REDUX
A collection of Python scripts for torsion angle Monte Carlo protein molecular simulations and analysis. The program is based on unified residue peptide model and is designed for more efficient exploration of conformational space compared to all atom peptide models.

Creation of REDUX was partially motivated by the publication of an efficient Cα-only local move algorithm in Python by Wouter Boomsma and Thomas Hamelryck, *BMC Bioinformatics* 2005, **6**:159.

The REDUX moveset includes a pivot move (around a single residue φ, ψ) and for each successful pivot move a user defined number of local fragment moves. The local move needs 10 sequential residues to work with: first 3 are static anchor, middle 4 change, and last 3 move *en bloc* to defined rmsd with their pre-move positions. This was designed to allow some flexibility in the pivoting arm to avoid VDW clash as the long arm moves.

The model includes a Cα atom, a single pseudo atom side chain with radius proportional to side chain size, and a backbone psuedo hydrogen bond site (of zero radius) between each pair of Cα atoms (See **Model** below).

The pivot moveset is based on single residue φ, ψ distributions in a PDB-derived coil library converted to α, τ values. REDUX includes a mapping of binned α, τ angle ranges into labeled mesostates for internal use (See **Movesets** below). The ability to use a probability mesostate file as a moveset was also included in order to use probability mesostates derived from NMR data. (Here the mesostates are in α, τ space.)

The energy functions include:

1. **soft debump** (V = Eps,i,j[(Sigma,i,j/ri,j)**12], see `soft_debump_score` in *Functions.py*)
2. **residue specific pairwise contact** (PDB derived, see below and `contact_score` in Functions.py)
3. **confinement** (Rg target, see `rg_confinement_score` in *Functions.py*)
4. **hydrogen bond** (distance and angle criteria – These were developed by converting a list of native PDB structures to the Ca reduced model and obtaining statistics on the distances and angle between respective backbone pseudo h-bond sites. See *Functions.py* `hbond_score` for details.)
5. **backbone mesostate distributions** (agreement with PDB derived, see `meso_state_score` in *Functions.py*)

Current version is 2.1 available as a tarball, `REDUX_2_1.tar.gz.`
To install,

```
tar xzf REDUX_2_1.tar.gz
```

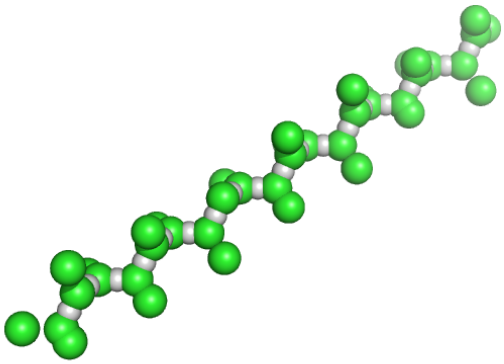
and then

```
cd redux/REDUX_2_1
python setup.py install
```

(using python version 2.x with appropriate write privileges)
REDUX requires Numeric and Biopython to be installed in the python distribution.

Model

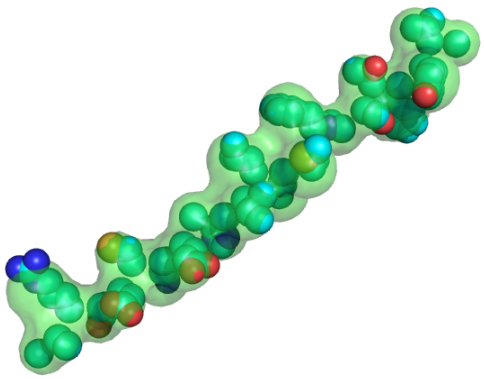
Below is an image of the model of all 20 residue types in alphabetical order, from ALA in the lower left, to VAL in the upper right. The white atoms are the hydrogen bonding sites which have a defined radius of zero in the program. (Here all atoms are default pymol size for carbon or hydrogen. The Cα – Csidechain bond length is variable depending on the defined size of the side chain pseudo atom [see below].)



