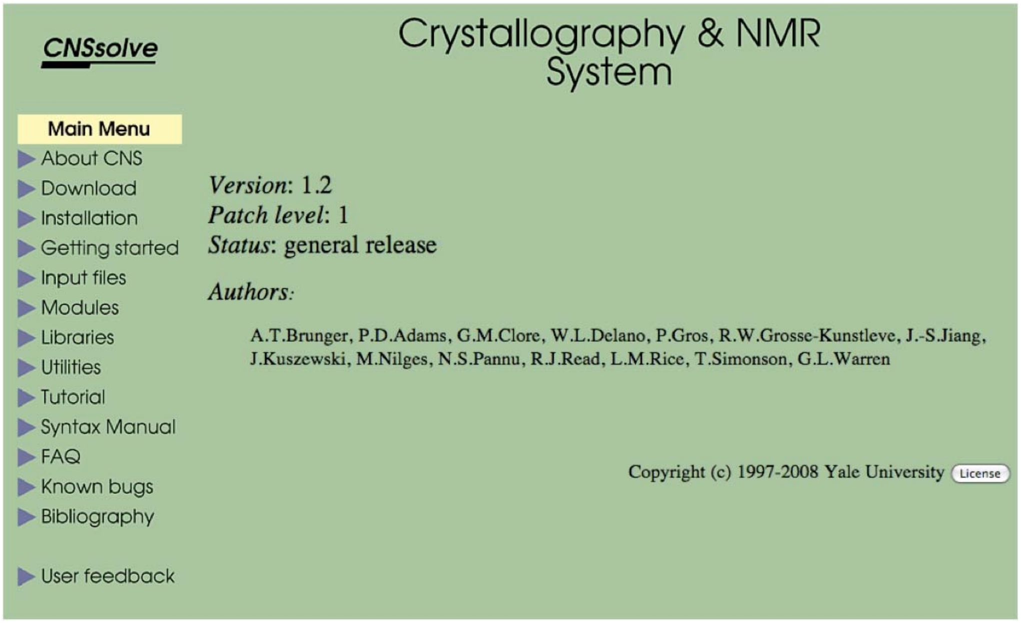


Crystallography & NMR System

Version: 1.2
Patch level: 1
Status: general release

Authors:

A.T.Brunger, P.D.Adams, G.M.Clore, W.L.Delano, P.Gros, R.W.Grosse-Kunstleve, J.-S.Jiang, J.Kuszewski, M.Nilges, N.S.Pannu, R.J.Read, L.M.Rice, T.Simonson, G.L.Warren

Copyright (c) 1997-2008 Yale University

http://cns-online.org/v1.21/

Structure Calculation using CNS

DO THE FOLLOWING, IF YOU HAVE NOT ALREADY DONE SO:

First, look in your home directory to see if there is a subdirectory named “cns”:

```
[your-user-name@localhost ~]$ pwd
     /home/your-user-name
[your-user-name@localhost ~]$ ls -F
     Desktop/    example/    Molecules/   cns/
[your-user-name@localhost ~]$
```

The “cns” subdirectory should contain a file called “anneal.inp” and two subdirectories, “reference” and “restraints”:

```
[your-user-name@localhost ~]$ cd cns
[your-user-name@localhost ~]$ pwd
     /home/your-user-name/cns
[your-user-name@localhost ~]$ ls -F
     anneal.inp   reference/   restraints/
[your-user-name@localhost ~]$
```

If you do not have the “cns” subdirectory, or its contents do not appear to be correct, you will have to create the directory, download the contents from the course website (as a “tar” file), and extract the “tar” file in your home directory. You may need assistance from the course instructors to do this.

You should now be ready to begin.

Introduction

In this exercise, we will use available NMR-derived restraint information (NOE-based distance restraints, hydrogen bond restraints and dihedral angle restraints) to calculate the structure of a protein/peptide complex using simulated annealing/restrained molecular dynamics as implemented in the program CNS (Crystallography and NMR System). The complex is a mixed disulfide between a human thioredoxin mutant (C35A, C62A, C69A, C73A) and a 13 residue peptide comprising its target site in human Ref-1 (residues 59-71 of the P50 subunit of Nfkb). We will then use the program PyMol to visualize the resulting calculated structure.

