Justin T. Douglas
1/21/09

A short guide to NMR structure calculations using Xplor-NIH.

The purpose of this guide is help people get started using Xplor-NIH for NMR structure calculations. It is not an exhaustive guide, nor is it intended to be. Much more information can be found at http://nmr.cit.nih.gov/xplor-nih/doc/current/xplor/.

This guide uses the classic X-PLOR scripting language, rather than the new python interface. People planning to use Xplor-NIH as a significant tool in their research would do well to learn the python interface. A good resource for the python interface can be found at http://nmr.cit.nih.gov/xplor-nih/doc/current/

The concept

Xplor-NIH is a highly sophisticated computer program that provides an interface between computational bioinformatics and experimental structural biology. For our purposes, we will consider Xplor-NIH as a tool to minimize a hybrid energy function.

$$E_{hybrid} = E_{chem} + \omega_{\exp} \cdot E_{\exp}$$

where

E_{chem} = a geometric force field

$E_{\exp}$ = experimental derived restraints (NOE, torsion angles, RDC, CSA, etc.)

$\omega_{\exp}$ = weighting factor

The geometric force field and experimental derived restraints are functions of the relative positions of every atom in the molecule. Hence we can think of minimizing the hybrid energy function as finding the configuration of atoms that satisfies both our experimental restraints and standard bond length, angles, etc. encoded in the geometric force field.

There are several techniques available for minimization. We will focus on Powell minimization and molecular dynamics. Powell minimization can be thought of as finding the local minimum. Molecular dynamics can be thought of as finding the global minimum.

To implement these techniques (and everything else Xplor-NIH is capable of) we will use scripts. Scripts are just a series of commands which the program implements. These scripts are just text files containing a series of Xplor-NIH commands. To execute a scripted named “script.inp”, one would type at the command line

```
%xplor < script.inp > script.out
```

This command opens the program xplor and feeds in the commands from the file “script.inp”. This command also redirects the output of the script from the screen to a file named “script.out”.

Justin T. Douglas
1/21/09

A Detailed Example – Calculating the NMR structure from a set of NOE and dihedral restraints downloaded from the PDB.

In this example we will walk though how to calculate the NMR solution structure (a PDB file) from a NOE and dihedral restraint tables downloaded from the BMRB. I will use BMRB entry 4245 (ubiquitin in reverse micelles from Josh Wand's lab). The restraints were downloaded from the BMRB NMR restraint grid, PDB id 1g6j.

The structure calculation will consist of 5 steps. 1) Creating the PSF file from the sequence; 2) Calculating an initial extended structure; 3) Calculating a family of structure using *ab initio* simulating annealing; 4) Refinement of the family of structures; 5) Selection of "successful" structures;

Each of these steps will use a script. Because these steps are used commonly, we can use the X-PLOR tutorial scripts as a template to build our own scripts. The tutorial scripts can be found in the directory xplor-nih-X.XX/tutorial/nmr. As a warning to new users, these scripts are considered obsolete. While these scripts work well for the type of NMR data that was cutting-edge 20 years ago (*e.g.* small protein domains), it is likely that modern application will require newer scripts.

1) Creating the PSF file from the sequence.

The first step is to create the Protein Structure File (PSF) from the protein sequence. Despite the name, this file does not contain the coordinates of the molecule (what is usually thought of as the structure), but rather consists of "topology" of the protein. Xplor-NIH uses this file to know what atoms are connected each other, important torsion angles, bond lengths, etc. This is often the most difficult step in structure calculations.

The PSF file can be calculated from a pdb file or a sequence file. In this tutorial I will create the PSF file from the sequence, because for *de novo* structure calculation you would not have a pdb file available for your protein.

Consider the following script ("gen_psf_from_seq.inp")

```
topology
      @TOPPAR:topallhdg.pro
end

segment
      name="    "
      chain
            @TOPPAR:toph19.pep
            sequence @@seq.txt end
      end
end

write structure
      output=col1.psf
end

stop
```

Let's consider what this script does in a bit of detail. First it executes a "topology" statement. The topology statement is where Xplor-NIH gets the information about the residues that comprise the macromolecule, including the atoms, connectivity, bond angles, etc. Rather than enter all this information in for every new script, we call the file topallhdg.pro. This file lives in the folder xplor-nih-

X.XX/toppar. The TOPPAR part is a shortcut to this directory. The “end” statement closes the topology block.