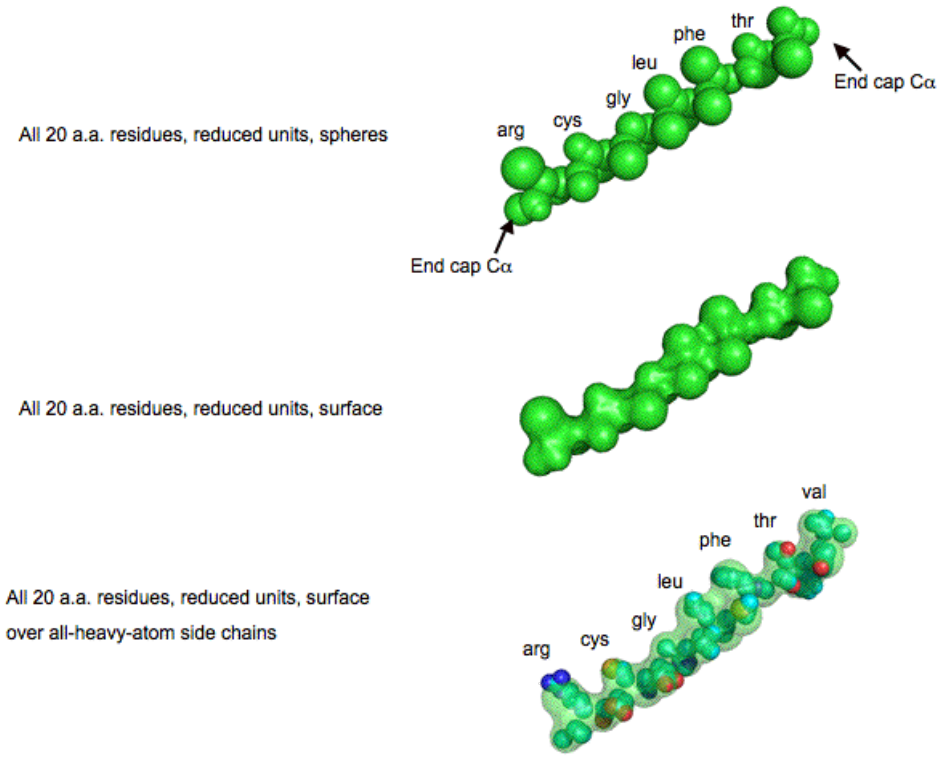
The defined side chain pseudo atom radii are (this is cut from *vdwrad.py* explained later):

```
cmd.alter('resn ala and name cb','vdw=2.0')
cmd.alter('resn arg and name cb','vdw=3.6')
cmd.alter('resn asn and name cb','vdw=2.7')
cmd.alter('resn asp and name cb','vdw=2.7')
cmd.alter('resn cys and name cb','vdw=2.6')
cmd.alter('resn gln and name cb','vdw=3.4')
cmd.alter('resn glu and name cb','vdw=3.4')
cmd.alter('resn his and name cb','vdw=3.2')
cmd.alter('resn ile and name cb','vdw=3.3')
cmd.alter('resn leu and name cb','vdw=3.3')
cmd.alter('resn lys and name cb','vdw=3.6')
cmd.alter('resn met and name cb','vdw=3.3')
cmd.alter('resn phe and name cb','vdw=3.5')
cmd.alter('resn pro and name cb','vdw=2.8')
cmd.alter('resn ser and name cb','vdw=2.4')
cmd.alter('resn thr and name cb','vdw=2.8')
cmd.alter('resn trp and name cb','vdw=4.2')
cmd.alter('resn tyr and name cb','vdw=3.7')
cmd.alter('resn val and name cb','vdw=3.0')
cmd.alter('name ca', 'vdw = 2.8')
cmd.alter('name hb', 'vdw = 0.0')
```

These radii were chosen by trial and error to make the pseudo atom representing the sidechain approximately cover the actual all atom side chain when it was in its most ususal rotamer, i.e. the pseudo atom would accupy approximately the same volume as the actual side chain. Here is an example of the REDUX model in surface rendering over the all heavy atom representation:

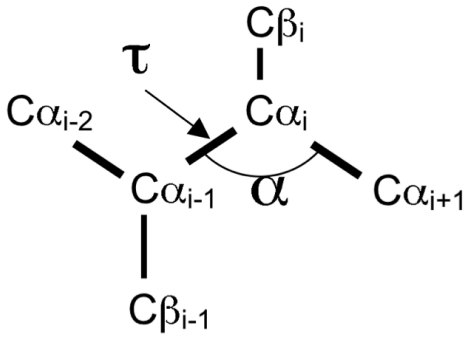


This is also illustrated in the following sequence of images:



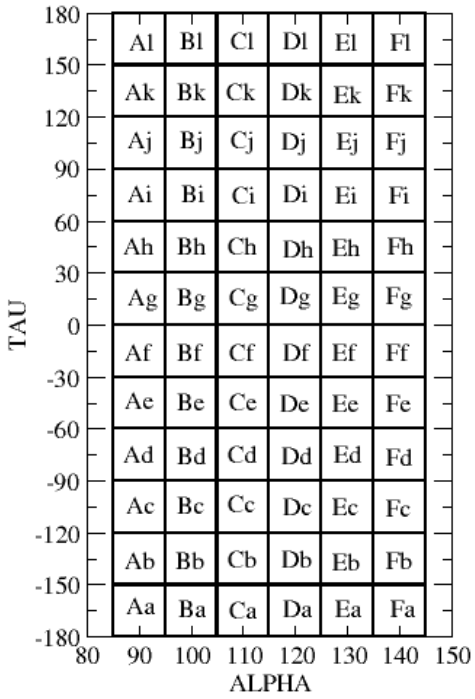
Movesets

The movesets are attempts to sample a single α, τ angle pair. The α, τ angles are defined below (note that each α, τ pair describes the relationship of 4 Cα atoms and the respective Cβ atoms of the central pair of Cα atoms):