CNS is a "newer version" of the program X-PLOR. Details about the programs can be found at the web sites (below). Also, the X-PLOR manual ("X-PLOR: Version 3.1, A System for X-ray Crystallography and NMR, by Axel T. Brünger, Yale Press) is an extremely good reference, and probably essential for those of you interested. It is also available online (below). A version of X-PLOR maintained by the NIH (X-PLOR-NIH) is also available (below):

CNS:	*http://cns-online.org/v1.21/*
X-PLOR-NIH:	*http://nmr.cit.nih.gov/xplor-nih/*
X-PLOR manual online:	

http://www.csb.yale.edu/userguides/datamanip/xplor/xplorman/htmlman.html

Using a web browser (Internet Explorer, Safari, Mozilla) you can go to *http://cns-online.org/v1.21/* and poke around a bit if you like before you begin the tutorial. However, we will be going to this web site at the end of the tutorial and using some of the tools there.

Exercise 1: Calculating the structure of a thioredoxin-peptide complex

Getting started

CNS can be run in an interactive mode, or in a non-interactive mode.

-**Click on the X11 icon** to get an X11 window. All commands are entered from the command line in the X11 window.

-To run in the interactive mode, simply type **cns** (return). You'll see the following:

```
[your-user-name@localhost ~]$ cns

============================================================
|                                                          |
|            Crystallography & NMR System (CNS)            |
|                        CNSsolve                          |
|                                                          |
============================================================
    Version: 1.2 at patch level 1
    Status: General release
============================================================
    Written by: A.T.Brunger, P.D.Adams, G.M.Clore, W.L.DeLano,
         P.Gros, R.W.Grosse-Kunstleve, J.-S.Jiang,
         J.Kuszewski, M.Nilges, N.S.Pannu, R.J.Read,
         L.M.Rice, T.Simonson, G.L.Warren.
    Copyright (c) 1997-2008 Yale University
============================================================
    Running on machine: c122-114 (x86_64/Linux,64-bit)
    Program started by: your-user-name
    Program started at:  18:14:09 on 14-Apr-2010
============================================================

 FFT3C: Using FFTPACK4.1

CNSsolve>
```

In this mode, all of the output of the program is to the screen. You enter commands sequentially and the program responds to individual commands. You can exit the program by typing **stop** (return) at the CNSsolve prompt.

-in the non-interactive mode, you designate an input file (macro) and an output file. The default for output (i.e. if you don't specify an output file) is to the screen. Examples are shown below (don't type any of these). We will use this mode later in this exercise.

cns <macro.inp> (no output file designated: output is to the screen)
cns <macro.inp> output.out (output is to the file "output.out")

Generating the molecular topology file (.mtf)

-go into the “reference” directory (…../cns/reference)

```
cd cns
cd reference
```

-type **ls** (return). Your directory contents should look like this:

```
generate_extended.inp  trx_a.seq
generate_seq.inp       trx_b.seq
```

One of the first tasks that we need to perform is to generate the molecular topology file. The molecular topology file contains information about molecular connectivity/’covalent topology’ for the molecule that you are working with. For your system (molecule or molecules), this file must be generated. For a protein, the amino acid sequence is necessary as input for generation of this file. In our case, we have two proteins; thioredoxin and a short peptide. The two sequence files are **trx_a.seq** (105 amino acids) and **trx_b.seq** (13 amino acids), respectively. You can use **BBEdit** or any other text editor to look at these files (they should look like those shown below….be careful NOT to make any changes to these files). For a shortcut, type **cat trx_a.seq** (return) and **cat trx_b.seq** (return) to list the contents to the screen (rather than using a text editor).

```
MET VAL LYS GLN ILE GLU SER LYS THR ALA
PHE GLN GLU ALA LEU ASP ALA ALA GLY ASP
LYS LEU VAL VAL VAL ASP PHE SER ALA THR
TRP CYS GLY PRO ALA LYS MET ILE LYS PRO
PHE PHE HIS SER LEU SER GLU LYS TYR SER
ASN VAL ILE PHE LEU GLU VAL ASP VAL ASP
ASP ALA GLN ASP VAL ALA SER GLU ALA GLU
VAL LYS ALA THR PRO THR PHE GLN PHE PHE
LYS LYS GLY GLN LYS VAL GLY GLU PHE SER
GLY ALA ASN LYS GLU LYS LEU GLU ALA THR
ILE ASN GLU LEU VAL
```

```
PRO ALA THR LEU LYS ILE CYS SER TRP ASN
VAL ASP GLY
```

-the file **generate_seq.inp** is a CNS macro for generating a molecular topology file for our molecules. Use **BBEdit** (Mac) or **gedit** (Linux), ***or any other text editor***, to look at this macro (do NOT make any changes). Scroll down and you'll see the following lines:

```
{* protein sequence file *}
{===>} prot_sequence_infile_1="trx_a.seq";
{* segid *}
{===>} prot_segid_1="A";
{* start residue numbering at *}
{===>} renumber_1=1;

{* protein sequence file *}
{===>} prot_sequence_infile_2="trx_b.seq";
{* segid *}
{===>} prot_segid_2="B";
{* start residue numbering at *}
{===>} renumber_2=106;
```

