radio buttons are used to determine which environmental variable to use to define the SGI temp directory. Choosing SGI temp 1 will use $memsgi1 up to temp 3 will use $memsgi3. Only one can be chosen and the default is SGI temp 1. This variable will be used to create a temporary directory on the remote SGI machine ($memsgi?/{Prefix}{unique}) to store files while in use for the script.

8) Run the Loop Relaxation script. See section 13.1 for details.

Section 13.1 – Running the Loop Relaxation Script

The loop relaxation script is a C-Shell script that will scrwl the sidechains if desired and then perform a last long minimization to prepare a structure for docking. All these steps are done in presense of lipid structures to help fine tune the positions of the sidechains. The last minimization with lipids fixed for 5000 steps and usually takes a day, and is followed by a everything movable minimization for Cassandra of 500 steps.

Command to run the script:

```
looprelax.script {output final looped bgf file, no
ions/lipids/NT/CT} {lipid + protein after RBMD bgf file}
{TM Predictions doc} {1 = no scwrl/2 = scwrl loops/3 =
scwrl everything} {Prefix} {prefix.ssb – file containing
one residue per line involved in CYX bridges} {SGI Machine}
{Linux temp drive #} {SGI Temp #}
```

Dependant files for the Program:

FF/Biograf.par	bin/loopfasta.pl
/FF/Biograf.cnv	bin/scwrl
FF/Mpsim.par	bin/change_ntct_crg.pl
bin/bgf2fasta-loop.pl	bin/mpsim

Input Files Needed:

Final looped BGF file	- This file is the last bgf file after all other steps have been done.
{Prefix}-rbmd-min.fin.bgf	- This file is used for it's lipid structures and to match the final structure back into the lipid barrel.
{Prefix}.ssb	- This file is created by the GUI as input for the script to give the Cystine bridge imput. Every residue number that you entered in textbox iv is entered in it's own line in the order that you entered them in the GUI.

Output Files from the Program:

{Prefix}.ssb	- This file is created by the GUI as input for the script to give the Cystine bridge imput. Every residue number that you entered in textbox iv is entered in it's own line in the order that you entered them in the GUI.
${prefix}.MembStruk.bgf	- This is the final file ready for Cassandra and docking.
${prefix}-lipid-MFMin.fin.bgf	- File with lipids and protein after all the minimizations have been done, but just before lipids are removed.
${prefix}-lipid-minF.fin.bgf	- This file contains the lipids and protein right after the 5000 step minimization.
${prefix}-matched-lipid.bgf	- This file is right after the scwrl (if chosen) and matching of the protein back onto the bundle.

Running Time for the Program:

Approx. 1 ½ days of time.

Appendix A – Files Produced from MembStruk

TM2ndS Output Files

{prefix}.txt	- This contains the fasta sequence of the original protein to be studied. This file is created after running the Special Tools Button 5, Section 1.6.1.
{prefix}?.out	- These files contain the output from running the complete TM Prediction cycle on the ? region. These files are created after running Special Tools Button 5, Section 1.6.1.
{Prefix}_get_centers_output	- Same as the get_centers_output file just saved under the Prefix name.
{Prefix}_inTMPred2	- The input TMPred file used with this prefix.
{Prefix}_predictions.txt	- Same as the predictions.txt saved under the prefix name.
{Prefix}_WIN12_plot.ps	- Same as the WIN12_plot.ps file saved under the Prefix name.
{Prefix}-Final.pdb	- This file is the final pdb bundle structure that will be used for the rest of the MembStruk protocol. Created from button 1 in the Create Template window (Section 1.5).
1.xpm	- Picture file for the GUI.
areas_before.txt	- Lists the area under the curve for each interval.
blastseq.aln	- This is an .aln file containing the same alignment information as the .pit file, just in a different format.
blastseq.pir	- This is a .pir file containing the alignment of all the sequences.
blastseq?.txt	- These files contain the blast sequences used for each region. These files are created after running Special Tools Button 5, Section 1.6.1.
blastseq_TMcore.ala	- This file is the output of a special high gaps penalty program of the sequences alignment in ala format. This file is used exclusively for the Special Tools Button 2 program. See Section 1.6.
blastseq_TMcore.dnd	- This file is the output of a special high gaps penalty program of the sequences alignment in dnd format. This

	file is used exclusively for the Special Tools Button 2 program. See Section 1.6.
blastseq_TMcore.pir	- This file is the output of a special high gaps penalty program of the sequences alignment in pir format. This file is used exclusively for the Special Tools Button 2 program. See Section 1.6.
bundle.pdb	- This is a structure pdb file used in the creation of the final structure bundle file. Created from button 1 in the Create Template window (Section 1.5).
data##_basechange.txt	- These files contain the graph information for each window size.
data##_nobasechange.txt	- These files contain the graph information for each window size.
data.txt	- This file contains the hydrophobic data from the sequence alignment.
data_after_cap.txt	- Contains the new numbers after capping.
data_before_cap.txt	- Contains numbers for the capping rules.
data_hyd_cent.txt	- Data output file from the getcenters program.
data1.txt	- Same as the data.txt file with the number of lines missing.
delH.macro*	- This is a biograf macro for removing hydrogens.
dreidii322-quanta.cnv*	- This is a biograf forcefield conversion file.
dreidii322-quanta.par*	- This is a biograf forcefield parameters file.
final.pdb	- This is a structure pdb file used in the creation of the final structure bundle file. Created from button 1 in the Create Template window (Section 1.5).
get_centers_output	- This is the output from the getcenters program.
get_centers_output?	- These files contain the get_centers output for each region. These files are created after running Special Tools Button 5, Section 1.6.1.