Next the “segment” statement is executed. The segment statement is where Xplor-NIH generates the molecular structure. The “name” statement can be used to put the molecule name at the end of each line of any PDB files generated from the molecular structure (in this case it is set to nothing). The “chain” statement is where Xplor-NIH receives the sequence of residues. The file toph19.pep is a macro that defines the peptide bond. Next the script executes the “sequence” statement, which reads in the protein sequence from the file seq.txt. There are “end” statements to close the sequence, chain and segment blocks. Finally the “write structure” statement writes the structure file just created, to the file “col1.psf”. The “end” statement closes the write structure block. The “stop” statement exits the script.

IMPORTANT POINT: Just because your script runs and you get a PSF file, does not mean that you are out of the woods. The real trick is to make sure the nomenclature of your PSF file matches the nomenclature in your restraint tables. I have provided a script “check_psf.inp” to help us find any inconsistencies. We can run this script and check the output file using the command

```
%grep '%' check_psf.out
```

If there are any lines like the following:

```
%NOESET-ERR: error in selection - no atoms spec.
```

then there are inconsistencies between the PSF file and the restraint table. Fixing these sorts of problems can be a major headache. The main tool used to fix these inconsistencies is the “vector do” statement. The syntax of this statement is

vector do (name=“what you want to change it to”) (name what it is)

for instance

```
vector do (name="H1") (name HT1)
```

changes every instance of HT1 in your PSF file to H1.

In the case of 1g6j NMR restraints, we find that every restraint raises an error. Clearly there is something wrong with our restraint tables. Consider a line from the NOE restraint table.

```
assi
   (( segid "   A" and resid 1    and name  HA  ))
   (( segid "   A" and resid 2    and name  HN  ))

      2.365     0.565     0.565
```

The likely culprit is the “segid” statement. At no point have we defined the segid, hence Xplor-NIH cannot find these atoms in the PSF file. We have two options. We could edit the NOE and dihedral angle restraint tables and remove the segid statements. Perhaps a more pedagogical solution is to set the segid to A. Consider the modified file “gen_psf_from_seq.inp”

```
topology
      @TOPPAR:topallhdg.pro
end

segment
```

```
     name="    "
     chain
          @TOPPAR:toph19.pep
          sequence @@seq.txt end
     end
end

vector do (segid="   A") (all)

write structure
     output=ubi.psf
end

stop
```

In this case we use a “vector do” statement to set the segid for every atom in the molecule. This change enables us to read in the restraints with no errors.

2) Calculating an initial extended structure.

To minimize our hybrid energy function, we need a starting structure. An extended starting structure can be calculated. We will use the file “generate_template.inp” in the nmr tutorial directory as a template. All the X-PLOR tutorial scripts use the symbol “{====>}” to alert the user to important parts of the script that might need to edited. I have edited these lines of the script and run it to generate the following PDB file.

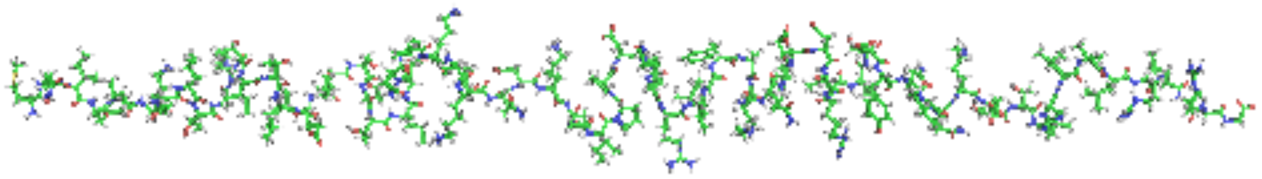


Figure 1. Extended structure calculated using “generate_template.inp”

3) Calculating a family of structure using *ab initio* simulating annealing.

Next, a family of 10 structures will be calculated using simulated annealing molecular dynamics to find the configuration of atoms that minimizes the hybrid energy function. We sill use the script “sa.inp” from the nmr tutorial directory. Edit the appropriate lines. There is one spot that is a bit tricky and I will describe in more detail.

(from original file)

```
noe
{====>}
   nres=3000             {*Estimate greater than the
actual number of NOEs.*}
   class all
{====>}
   @g_protein_noe.tbl                          {*Read
NOE distance ranges.*}
end

{====>}
@g_protein_dihe.tbl                      {*Read
dihedral angle restraints.*}
```

(edited file)

```
noe
{====>}
   nres=10000            {*Estimate greater than the
actual number of NOEs.*}
   class all
{====>}
   @noe.tbl                          {*Read NOE
distance ranges.*}
end

{====>}
restraints

  dihedral

  reset
```

```
   @dihe.tbl                    {*Read dihedral angle
restraints.*}

end
```

Let's compare the two files. The "noe" statement allows us to read in the NOE restraints from the file "noe.tbl". The "nres" statement defines the maximum number of restraints in the file, and the "class" statement enables us to set-up different NOE classes, if we chose (here we do not). The "end" statement closes the noe block. In the original file, control is passed to the file "g_protein_dihe.tbl", which live in the nmr tutorial directory and contains Xplor-NIH statements. The file "dihe.tbl" does not contain these Xplor-NIH statements, so to be more consistent with the logic of the "sa.inp" script, I have added the "dihedral" statement to the script.