Our system is broken into two "segments", with segment identifiers ('segid') "A", and "B". Segment A (residues 1-105) is thioredoxin, and segment B (residues 106-118) is the peptide. Scroll down a bit further and you'll see the following lines:

```
{========================== disulphide bonds ==============================}

{* Select pairs of cysteine residues that form disulphide bonds *}
{* First 2 entries are the segid and resid of the first cysteine (CYS A). *}
{* Second 2 entries are the segid and resid of the second cysteine (CYS B).
*}
{+ table: rows=8 numbered
   cols=5 "use" "segid CYS A" "resid CYS A" "segid CYS B" "resid CYS B" +}

{+ choice: true false +}
{===>} ss_use_1=true;
{===>} ss_i_segid_1="A"; ss_i_resid_1=32;
{===>} ss_j_segid_1="B"; ss_j_resid_1=112;
```

These lines allow for disulphide bonds between cysteine residues in proteins or between protein segments. In our case, these lines define a disulfide bond between residue 32 (cysteine) in thioredoxin and residue 112 (another cysteine) in the peptide.

Now, scroll down a bit farther and you'll see the following lines:

```
{======================== generate parameters =============================}

{* hydrogen flag - determines whether hydrogens will be retained *}
{* must be true for NMR, atomic resolution X-ray crystallography
   or modelling.  Set to false for most X-ray crystallographic
   applications at resolution > 1A *}
{+ choice: true false +}
{===>} hydrogen_flag=true;
```

It is important that the **`hydrogen_flag=true`** for NMR, etc.

-now we can generate our .mtf file. Type ONE of the following:

```
        cns  <generate_seq.inp   (then return, of course)
    OR
        cns  <generate_seq.inp> generate_seq.out   (then return, of course)
```

-if you type the first command, the output of the program will be shown on the screen….it will come very quickly and you'll not be able to read it. For the most part it is not all that interesting unless something goes wrong, and then it is useful for diagnostic purposes. If you type the second command, the output goes into the file 'generate_seq.out', so you can look at it with an editor like **BBEdit/gedit**.

-the above commands will generate a file called **trx.mtf**. That is the molecular topology file for our system of two protein molecules connected by a disulfide bond. You can use **BBEdit/gedit** to look at this file (be careful NOT to change anything). The first information you will see is information concerning the identity of each particular atom, including segment identifier, (amino acid) residue number, atom name and type of atom, and atomic charge and mass. If you scroll way down in the file, you'll find information about how each atom is connected to other atoms in the system.

Generating initial (extended) coordinates

-you should still be in the “reference” directory

-type **ls**. Your directory contents should look like this:

```
generate_extended.inp  trx.mtf         trx_b.seq
generate_seq.inp       trx_a.seq
```

In order to begin a restrained molecular dynamics/simulated annealing calculation, a starting structure is needed. A starting structure should have good local geometry and, for calculations at the initial stages of a structure determination, should not be biased in any way with respect to tertiary structure. In our case, we will choose as a starting structure an extended polypeptide chain (actually, two extended chains in our case).

-the file **generate_extended.inp** is a CNS macro for generating extended polypeptide chains as starting structures for our calculations. Use **BBEdit/gedit** to look at this macro (do NOT make any changes). Scroll down and you’ll see the following lines:

```
{======================= molecular structure =========================}

{* structure file(s) *}
{===>} structure_file="trx.mtf";
```

The macro uses as input the trx.mtf file that we generated previously. Scroll down a bit further and you’ll see the following lines:

```
{=========================== output files ============================}

{* output coordinates *}
{===>} output_coor="trx_extended.pdb";
```

The output file (coordinate file) generated by this macro will be called **trx_extended.pdb**, and will be (approximately) in the standard PDB format, representing the atomic coordinates of each of the atoms in our molecule(s).

-now we can generate our initial coordinates (.pdb file). Type ONE of the following:

cns <generate_extended.inp (return)

OR

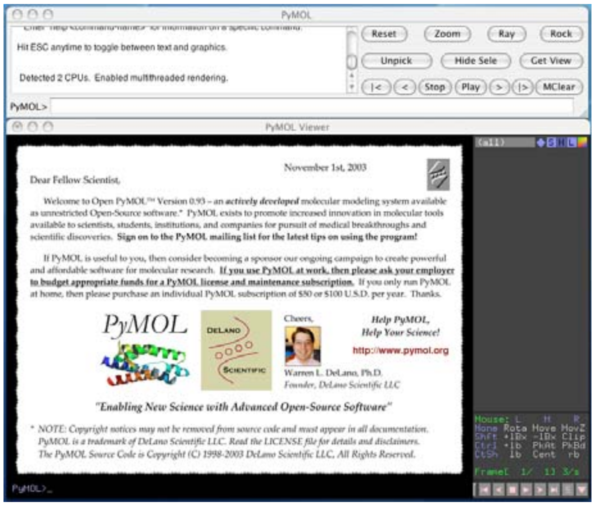
cns <generate_extended.inp> generate_extended.out (return)