hel?.pdb	- These are the pdb helix structure files created from Biograf. These files are created during the Create Template button 1 in the Create Template Window (Section 1.5).
hel?-bgf.pdb	- Saved files after the bgf conversion to pdb. Created from button 1 in the Create Template window (Section 1.5).
hel?-final.pdb	- These are the final minimized helix structure files that are used in the alignment of the bundle. Created from button 1 in the Create Template window (Section 1.5).
hel[1-7]	- These files contain the predictions per helix and are used to build the final pdb structure.
HPMCenter.txt	- This file contains the HPM centers used for translation
HPMCenter?.txt	- These files represent the get_centers that were run for each regions prediction. These files are created after running Special Tools Button 5, Section 1.6.1.
info_pred.txt	- input file for the predictions program.
ingnu*	- This is a gnuplot macro file for the sequence window plot.
ingnu_lastline	- This is the end of the gnuplot macros.
ingnu2*	- This is another gnuplot macro more detailed for the sequence window plot.
ingnu3	- Input file for GnuPlot.
ingnu4	- Input file for GnuPlot.
input_first	- This is an input file for a C+ program.
input_interactive	- Gives the input for the interactive program to get all the data files.
inputalign*	- This is an input file for Clustal.
inputalign2	- This input file is used to run a special alignment with high gap penalties. This file is used exclusively for the Special Tools Button 2 program. See Section 1.6.
inTMPred2	- Information needed to run the TM2ndS Fine program.

local_average.txt - Gives the local average used to get the helix interval.

logfile.macro - This is a file created by Biograf that contains a macro of all commands used in the last execution of Biograf in the current directory.

macromin.ctl* - This is a macro file for an mpsim minimization ***(this will need to be edited for systems running in a different directory setup to find the correct forcefield file).***

min.out - This contains the output from mpsim of the last minimization run on the last .bgf helix file. Created from button 1 in the Create Template window (Section 1.5).

original.pdb - This is a structure pdb file used in the creation of the final structure bundle file. Created from button 1 in the Create Template window (section 1.5).

out_clustalw - This is the direct output from running clustalw.

out_clustalw? - ClustalW output for each region. These files are created after running Special Tools Button 5, Section 1.6.1.

output_TM_core.txt - This was altered to tell which core sequences had gaps.

output_TM_core.txt - Contains the middle 15 residues from the predictions.

output_TM_core1.txt - This is the backup copy of the original output_TM_core.txt file after being run through the iterative TM prediction. This file is created after running Special Tools Button 5, Section 1.6.1.

output_TM_hel_extended.txt - Gives the TM predictions with the regions being extended by 20 residues.

output_TM_hel_extended_size.txt - Gives the size of the extended regions.

prediction_numbers.txt - This contains the input needed for the getcenters program.

predictions.txt - The predictions from TMPred are located here. (On the 1st try I get only six helices predicted in this file.)

predictions?.txt - These files contain the predictions found for each region tested. These are the file to use for the final TM prediction. These files are created after running Special Tools Button 5, Section 1.6.1.

PREFIX_temp.txt	- This contains the prefix entered.
premac.macro*	- This is a biograf macro file for starting helix building.
profile_template	- This file contains your sequence and can be modified as described in the header of this file to force the high gap alignment program to extend TM regions so specified. This file is used exclusively for the Special Tools Button 2 program. See Section 1.6.
rhodopsin.fta	- This is my Fasta formatted file containing the rhodopsin sequence
sequence.txt	- This file contains the sequence alignment of all the sequences.
sequence_identities.txt	- This contains the sequence homologies of the blast sequences. (Use this file to determine which sequences to keep in the blastseq.txt file to have a good distribution of sequences used for alignment.)
sequence_identities?.txt	- These files contain the sequence homology information obtained from clustalW for each region. These files are created after running Special Tools Button 5, Section 1.6.1.
temp_base_plot	- GnuPlot setup script information.
temp2.out	- This is the output generated from biograf during the conversion process. Created from button 1 in the Create Template window (Section 1.5).
temphelix.bgf	- This is a temp file to save the bgf conversion of the helix pdb file. This file is part of button 1 in the Create Template Window (Section 1.5).
template.pdb	- This is a structure pdb file used in the creation of the final structure bundle file. Created from button 1 in the Create Template window (Section 1.5).
template-final.pdb	- This is a structure pdb file used in the creation of the final structure bundle file. Created from button 1 in the Create Template window (Section 1.5).
temppirfile	- This is the temp file created to form the blastseq.pir file, this is needed for running TM2ndS coarse and Fine.

tempresultfile	- Another temp file used in the TMPrediction program.
tmp2.bgf	- A temp file for holding a helix structure during conversion from pdb to bgf in the create template function. See Section 1.5.
which_baseline_used.txt	- Gives the baselines used to determine the helical intervals.
WIN12_plot.ps	- PostScript plot of the Hydrophobicity at Window 12.
y	- This file contains numerical output about the sequences.

MembStruk Output Files

${prefix}.MembStruk.bgf - This is the final file ready for Cassandra and docking.

${prefix}-lipid-MFMin.fin.bgf - File with lipids and protein after all the minimizations have been done, but just before lipids are removed.

${prefix}-lipid-minF.fin.bgf - This file contains the lipids and protein right after the 5000 step minimization.

${prefix}-matched-lipid.bgf - This file is right after the scwrl (if chosen) and matching of the protein back onto the bundle.

{HPrefix}.ctl - This file contains the choosen dynamics ctl file to use for individual helix runs with the HPrefix. Once this file is in your directory, you will need to press the setup files button to change it to the correct file if you plan on using the same HPrefix with a different method.

{HPrefix}.dat - This file contains the list of the files produced in a script that is usable for MolecularGL.

{HPrefix}.eng - This file contains all the energies of each snapshot BGF file for the user to choose the best structure.