This script runs for ~ 6 minutes on my MacBook Pro with 2.4 GHz Intel Core 2 Duo Processor and 4 GB 667 MHz DDR2 SDRAM. 8 of 10 calculated structures of no violations and total energy less than 150 kcal.

4) Refinement of the family of structures.

The next step is to refine the family of structures generated by simulated annealing. We will use the script "refine.inp" in the nmr tutorial directory. At this point we can run refine.inp iteratively until there are no more improvements in the energy. To do this we edited the input and output file names defined in the script

Initial script

```
{====>}
{*Filename(s) for embedded coordinates.*}
   evaluate
($filename="sa_"+encode($count)+".pdb")

{====>}
{*Name(s) of the family of final
structures.*}
evaluate
($filename="refine_1_"+encode($count)+".p
db")
```

Updated for 2nd run

```
{====>}
{*Filename(s) for embedded coordinates.*}
 evaluate
($filename="refine_1_"+encode($count)+".p
db")

{====>}
{*Name(s) of the family of final
structures.*}
evaluate
($filename="refine_2_"+encode($count)+".p
db")
```

I did a total of 4 rounds of refinement to minimize the average total energy.

5) Selection of "successful" structures

Finally, we must select which of our calculated and refined structures are to be included in the final bundle. In the case of 1g6j NMR restraints, this step is not critical, because all structures are acceptable, however, for most cases some structures will be stuck in local minima and should be rejected. We use the script "accept.inp" in the nmr tutorial directory, with appropriate changes. All 10 structures are accepted.

A superposition of the PDB file for 1g6j (blue) and the lowest energy accepted structure (red) is shown below. These structures agree well, except for the relatively unrestrained C-terminal region.