-as before, if you use the first command, the output of the program goes to the screen. If you use the second command, it goes into a file (generate_extended.out) that you can look at. Either way, the file called **trx_extended.pdb** is generated. This is the coordinate file for our starting structure (extended conformation, good local geometry)

Visualization with Pymol

-you should still be in the "**reference**" directory

It is important to be able to quickly and easily visualize the three dimensional structures of proteins from their PDB (coordinate) files. For us, at this point, it is important to verify that we have indeed generated an extended polypeptide chain (actually, two extended polypeptide chains). PyMol is a very nice program that allows you to do this.



Hit ESC anytime to toggle between text and graphics.

Detected 2 CPUs. Enabled multithreaded rendering.

November 1st, 2003

Dear Fellow Scientist,

Welcome to Open PyMOL™ Version 0.93 – an ***actively developed*** molecular modeling system available as unrestricted Open-Source software.* PyMOL exists to promote increased innovation in molecular tools available to scientists, students, institutions, and companies for pursuit of medical breakthroughs and scientific discoveries. **Sign on to the PyMOL mailing list for the latest tips on using the program!**

If PyMOL is useful to you, then consider becoming a sponsor our ongoing campaign to create powerful and affordable software for molecular research. **If you use PyMOL at work, then please ask your employer to budget appropriate funds for a PyMOL license and maintenance subscription.** If you only run PyMOL at home, then please purchase an individual PyMOL subscription of $50 or $100 U.S.D. per year. Thanks.

Cheers,

Warren L. DeLano, Ph.D.
Founder, DeLano Scientific LLC

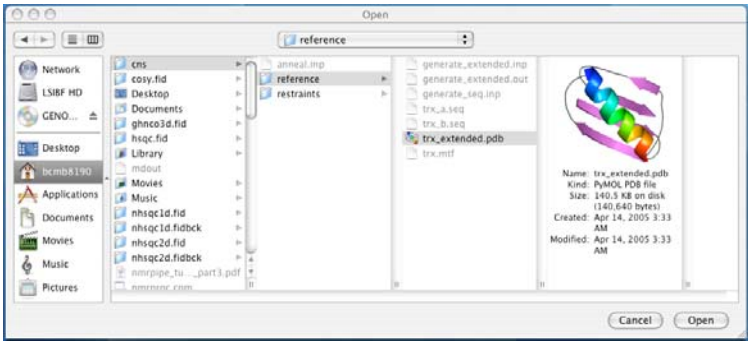
Help PyMOL, Help Your Science!

http://www.pymol.org

"Enabling New Science with Advanced Open-Source Software"

* *NOTE: Copyright notices may not be removed from source code and must appear in all documentation. PyMOL is a trademark of DeLano Scientific LLC. Read the LICENSE file for details and disclaimers. The PyMOL Source Code is Copyright (C) 1998-2003 DeLano Scientific LLC. All Rights Reserved.*

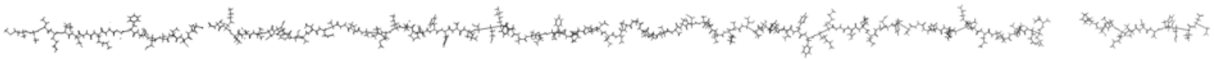
-**Mac**: click on the PyMol icon on the Dock to start the program.
-Linux: type "pymol" on the command line and return
- From the menu at the top, select **File**, then **Open** then select the file /home/your-user-name/cns/reference/**trx_extended.pdb** and click **Open**



Name: trx_extended.pdb
Kind: PyMOL PDB file
Size: 140.5 KB on disk (140,640 bytes)
Created: Apr 14, 2005 3:33 AM
Modified: Apr 14, 2005 3:33 AM

(displays left and above are from Mac)

-You should see your extended chain structure appear (probably with black background), similar to that shown below.



-when we calculate the three-dimensional structure of our protein(s), we'll return to PyMol and learn more about how to manipulate molecules on the screen. For now, from the menu select **PyMol (Mac) or File (Linux)** then **Quit**.