{HPrefix}.fix - This is a ctl file used for MpSim after the $memstruk variable has been used.

{HPrefix}-fixhelixLv4.10 - This file holds the qsub batch script if a linux cluster is going to be used for the fixhelix.IndvHel script.

{HPrefix}-fixhelixPv4.10 - This is the cluster output file ready for submission for the fixhelix.IndvHEL script.

{HPrefix}-snap[25K-99K].bgf - These files hold the BGF files of the dynamics run at every 1,000 steps for you to choose the best structure.

{Prefix}.abgfR.ps - This file is a postscript view of the bgf file after rotation.

{Prefix}.abgfR.txt - This file contains the text information of the after rotation analysis.

{Prefix}.afterR.ps - This PostScript file contains the visual plot of the data contained in the {Prefix}.afterR.txt file.

{Prefix}.afterR.txt - This file contains the analysis of the pdb template file after translation and rotation.

{Prefix}.beforeR.ps - This PostScript file contains the visual plot of the data contained in {Prefix}.beforeR.txt.

{Prefix}.beforeR.txt - This file contains the analysis of the pdb file after translation and before rotation. See examples below for more details.

{Prefix}.bRbgf.ps - This contains the postscript view of bgf file before rotations to use as a reference.

{Prefix}.bRbgf.txt - This is the text output of the analysis of the bgf file before rotation to use as a reference.

{Prefix}.ctl - This is the ctl file that will run the main MpSim dynamics simulation.

{Prefix}.fix - This is a fixed version of the {Prefix}.ctl file that has the correct $membstruk directory for the FF.

{Prefix}.fixhelix.bgf - This is the final fixhelix file ready to be rotated, it can also contain the changes made to it by the fixhelix.IndvHEL script.

{Prefix}.fixhelix.bgf.bak - This is the backup file of the original fixhelix file. CAUTION: This file will be overwritten if the fixhelix.IndvPUT script is used more than once to place a helix into the bundle file.

{Prefix}.rotmin.bgf - This is the final file after finishing rotminL.script. It contains in it's remarks section the rotations used on it.

{Prefix}.rotmin.out - This contains all the energy and rotational information used by rotmin.

{Prefix}.ssb - This file is created by the GUI as input for the script to give the Cystine bridge imput. Every residue number that you entered in textbox iv is entered in it's own line in the order that you entered them in the GUI.

{Prefix}-[1-5]-afterscwrl.pdb - These files contain the bundle with loops after running scrwl.

{Prefix}-[1-5]-energy.out - This contains the energy output from mpsim for the looped file.

{Prefix}-[1-5]-finalmin.bgf - This is the looped structure after minimization.

{Prefix}-[1-5]-finalmin-done.snap* - This is an output file from mpsim.

{Prefix}-[1-5]-finalmin-done.traj1 - Another output file from mpsim.

{Prefix}-[1-5]-finalstruct.bgf - This is the final cleaned file that you will use for the next step in the method.

{Prefix}-[1-5]-loopmin.bgf - This is the structure after modeler has added loops and been minimized.

{Prefix}-[1-5]-loopmin-done.snap* - This is an output file from MpSim.

{Prefix}-[1-5]-readymin.bgf - This file is the structure minimized and ready for modeler.

{Prefix}-[1-5]-readymin-done.snap* - This file is an output file from MpSim.

{Prefix}-[1-5]-readymin-done.traj1 - This file is an output file from MpSim.

{Prefix}-bundle332.bgf - This file contains the first bundling of the new relaxed helices in fixhelix.

{Prefix}-bundle-ion.bgf - This is the next step in the bundling with counter ions added in fixhelix.

{Prefix}-bundle-min.fin.bgf - This is the bundle file after a full minimization in fixhelix.

{Prefix}-formod.ali - This contains the alignment information for modeler to use.

{Prefix}-H{Helix #}.rotmin.out - Output file that contains all the information from the rotational scan. The 5 degree block of lines, contain zero rotation information as well. Under the energies line of this first block it goes: {energy 0} {energy +5} {energy -5} {saltbridges 0} {saltbridges +5} {saltbridges -5} {HBonds 0} {HBonds +5} {HBonds -5}. For every other block of lines, it looks like this: {energy +deg} {energy -deg} {saltbridges +deg} {saltbridges -deg} {HBonds +deg} {HBonds –deg}.

{Prefix}-H{Helix #}.rotmin.ps - This is a postscript file of the information contained in the output file graphed on a scale of the largest

negative energy score or a 300 scale if there are no negative numbers.

{Prefix}-H{Helix #}.rotmin.ps-3 - This is a postscript file of the information contained in the output file graphed on a scale of the top 1/3 negative energies, or if no negative energies exist, then a scale of 100.

{Prefix}-H{Helix #}-rotminIHv4.10 - Lunux qsub script for running Individual Rotation Scan on a cluster.

{Prefix}-H{helix number}{degree}.IH.bgf - This is a snapshot bgf file of the protein after helix {number} has been rotated {degree} amount, scrwled, and minimized. This file can be chosen to replace the original file if the profile looks better.

{Prefix}-hcenterTRBGFv4.10 - Borg qsub script to run the hcenterTR-bgf.script.

{Prefix}-hcenterTRv4.10 - This file is a PBS script used to run hcenterTR on a linux cluster.

{Prefix}-helix?.bgf - BGF conversion of the unminimized pdf helix file that has been recentered for fixhelix.

{Prefix}-helix?-cm.bgf - The helix file after running helixcm.exe for fixhelix.

{Prefix}-helix?-final.bgf - This is the final helical file that has been matched back to it’s original position and is ready for bundling in fixhelix. This is the file to check first to see if the unusual bends in the final bundle file is due to dynamics or due to scrwl.

{Prefix}-helix?-ion.bgf - The centered helix file with counterions before minimization for fixhelix.