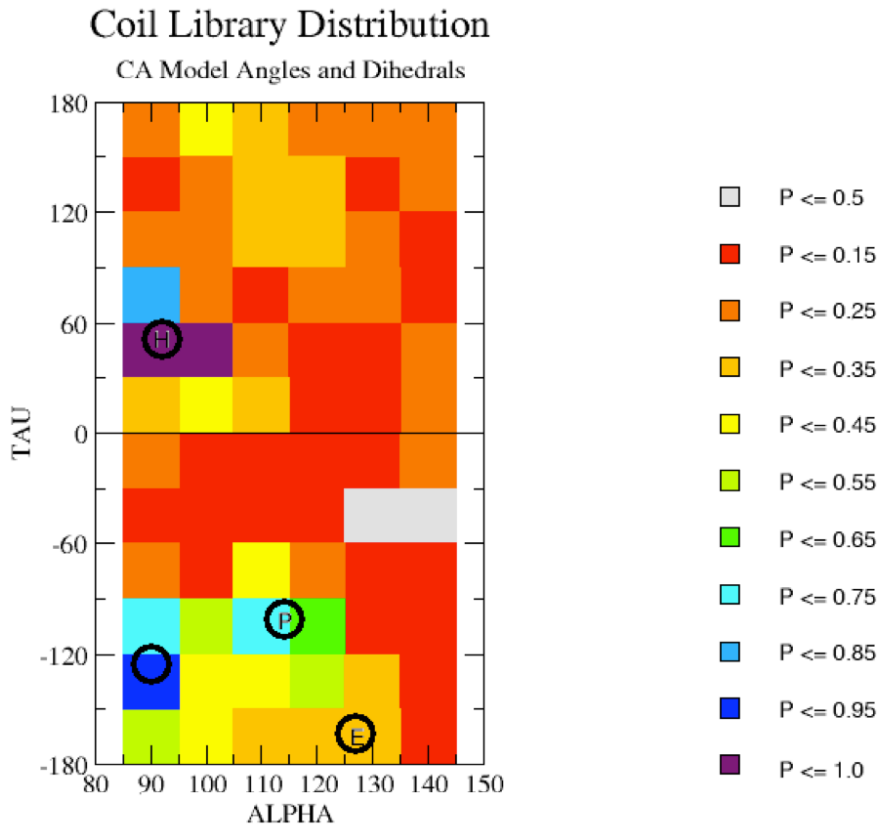


The α, τ space was split into bins and each bin is labeled as a specific mesostate for use within REDUX,