Calculating a structure using restrained molecular dynamics and simulated annealing

-you should still be in the “reference” directory

-change to the **restraints** directory (type **cd ../restraints** (return)). The directory contents should show three files:

```
trx_dihed_rama.tbl   trx_noe_all.tbl    trx_noe_hbond.tbl
```

-these files contain the distance (`trx_noe_all.tbl`), dihedral (`trx_dihed_rama.tbl`), and hydrogen bond (`trx_noe_hbond.tbl`) restraints for our molecule(s). Use **BBEdit/gedit** to look at `trx_noe_all.tbl` (do NOT make any changes). The first few lines are shown below:

```
!EDIT_HISTORY
!      A(963)  nh_noetotal.pck   Fri Sep 15 16:42:03 1995

!M1

!V2
assign (resid 2 and name HG2#) (resid 3 and name HN) 4.0 2.2 1.5        !#A 762  2.78e+05
assign (resid 2 and name HB) (resid 3 and name HN) 4.0 2.2 1.0  !#A 760  2.82e+05
assign (resid 2 and name HA) (resid 3 and name HN) 2.5 0.7 0.4  !#A 34  2.36e+06
assign (resid 2 and name HG1#) (resid 3 and name HN) 2.5 0.7 0.9        !#A 23  1.27e+06
assign (resid 2 and name HG2#) (resid 46 and name HN) 4.0 2.2 1.5       !#A 637  1.85e+05
assign (resid 2 and name HG1#) (resid 56 and name HN) 3.0 1.2 1.2       !#A 348  8.33e+05

!K3
assign (resid 3 and name HB#) (resid 3 and name HN) 2.5 0.7 0.4 !#A 22  1.45e+06
assign (resid 3 and name HA) (resid 3 and name HN) 3.0 1.2 0.5  !#A 21  7.75e+05
assign (resid 3 and name HB#) (resid 4 and name HN) 4.0 2.2 1.0 !#A 74  3.87e+05
```

The “assign” lines stipulate NOE based distance restraints. For instance, the first “assign” statement describes a restraint between the protons (# is a wildcard) on gamma carbon 2 (HG2#) of residue 2 (resid 2) and the amide proton (HN) of residue 3 (resid 3). The interpretation of the three numbers (4.0 2.2 1.5) that follow depends on the restraining function, but in general the first is a measure of the distance expected between the two groups specified, and the second and third are subtracted and added from the first to give lower and upper bounds, respectively. Anything following an exclamation mark (!) is a comment and is ignored by CNS.

-use **BBEdit/gedit** to look at `trx_dihed_rama.tbl`. This file contains restraints for the dihedral angles ϕ and ψ. The first few lines are shown below:

```
!remark phi angle constraints

 !!   v2
 assign (resid  1  and name c ) (resid  2  and name n )
      (resid  2  and name ca) (resid  2  and name c )   1.0  -125.0 25.0 2
 !!   k3
 assign (resid  2  and name c ) (resid  3  and name n )
      (resid  3  and name ca) (resid  3  and name c )   1.0  -152.0 20.0 2
```

The first "assign" statement describes the ϕ angle for residue 2; the torsion angle for rotation around the C^{α}-N bond. The positions of the four nuclei indicated can be used to define this angle. The numbers following the statement indicate an energy constant (1.0), the value for the angle in degrees (-125°), the range around the restrained angle (± 25°) and the exponent for the particular restraining function that is used (in this case, 2).

-the final restraint file in this directory is `trx_noe_hbond.tbl`. This file contains restraints for hydrogen bonds. The format is identical to the NOE based distance restraints in the `trx_noe_all.tbl` file.

-change to the parent (main cns) directory by typing **cd ../** (return)

-the directory contents should look like this:

```
anneal.inp   reference   restraints
```

-"reference" and "restraints" are the directories that we've been working in. The file **anneal.inp** is the input file for the simulated annealing/restrained molecular dynamics calculations. Use **BBEdit/gedit** to look at this file (be careful NOT to change anything):

```
{* parameter file(s) *}
{===>} par.1="CNS_TOPPAR:protein-allhdg.param";
{===>} par.2="";
{===>} par.3="";
{===>} par.4="";
{===>} par.5="";

{* structure file(s) *}
{===>} struct.1="./reference/trx.mtf";
{===>} struct.2="";
{===>} struct.3="";
{===>} struct.4="";
{===>} struct.5="";

{* input coordinate file(s) *}
{===>} pdb.in.file.1="./reference/trx_extended.pdb";
{===>} pdb.in.file.2="";
{===>} pdb.in.file.3="";
```

The “parameter files” are standard files found in the CNS program directory that the macro reads into the program. The “structure files”, in this case, consist of our single .mtf file, **trx.mtf** (note that the path to the file is defined relative to the location of anneal.inp). The “input coordinate files” in our case consist of our initial extended starting structure file **trx_extended.pdb**. So, this is how our molecular topology and starting structure information are found and read by the program.