{Prefix}-helix?-min-A.bgf - This is the minimized helical file before Dynamics for fixhelix.

{Prefix}-helix?-neimo.fin.bgf - This if the final structure file after the dynamics simulation for fixhelix.

{Prefix}-helix?-neimo.NM.bgf - This is the chosen snapshot file according to energy that has been converted to a BGF file for fixhelix.

{Prefix}-helix?-neimo-rmion.bgf - Final dynamics helix file with counterions removes and is ready to bundle for fixhelix.

{Prefix}-helix?NM-ion.bgf - This is the chosen Dynamics snapshot with counterions added for fixhelix.

{Prefix}-helix?NM-min.fin.bgf - This is the Dynamics snapshot after minimization in MpSim for fixhelix.

{Prefix}-lipid-min.fin.bgf - A bgf file after merger of the lipid and protein files and a minimization.

{Prefix}-lipid-rbmd.bgf - The bgf file of only the lipids with no protein after RBMD has finished but before scwrl or the final minimization.

{Prefix}-lipid-rbmd.pdb - The pdb file conaining just the lipids after RBMD but before the final minimization and scwrl.

{Prefix}-mod.out - This is the output file from modeler.

{Prefix}-modeller.script - This is a C-Shell script that will run the makeloops.pl script on a SGI machine. It can be edited to run the makeloops-stepwise.pl script by simply replacing the makeloops.pl with makeloops-stepwise.pl.

{Prefix}-ptnonly.bgf - This file contains the input file with no HETATMs.

{Prefix}-rbmd.fin.bgf - This is the final file after the long RBMD run finishes, but before scwrl or minimization.

{Prefix}-rbmd.snap* - MpSim snapshots of the RBMD simulation.

{Prefix}-rbmd.swl.pdb - Final file from RBMD for use with whatif or SwissPDB Viewer to add loops.

{Prefix}-rbmd-min.fin.bgf - This is the final file after RBMD and scwrl. This is the file to use for the Modeller scripts.

{Prefix}-RBMDv4.10 - Borg qsub script generated for use on a linux cluster to run the RBMD script.

{Prefix}-ready.bgf - This is the cleaned up file ready for modeler.

{Prefix}-Rot.bgf - This is the final output file that has been rotated and scrwled.

{Prefix}-rot.pdb - This file contains the new template that has been translated and rotated.

{Prefix}-rotminv4.10 - This executable file contains the qsub script to run rotminL.script on a linux cluster.

{Prefix}-scw.scwrl.bgf - This is the bundle file right after scwrl for fixhelix.

{Prefix}-TMs.doc - This is the default name (any can be chosen) to contain the information about the TM prediction.

{Prefix}-withloop.* - These are modeler ouput files.

afterssbonds-{Prefix}-[1-5].pdb - These files are right after modeler and the cysbridge bonds have been made.

ANNEAL_LOOP.script - C-Shell script to run the Annealing macros.

degree.dat - This file contains the order and degrees that will be tested. This can be edited before running the script.

degree-IH.dat - This file contains the degrees to rotate the helix. It starts with all degrees from 0 to 360 in 5 increments, but can be modified to restart a script that has failed.

dreidii322-* - These are forcefield files used with modeler and biograf.

helix.dat - This file contains the order and helices to run fine rotations on. This can be edited before running the script.

HPMCenter.txt - This file contains the HPM centers used for translation.

HPREFIX.TXT - This file holds the Prefix used for the Fixhelix.IndvHel and Fixhelix.IndvPut C-shell scripts.

logfile.macro - This is a file created by Biograf that contains a macro of all commands used in the last execution of Biograf in the current directory.

make_*.macro - These macros are used to create the Cys bridges in option 2 of the EC2 folding program.

min_IH_m2.ctl - This file contains the ctl file for MpSim to run the first full protein minimization.

min_IH_m2.fix - This is an edited form of the min_IH_m2.ctl file that contains corrected directory information.

min_IHrotmin.ctl	- This is a ctl file for MpSim to run the fixed protein helix movable minimization.
min_IHrotmin.fix	This is an edited form of the min_IHrotmin.ctl file with corrected directory information.
min_rotmin.ctl	- This is the MpSim input file for the second protein fixed, helix movable minimization and can be edited before running the script.
min_rotmin.fix	- This is the min_rotmin.ctl file updated with the $membstruk variable.
min_rotmin_m2.ctl	- This is the MpSim input file for the first minimization with the whole protein movable, and can be edited before the script runs.
min_rotmin_m2.fix	- This is the min_rotmin_m2.ctl file updated with the $membstruk variable.
min2_good_anneal*.macro	- These are macros for biograf from EC2 option 4.
mine_*.macro	- These macros are used to create the Cys bridges in option 2 of the EC2 folding program.
mine2*.macro	- These are macros to start the annealing simulation for the EC loops.
model-{Prefix}-withloop.top	- This file contains the modeler script for adding the top loops.
out_anneal_open_loops	- Output from option 1 in the EC2 program.
out_run_em	- This is the output file from EC2 Option4.
out-{HPrefix}.fix	- This file contains the output from the script fixhelix.IndvHel if run outside a cluster.
out-{HPrefix}.fixP	- This file contains the output from the fixhelix.IndvPUT script.
out-{Prefeix}.RBD	- This file contains the output from the RBMD script.
out-{Prefix}.fix	- This is the output file for a non-borg job for fixhelix.

out-{Prefix}.rot	- This contains the output from the hcenterTR-bgf C-Shell script.
out-{Prefix}.rot	- This file contains all the output from running the hcenterTR script.
PREFIX.txt	- This file contains the current working prefix name for the MembStruk GUI.
run_em.script	- This will setup the jobs to run in biograf for EC2 option 4.