Justin T. Douglas
1/21/09

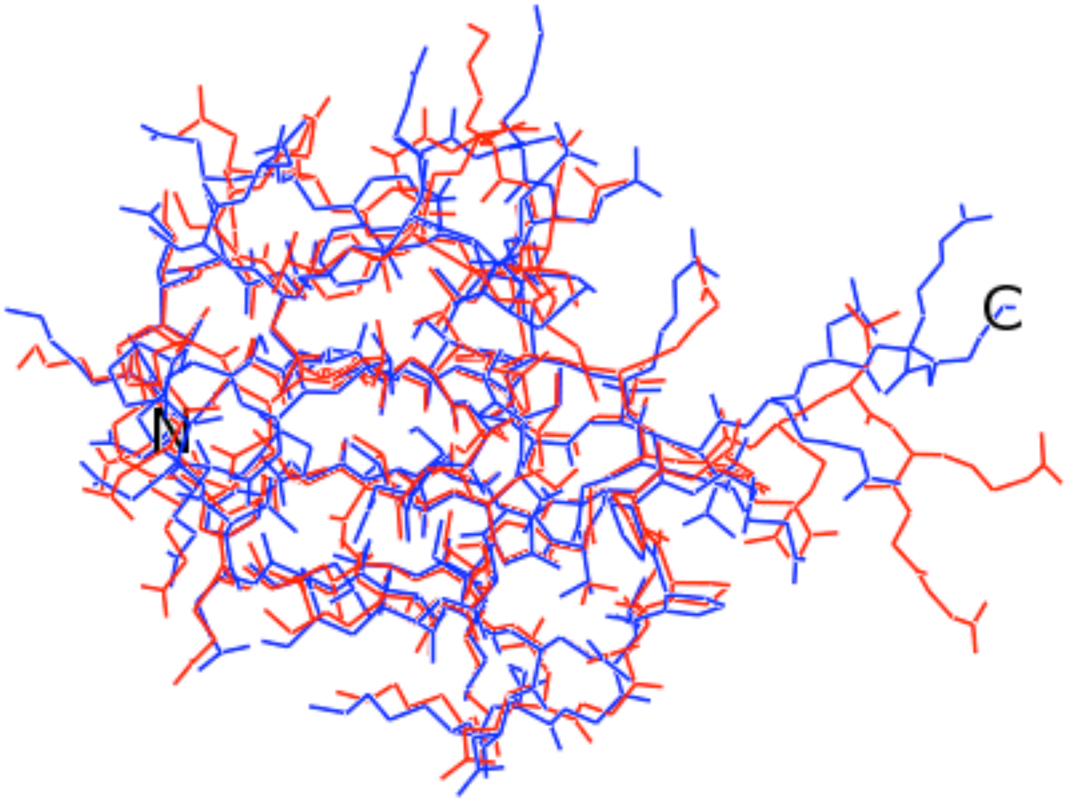
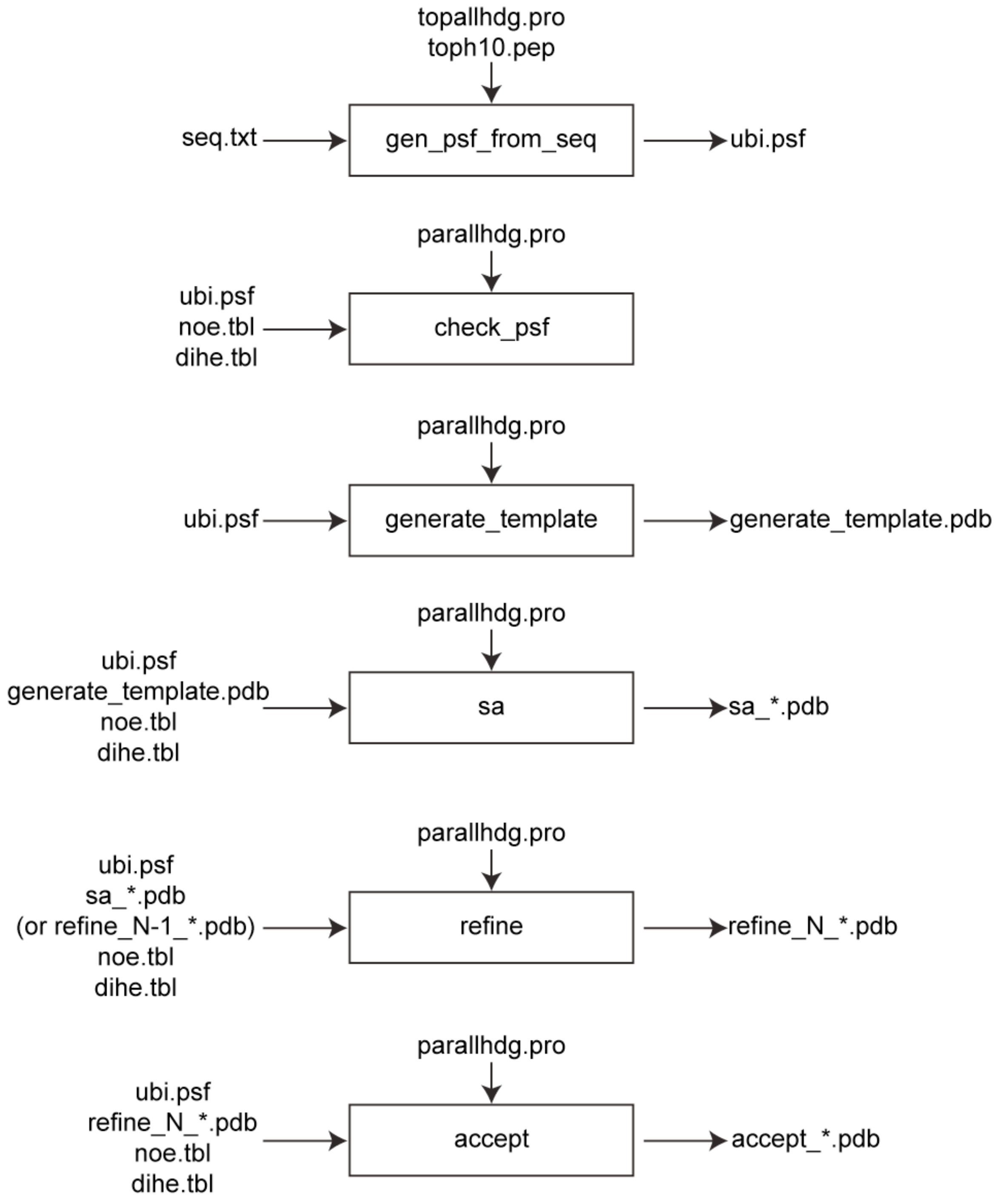


Figure 2. Superposition of lowest energy structure calculated from NOE and dihedral restraints downloaded from BMRB, accession code 4245, (red) and lowest energy structure of 1G6J.pdb.

Justin T. Douglas
1/21/09

Flow Chart for Xplor-NIH Scripts used for structure calculation



Justin T. Douglas
justindo@gmail.com