-scroll down further. As you scroll, you’ll scroll past “refinement parameters”, “torsion dynamics parameters”, “parameters for high temperature annealing stage”, “parameters for the first slow-cool annealing stage”, “parameters for a second slow-cool annealing stage”, and “parameters for final minimization stage”. There are many adjustable parameters here that control the molecular dynamics calculations and minimization routines. The CNS manual and the XPLOR manual serve as pretty good references for these parameters. For our calculations, we are using pretty standard values.

-if you scroll down further you’ll see “noe data”, and you’ll see our NOE restraint file `trx_noe_all.tbl` there. Scroll down further and you’ll see our hydrogen bond and dihedral angle files also. Keep scrolling and you’ll see the following:

```
{====================== input/output files =========================}

{* base name for input coordinate files *}
{===>} pdb.in.name="";

{* base name for output coordinate files *}
{===>} pdb.out.name="trx_structure";
```

The coordinate file will be called **trx_structure** (actually, it will be named **trx_structure_1.pdb** because of convention that the program uses).

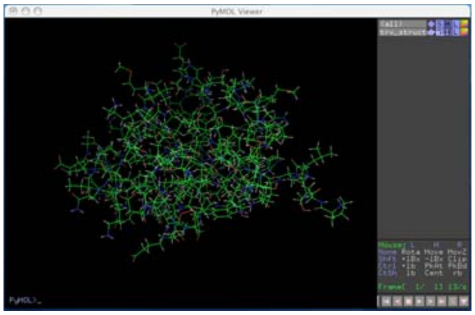
-so, all we have to do now to calculate our structure is to enter the following command:

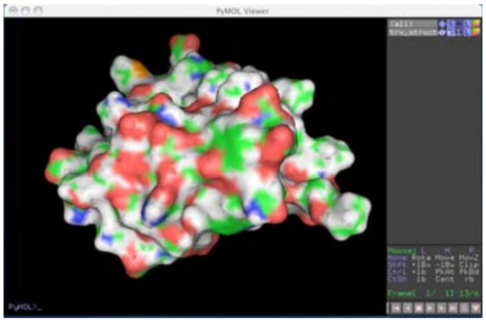
cns <anneal.inp> anneal.out (return)

-it will take your computer a few minutes to calculate a structure. CNS creates the file **anneal.out** immediately and starts dumping output into this file. If anything goes wrong during the calculation, this output can help to diagnose the problem. Also, there is a lot of very important information in this file concerning the calculated structure including restraint violations, energies, etc. When you are doing calculations “for real”, you should *never* delete this file. When the calculation finishes, a file called “**trx_structure.pdb**” will be created. This file is the coordinate file (in .pdb format) for our calculated structure. We can look at our structure using this file and PyMol.

-Start PyMol as before. As before, from the menu at the top, select **File**, then **Open** then select the file /home/your-user-name/cns/**trx_structure.pdb** and **Open**. The molecule will appear as a "line figure" (shown at the right).

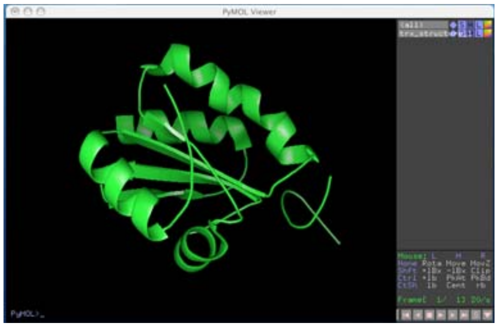
-in the upper right hand corner of the PyMol graphics window, the buttons with the letters "S", "H", and "L" stand for "show", "hide", and "label". You can show/hide a ribbon, a cartoon (tube), spheres (van der Waals/cpk drawing), a surface (shown to the right), etc. The "hide" commands will turn these options off.