MembStruck Program Files

$membstruk directory Files

comparison.exe	- This is the comparison GUI program used for MembComp. See MembComp manual for details.
comparison.script	- This is a C-shell script to run the comparison program from MembStruk. `MEMBSTRUK.exe {$membstruk/pix}`
MEMBSTRUK.exe	- This file is a C-shell script that will run the membstruk executable.
membstruk.sh	- This is the file to source in your .cshrc to setup all needed environmental variables. Sets up Biograf, Modeller, MembStruk, and TMPred2ndS.
MEMBSTRUK-410.exe	- This is the main GUI executable created from Glade. It requires one input: the picture directory. You run this by typing: `MolGL.script {moleculeGL script to run, Default is {HelPrefix}.dat}`
MolGL.script	- This is a C-shell script that will start up on the current machine an xterm window and from that window the moleculeGL program. This script requires one input of the moleculeGL script to run. This is used to run the moleculeGL program to look at the Fixhelix individual helices.

$membstruk/FF directory Files

Biograf.cnv	- This is the current Biograf conversion file used for all scripts. Currently this is equal to the dreidii322-quanta.cnv file.
Biograf.par	- This is the current Biograf parameters file used in all scripts. Currently this is equal to the dreidii322-quanta.par file.
Biograf-qeq.cnv	- This is the current Biograf qeq conversion file used in the Fixhelix script. Currently this is set to be qeq-neutral-vaid.cnv.
dreidii322-dz-chrm22.cnv	- This is David's Charm22 conversion file for Biograf.
dreidii322-dz-chrm22.par	- This is David's Charm22 parameters file for Biograf.

dreidii322-mpsim.par	- This is the MpSim parameters file that uses Dreiding 322 forcefield set up for the MpSim program.
dreidii322-mpsim-rbmd.par	- This is the MpSim parameters file that was developed by Wely, Vaidehi, and Spencer for use in the RBMD script.
dreidii322-quanta.cnv	- This is the conversion file for Biograf that takes pdb files created from Quanta and converts them correctly into Dreiding 322 BGF files. (Works best if the pdb has no H's).
dreidii322-quanta.par	- This is the parameters file for Biograf that will convert a Quanta pdb file into a Dreiding 322 BGF file.
Mpsim.par	- This is the default MpSim parameters file used in all MpSim runs. This is currently set to equal dreidii322-mpsim.par.
Mpsim-RBMD.par	- This is the MpSim parameters file used for the RBMD ctl file. Currently this is set to equal the dreidii322-mpsim-rbmd.par file.
qeq-neutral-vaid.cnv	- This file was developed by Vaid and John Wendel for the conversion of pdb files into a QeQ neutral state for all residues in the Dreiding 322 format.

$membstruk/compare Directory Files

HsprotL.exe	- This is the rotational code that rotates individual helices. Written in Fortran 77. See the MembComp manual for more details.
autofind.script	- This C-shell script is used in the MembComp GUI to find the beginnings and endings of TM BGF files. See MembComp manual for more details.
cmp2txt.awk	- This is an awk program file to compare two MembComp.txt output files. Command: `awk –f $membstruk/compare/cmp2txt.awk {reference txt file} {txt file to compare}`
compare2ps.script	- This C-shell script is used to merge two PostScript files into one file that contains both data sets in different colors. See the MembComp manual for more details.

compareL.exe - This is the main (Linux) executable that runs the comparision program used in the MembComp GUI. This was written in Fortran 77. See the MembComp manual for more details.

eisenberg.dat - This is the data file that contains the eisenberg hydrophobic index used to find the hydrophobic moments.

saltbridgeL.exe - This is an executable that will find the number of salt bridges contained in the BGF file (will not look for HSP). Written in Fortran 77. See MembComp manual for more details.

transhelix.exe - This is the Fortran 77 program used to translate individual helices. See the MembComp manual.

$membstruk/bin Directory Files

bgf2fasta.pl - This Perl script will take a BGF formatted file and output the fasta sequence. Command: `bgf2fasta –x/-X {bgf file}`, the little –x option will print the sequence in lowercase and the –X option will print the sequence in uppercase.

bgf2fasta-loop.pl - This is a special version of the bgf2fasta.pl script that is used to get fasta information for scrwl that will leave the lipids and CYX residues alone.

bgflipidscwrl - This C-Shell script will take a file containing lipids and run scwrl on it and use the lipids as a bump check for scwrl. Command: `bgfscwrl {input .bgf} {output file w/o .bgf} {SGI machine to run biograf} {usable temp dir on the SGI machine, no “/” at the end}`

bgfscwrl - This C-shell script will take a bgf file and run scwrl on the entire protein and return a bgf file. Command: `bgfscwrl {input .bgf} {output file w/o .bgf} {SGI machine to run biograf} {usable temp dir on the SGI machine, no “/” at the end}`

BiPeakanalysis.exe - This is the executable used to find the two highest peaks in the center of a TM prediction. See section 1.4.4 for more details.

bundlecm.exe	- This fortran 77 program will take a bgf file and place the center of mass at the 0,0,0 coordinant. Command: `bundlecm.exe <enter> {input BGF file} <enter> {output BGF file} <enter>`
change_ntct_crg.pl	- This Perl script will take a BGF file and check the charges on the N and C terminus to make sure they are neutral and fix any errors. Command: change_ntct_crg.pl {input bgf file} {output bgf file}
helixcm.exe	- This fortran 77 program will take a bgf file and place the center of mass at the 0,0,0 coordinant. Command: `bundlecm.exe <enter> {input BGF file} <enter> {output BGF file} <enter>`
HelRot.script	-This is a C-Shell script that allows the user to manually rotate a helix in a bundle if MembComp is unavailable to them (BS edition). HelRot.script {file to rotate .bgf} {ouput file prefix} {helix number to rotate} {degrees to rotate, + is clockwise from extracellular looking down}
HsprotL.exe	This is the rotational code that rotates individual helices. Written in Fortran 77. See the MembComp manual for more details.
liblapack.so	- Needed Linear Algebra library for MpSim to run.
loopfasta.pl	- This is a special version of the bgf2fasta.pl script. This will take as an argument the {Prefix}-TMs.doc file and use that to determine where the loops are located and only allow the loops (ignoring CYX) to be Capitalized for sidechain replacement.
mpsim	- This is the mpsim executable used in all scripts.
mpsimdata/	- This contains old startup data used for MpSim. Set your environmental variable to this directory.
Mpsim-linux	- This is a Linux compiled version of MpSim on a Red Hat 9 system. It should be linked/copied as the bin/mpsim file when on linux machines.
mpsim-unix	- This is a unix compiled version of MpSim for SGI machines, and should be linked/copied as the bin/mpsim file when on SGI machines.

nacl-append	- This Fortran 77 Linux program will print out the bgf file inputted with counter ions in the right places. Command: `nacl-append {input bgf file} > {output bgf file}`
nacl-append-iris	- This Fortran 77 SGI program will print out the bgf file inputted with counter ions in the right places. Command: `nacl-append-iris {input bgf file} > {output bgf file}`
nacl-appendL	- This Fortran 77 Linux program will print out the bgf file inputted with counter ions in the right places. Command: `nacl-append {input bgf file} > {output bgf file}`
pdb2fasta.pl	- This Perl script will take a PDB formatted file and output the fasta sequence. Command: `pdb2fasta –x/-X {pdb file}`, the little –x option will print the sequence in lowercase and the –X option will print the sequence in uppercase.
README	- This file will contain notes on the usage and maintenance of the files in the bin directory.
rotate3L.exe	- This is a fortran 77 program that will take a bgf file and rotate it + and – the degrees specified and output 3 files: one centered about the origin, and 2 rotated +/- after being centered about the origin.
RotminCompare.script	- This is a C-Shell script that will take a bgf file and run it through the compare program. Its uses are limited and is currently not supported.
scwrl	- This is the Free unlicensed version of scwrl3.0 used for sidechain replacement.
scwrl-unix	- This is the free unlicensed version of scrwl2.1 on the unix platform. This is used only for the modeler scripts.
snap2bgf	- This C+ program will take a MpSim snapshot file and convert it back into a bgf file. Unix compiled. Command: `snap2bgf {snapshot file} {starting MpSim bgf file} > {output file name}`
snap2bgfL	- This C+ program will take a MpSim snapshot file and convert it back into a bgf file. Unix compiled. Command:

```
snap2bgf  {snapshot file} {starting
MpSim bgf file} > {output file name}
```

sparse_pdbL - This Fortran 77 Linux program will take a template file and split it into individual helix pdb files.

transhelix-res.exe - This is a Fortran 77 Linux program that will take a seven helical BGF file and translate each helix from it's geometric center to a specified residue center. This is used in the hcenterTR.script to translate the template onto it's hydrophobic centers.

$membstruk/extras Directory Files

membcomp.version - This file contains the version history of the MembComp program.

membstruk.version - This file contains the version history of the MembStruk program.

xv - This is a free program used to view PostScript files in the MembStruk and MembComp GUIs.

$membstruk/hcenterTR Directory Files

HPMCenter.txt - This is a default template file used as reference to show what this file should look like.

White.dat - This file contains the White scale that can be used in the comparison program.

eisenberg.dat - This file contains the Eisenberg hydrophobic scale to be used with the comparison program.

hcenterTR-bgf.script - This C-Shell script will run the rotation of a bgf file that has only helices setup in the standard Membstruk format fro BGF files. See section 6.1 for details.

hcenterTR.script - This C-shell script will run the translation and rotation of a template pdb file. See section 3.1 for detail on how to run this script and it's dependences.

$membstruk/fixhelix Directory Files

NEIMOfnd.awk - This is a C-shell script that will search through the output of a MpSim dynamics run and find the lowest energy and

	convert the saved snap file into a bgf file for use. Command: `NEIMOfnd.awk {out.neimo file} {prefix before .snap????} {Neimo start file}`
fixhelixL.DYNqeq	- One of four main Fixhelix C-shell scripts to run on the template. See section 4.1 for more details.
fixhelixL.Dynamics	- One of four main Fixhelix C-shell scripts to run on the template. See section 4.1 for more details.
fixhelixL.IndvHel	- A C-shell script used to run dynamics on individual helices that have problems in the final {prefix}.fixhelix.bgf file. See section 5.1 for more details.
fixhelixL.IndvPUT	- A C-shell script used to place a finalized helix structure back into the fixhelix file. This is often used to fix scrwl problems. See section 5.2 for more details.
fixhelixL.NQEQ	- One of four main Fixhelix C-shell scripts to run on the template. See section 4.1 for more details.
fixhelixL.script	- One of four main Fixhelix C-shell scripts to run on the template. See section 4.1 for more details.
neimo_Dyn.ctl	- This is the default ctl file used for the Dynamics, Charged fixhelix.
neimo_Dynqeq.ctl	- This is the default ctl file used for the Dynamics, QeQ fixhelix.
neimo_NQEQ.ctl	- This is the default ctl file used for the Neimo, QeQ fixhelix.
neimo_orig.ctl	- This is the default ctl file used for the Neimo, Charged Fixhelix.

$membstruk/rotmin Directory Files

Default.CD.rot	- This file contains the default starting rotations for the Rotmin.CombDiv script.
HPMCenter.txt	- This is an example of the format for the HPMCenter.txt file that is read by MembStruk.