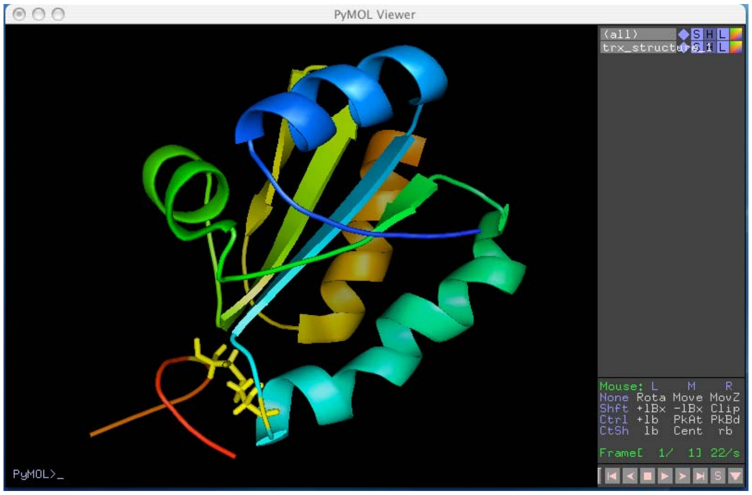


-We'll create a fancy picture showing secondary structure, etc. as follows. First, return to the initial line figure display (hide all other options, show lines).

-next, in the upper PyMol text window, enter the command **util.ss all** (return). This will assign secondary structure elements. Then enter the command **show cartoon, all** (return). Finally, enter **hide lines, all** (return). You should have a ribbon diagram as shown at the right. Both molecules are green, with secondary structural elements (helices/sheets) drawn.



-now, we'll color the molecules and display the disulfide bond. Using one of the multicolored buttons next to the "L" button, choose "**spectrum**" and "**rainbows**". Next, in the upper PyMol text window, enter the command **show sticks, (resi 32, 112)** (return), and then enter the command **color yellow, (resi 32, 112)** (return).

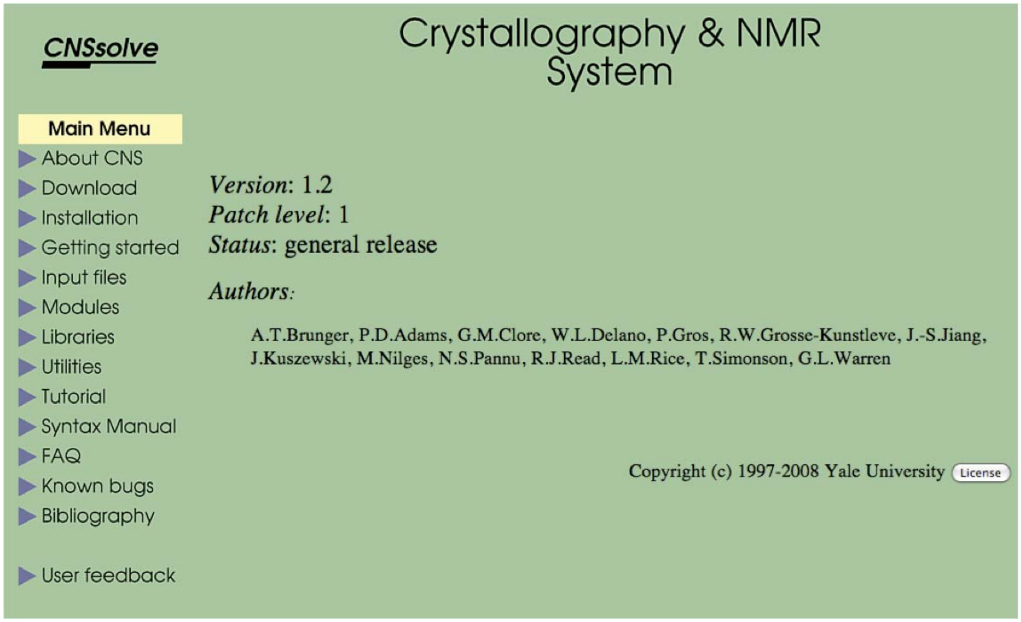


-there are several PyMol tutorials available on the web and other PyMol resources. Once you've completed the CNS tutorial, feel free to invoke a web browser and go find some of these and continue exploring PyMol.

Creating new CNS macros and editing existing ones

CNS has a convenient web-based utility for creating new CNS macros and for editing existing ones. This utility can be accessed via the CNS web site (*http://cns-online.org/v1.21/*) For the exercises above, we simply used existing macros. Now, we'll go through the process of creating a new one and then editing an existing one so that you'll know how to do this should you have occasion to need to.