Template-IH.ps	- This is a template file of the syntax used for the postscript files in the output from the rotmin.IndvHel script.
awkchains	- This file is an input file for awk to add chains to a bgf file.
awkrotminIH	- This is an awk input file to read a rotmin output file and convert it into a postscript graph. It will show only the negative energies, or failing to have negative energies it will show the lowest energies within 300 kcal.
awkrotminIH-3	- This ia an awk input file to read a rotmin output file and convert it into a postscript graph that keeps the top 1/3.
degree-IH.dat	- This contains the default rotations for rotmin.IndvHel to use (-180 to 180 in 5 degree increments).
degree.dat	- This contains the default order to check rotations in a select combination for rotmin.script.
dreidii322-mpsim.par	- This is an obsolete forcefield file kept for older versions.
helix.dat	- This file contains the order that the helices are rotated for the rotmin.script.
min_CD_m2.ctl	- This file contains the default MpSim settings used for the 1st protein movable minimization for rotmin.CombDiv.
min_CDrotmin.ctl	- This file contains the default MpSim settings used for the 2nd only focused helix movable minimization for rotmin.CombDiv.
min_IH_m2.ctl	- This file contains the default MpSim settings used for the 1st protein movable minimization for rotmin.IndvHel.
min_IHrotmin.ctl	- This file contains the default MpSim settings used for the 2nd only focused helix movable minimization for rotmin.IndvHel.
min_rotmin.ctl	- This file contains the default MpSim settings used for the 2nd only focused helix movable minimization for rotmin.script.
min_rotmin_m2.ctl	- This file contains the default MpSim settings used for the 1st protein movable minimization for rotmin.script.

rotminL.CombDiv - This will rotate several combinations. See section 9.1 for details.

rotminL.IndHel - This will rotate an individual helix 360 and produce a graph. See section 8.1 for details.

rotminL.script - This will run through a series of rotations keeping the lowest in energy. See section 7.1 for details.

$membstruck/rotmin/extras Directory Files

awkrotmin.force300 - This is an awk program file that will read a rotmin.script output and printout a postscript file that will look at the lowest energies within 300 kcal of each other. `awk –f $membstruck/rotmin/extras/awkrotmin.force300 {output file} > {postscript file}`

awkrotmin.force300-3 - This is an awk program file that will read a rotmin.script output and printout a postscript file that will look at the lowest energies within 100 kcal of each other. `awk –f $membstruck/rotmin/extras/awkrotmin.force300-3 {output file} > {postscript file}`

awkrotminIH2Comp – This is an awk program file that will read two rotmin.IndvHel outputs and compare them on the same graph (300 Kcal scale). `awk –f $membstruck/rotmin/extras/awkrotminIH2Comp {output file} {output file} > {prostscript file}`

awkrotminIH2Comp-3 - This is an awk program file that will read two rotmin.IndvHel outputs and compare them on the same graph (100 Kcal scale). `awk –f $membstruck/rotmin/extras/awkrotminIH2Comp-3 {output file} {output file} > {prostscript file}`

awkrotminIH360ALL - This is an awk program that will read the seven helical rotmin.IndvHel outputs and place the graphs on a MembComp output (300 Kcal scale). `awk –f $membstruck/rotmin/extras/awkrotminIH360ALL {MembComp postscript output} {helix 1 IndvHel output} {helix 2 IndvHel output} {helix 3 IndvHel output} {helix 4 IndvHel output} {helix 5 IndvHel output} {helix 6 IndvHel`

```
output} {helix 7 IndvHel output} >
{output postscript file}
```

awkrotminIH360ALL-3 - This is an awk program that will read the seven helical rotmin.IndvHel outputs and place the graphs on a MembComp output (100 Kcal scale). `awk –f $membstruck/rotmin/extras/awkrotminIH360ALL {MembComp postscript output} {helix 1 IndvHel output} {helix 2 IndvHel output} {helix 3 IndvHel output} {helix 4 IndvHel output} {helix 5 IndvHel output} {helix 6 IndvHel output} {helix 7 IndvHel output} > {output postscript file}`

Cont-rotm.IndHel - This is a script that will continue where a previous rotmin.IndvHel script failed. (Beta no support available)

rotminL.portablescript - This is a script that was written to run rotmin.script on any platform. (Beta with bugs, no support)

rotmin-totalrot.script - This is another Beta script that is obsolete.

$membstruk/rotmin/old Directory Files

rotminL.07032004.Aventis - This is the original rotmin that was used for Aventis with the translation bug.

$membstruk/RBMD Directory Files

RBMDL.script - This will run the rigid body molecular dynamics in MpSim. See chapter 10.

lipidbarrelgood-cm.bgf - This is a file containing the 52 lipid structures around the center of the axis for the system.

lipidbarrelgood.bgf - This contains the 52 lipids in their original formation from Wely.

$membstruk/RBMD/extras Directory Files

Min.Borg - This was a test linux cluster file created for RBMDL.script.

Restart.RBMDl - This was created to restart a failed RBMD job. It is no longer supported.

Restart.borg - This is a PBS script file created to run Restart.RBMDl on a linux cluster.

$membstruk/RBMD/old Directory Files

RBMDL-3.5.script - an older copy of the RBMD script.

RBMDL.3.5 - an older copy of the RBMD script.

$membstruk/loops Directory Files

looprelax.script - This is a C-Shell script that will take a final bgf file with loops and minimize the loops and scrwl the bundle and then perform a final long minimization. See chapter 13 for details.

pdb2CYXbgf.script - This script was designed for those who use whatif. It will take the final structure and minimize the loops, scwrl the loops and keep the cystine bridges intact. To use type:

```
pdb2CYXbgf.script {output pdb} {whatif doc} {Prefix} {prefix.ssb – this file will have res res each line for the residues found in a cystine bridge} {SGI Machine} {Linux temp drive} {SGI Temp}
```

$membstruk/loops/modeller/ Directory Files

README - This file contains changes and modifications to the original scripts.

bgf2pdb.pl - This will take a bgf file and convert it to a pdb file.

bgfrenum.pl - This will renumber the residues of the bgf file to be sequential.

biovar - This script will run biograf given a forcefield prefix and a macro.

chainA.pl - This will add chains to the bgf file.

change_ntct_crg.pl - This will take a bgf file and make sure the end charges are correctly placed.

dreidii322-quanta.cnv - Biograf forcefield file.

dreidii322-quanta.par - Biograf forcefield file.

fixcharges.pl - This file will correct any charge inequalities created in the pdb conversion to bgf.

makeloops-stepwise.pl - Main script to use modeler individually. See section 11.1.

makeloops.pl	- Main script to use modeler simultaneously. See section 11.1.
mod6v2_iris	- This is the modeler symbolic link.
modeller-template.macro	- This is a template for creating the modeler macro.
mpsim.pl	- This will setup a .ctl file for MpSim and run it.

$membstruk/loops/whatif Directory Files

awkwhatif	- An awk script that the TM Predictions Text File button (see chapter 2) uses.
renumbgf2pdb	- This is a simple script to renumber a bgf file and set it up as a pdf file ready for whatif.
renumb-pdb	- This will renumber a pdf file after whatif has added loops.
whatif.com	- C-Shell script set up to find the whatif executable and run it. This is created when whatif is setup.
whatif-doc	- This is the C-Shell script that runs the awkwhatif script to produce the text file in Chapter 2.
whats	- A simple script to start whatif and setup the memory and graphics. It only works if the file to use is less than 6 characters long.

Program References

AVGB Solvation - Zamanakos, G. (2001). A fast and accurate analytical method for the computation of solvent effects in molecular simulations. PhD Thesis, Physics, California Institute of Technology, Pasadena, CA

Blast

Altschul, S. F., W. Gish, W. Miller, W. E. Myers, and D. J. Lipman. (1990). Basic local alignment search tool. *J. Mol. Biol.* 215, 403–410.

Altschul, S. F., T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.* 25, 3389–3402.

ClustalW - Thompson, J. D., D. G. Higgins, and T. J. Gibson. (1994). Clustal-W—improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.* 22, 4673–4680.

Crystal Rhodopsin - Palczewski, K., T. Kumasaka, T. Hori, C. Behnke, H. Motoshima, B. Fox, I. Trong, D. Teller, T. Okada, R. Stenkamp, M. Yamamoto, and M. Miyano. (2000). Crystal structure of rhodopsin: a G-protein-coupled receptor. *Science*. 289, 739–745.

HBPLUS – McDonald, I. K., and Thorton, J. M., (1994). Satisfying Hydrogen Bonding Potential in Proteins. *JMB* 238, 777-793.

MembStruk - Vaidehi, N., Floriano, W. B., Trabanino, R., Hall, S. E., Freddolino, P., Choi, E. J., Zamanakos, G., and William A. Goddard II, (2002). Prediction of structure and function of G protein-coupled receptors. *Proc. Nat. Acad. Sci.* 99(20), 12622-12627.

MpSim - Lim, K-T, Brunett, S., Iotov, M., McClurg, R.B., Vaidehi, N., Dasgupta, S., Taylor, S. and Goddard III, W.A. (1997). Molecular dynamics for very large systems on massively parallel computers: The MPSim program. *J Comput. Chem.* 18(4), 501-521.

Mpsim & Biograf Forcefield - Mayo, S. L., Olafson, B.D. and Goddard III, W.A. (1990). DREIDING: a generic force field for molecular simulations. *J. Phys. Chem.* 94(26), 8897-8909.

NEIMO

Mathiowetz, A.M., Jain, A., Karasawa, N., and Goddard III, W.A. (1994). Protein Simulations using Techniques Suitable for Very Large Systems: the Cell Multipole Method for Nonbond Interactions and the Newton-Euler Inverse Mass Operator Method for Internal Coordinate Dynamics. *Proteins* 20, 227.

Vaidehi, N., Jain, A., and Goddard III, W.A. (1996). Constant Temperature Constrained Molecular Dynamics: The Newton-Euler Inverse Mass Operator Method. *J. Phys. Chem.* 100(25), 10508.

Modeller

Sali, A., and Blundell, T. L. (1993). Comparative protein modelling by satisfaction of spacial restraints. *J. Mol. Biol.* 234, 779-815.

Fiser, A., Do., R. K., and Sali, A., (2000). Modeling of loops in protein structures. *Protein Science* 9, 1753-1773.

Marti-Renom, M. A., Stuart, A., Fiser, A., Sanchez, R., Melo, F., and Sali, A. (2000). Comparative protein structure modeling of genes and genomes. *Annu. Rev. Biomol. Struct.* 29, 291-325

Scwrl - Bower, M., Cohen, F. E., and Dunbrack, R. L., Jr. (1997). Prediction of protein side-chain rotamers from a backbone-dependent rotamer library: a new homology modeling tool. *J. Mol. Biol.* 267, 1268-1282

Spdbv - Guex, N., Diemand, A., Peitsch, M. C., and Schwede T. (2003) spdbv 3.7. http://us.expasy.org/spdbv/

TM2NDS & EC_LOOP_SIM - Trabanino R.J., Hall, S.E., Vaidehi N., Floriano W.B., and Goddard III, W.A. (2004). First Principles Predictions of the Structure and Function of G-Protein Coupled Receptors: Validation for Bovine Rhodopsin. *BioPhys. J.* 2004 86, 1904-1921

Whatif - Vriend, G. (1990). WHAT IF: A molecular modeling and drug design program. *J. Mol. Graph.* 8(1), 52-56.