-with a web browser (Mozilla, Safari, Internet Explorer, etc.) go to *http://cns-online.org/v1.21/*)



Crystallography & NMR System

Version: 1.2
Patch level: 1
Status: general release

Authors:

A.T.Brunger, P.D.Adams, G.M.Clore, W.L.Delano, P.Gros, R.W.Grosse-Kunstleve, J.-S.Jiang, J.Kuszewski, M.Nilges, N.S.Pannu, R.J.Read, L.M.Rice, T.Simonson, G.L.Warren

Copyright (c) 1997-2008 Yale University

There are lots of useful things here. You can look through them at your leisure. For instance, click **Tutorial** and then use the scroll bar to scroll down to **NMR Tutorials**:

NMR Tutorials:

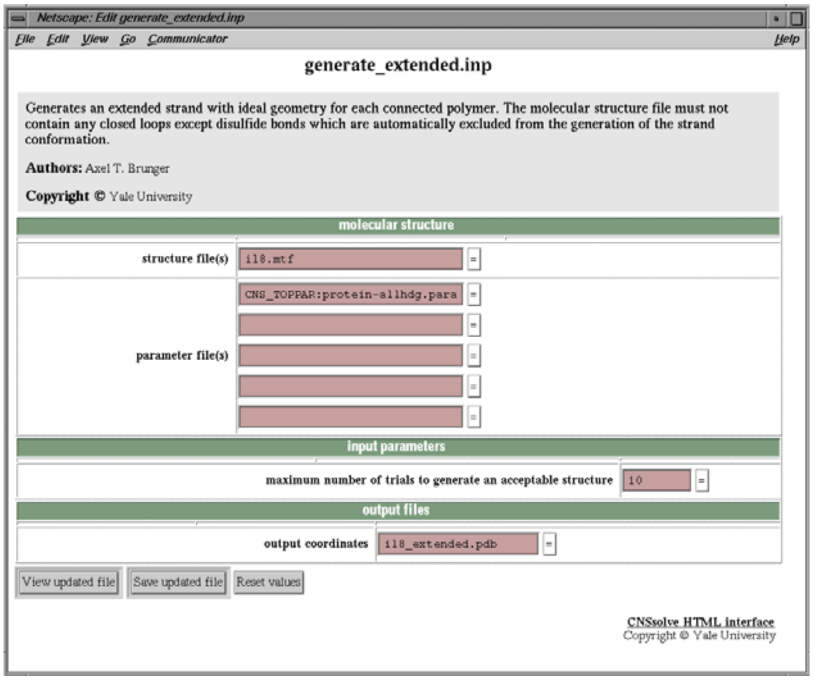
- Initial Template Generation
  - Generating the Molecular Topology
  - Generating initial Extended Coordinates
- Structure Calculation
  - Simulated Annealing
  - Distance Geometry Simulated Annealing

Most of the exercise that we performed today was taken from these tutorials (as you'll see if you click on the links).

You'll also notice, that at the bottom of each page, at the left side, is a link **"Main Page"** to return you to the main (home) page. Return to the main page now.

-first, we’ll generate a new “generate_extended.inp” CNS macro. On the main page, click on **Input Files**. A long list of macros will appear under subheadings. The first subheading is “General”. If you scroll down a bit, you’ll see “generate_seq.inp”, for instance. The listing for “generate_extended.inp” is way down at the bottom of the list under the subheading “NMR”. Once you find it, at the right you’ll see 3 buttons: “View”, “Edit”, and “Edit+Help”. Select (click) **Edit+Help**.

-the utility (new window) for generating a “generate_extended.inp” macro will then appear (right). It is pretty simple to enter the appropriate information in this case (enter the appropriate .mtf file, and then enter the name that you’d like to give the output coordinates). The parameter files, in most cases, are standard files that are accessed from the CNS program directories, and do not need to be changed. You’ll notice a small white button with an “=” sign on it beside each text entry window. If you click on one, a small “help” window will appear telling you what goes in the text window.


Netscape: Edit generate_extended.inp
File Edit View Go Communicator
Help

generate_extended.inp

Generates an extended strand with ideal geometry for each connected polymer. The molecular structure file must not contain any closed loops except disulfide bonds which are automatically excluded from the generation of the strand conformation.

Authors: Axel T. Brunger

Copyright © Yale University

molecular structure

structure file(s) `i18.mtf` =

parameter file(s) `CNS_TOPPAR:protein-allhdg.para` =

input parameters

maximum number of trials to generate an acceptable structure `10` =

output files

output coordinates `i18_extended.pdb` =

CNSsolve HTML interface
Copyright © Yale University

-enter **abc.mtf** in the “structure files” window. Enter **abc.out** in the output coordinates window. Use the **Save updated file** button to save a copy of this file in your cns directory (name it **test.inp**). After you’ve saved the macro, from this window select **File** then **Close**. This will take you back to the **Input Files** page.

-now we’ll learn how to edit a file that has already been created. At the very top of the “Input Files” page, you’ll see a subheading “Edit File”, and a text window (“Enter the name of your CNSSolve input file”). Enter the name of your file (**test.inp**: note, you’ll need the full path to the file, like /21/users/urbauer/cns/test.inp) in this window, then click **Edit+Help**. Your macro will appear, so that you can change/edit the macro, etc.