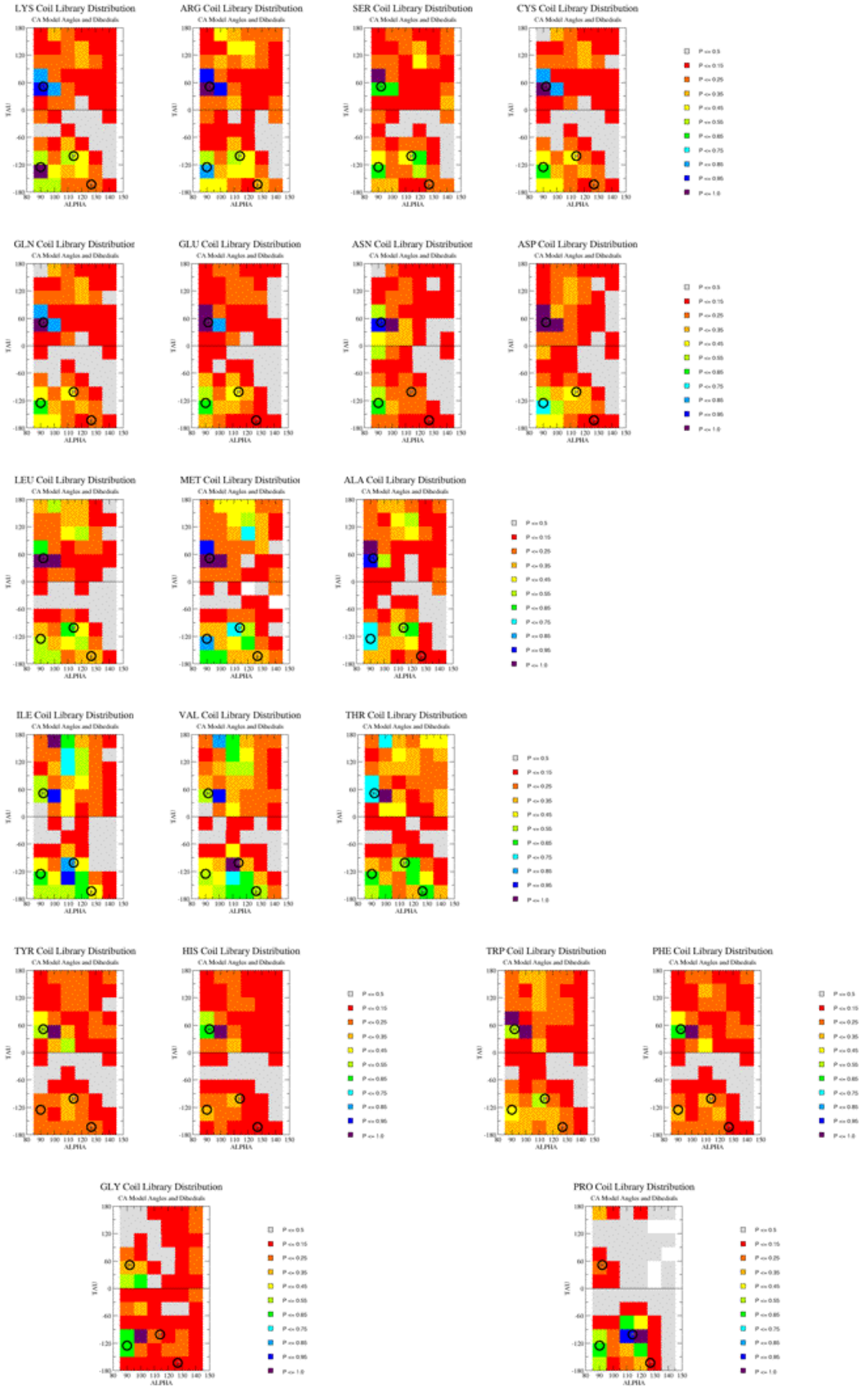
Alpha Tau Mesostates



The overall coil library distribution of α, τ angles, as fractional incidence, is illustrated below as a mesostate plot. Also indicated are the regions that correspond to helix (H), PII (P), strand (E), and turn type 1 (O).



Each attempted pivot move is chosen from the specific residue type α, τ probability mesostate distributions illustrated next:



Components of REDUX

AlfTau2Meso.py
Utility to convert list of α, τ angles to new mesotates.

Build_pept.py
Build model from sequence file

ContactEnergies.py
Dictionary of Residue contact energies from the PDB

Functions.py
Common functions used in REDUX

MesoDict.py
Dictionary of α, τ mesostate definitions

MesoTypes.py
Dictionary of mesostate indices

Mk_camatrix_filelist.py
Makes a data list of Cα distance vectors for input to Pycluster

Mk_resi_pmeso_wt.py
Makes a residue probability mesostate file from a native PDB file. Residue mesostate probabilities will be 1.0 for the mesostate of the native structure.
This pmeso file can be used to set the choice of mesostates during simulation to the same as the native PDB file.

Mk_resi_pmeso_wt_GLY.py
As above but if residue is GLY the probabilities are set to the PDB distribution of GLY alf-tau values.

PDB2seq.py
Makes a sequence file in format for input to Build_pept.py

PDBio.py, PDBio_PDB.py
Functions for reading PDB files. The first is to read REDUX files, the second is to read native PDB files. These are imported by the approriate scripts as needed.

Pmeso_Weights.py
Contains data for residue specific alf-tau mesotate probabilities.

Redux.py
Main command or run file for REDUX

Redux_energies.py, Redux_energies_PDB.py

Calculates REDUX energies for structures. First is for REDUX PDB files, second if for native PDB files.

Redux_from_PDB.py

Convert a native PDB structure to a REDUX reduced model structure with same Capha postions.

Redux_from_Seq_Ang.py

Build model from list of residues and phi, psi angles.

Redux_meso_run1.py, Redux_meso_run2.py

Command or run files to run a REDUX simulation using a probability mesostate file. These are the same file with different parameters and options.

Torsion.py

Calculates torsion angles α, τ for Cα model of protein.

Zmatrix.dat

Description of REDUX reduced model.

new_cluster.py

Cluster by structure using Pycluster.

vdwrad.py

List of atom radii to read into Pymol for display of REDUX models.

Running a simulation with REDUX

Note: For the following commands you must have set an environmental value to define the path where REDUX_2_1 resides. E.g.

```
setenv REDUX  '/usr/local/lib/python2.4/site-packages/redux_2_1'
```

You may start with the sequence of the polypeptide in a special format. This can be created with an editor or obtained from a standard PDB file with the following type of command,

```
python $REDUX/PDB2seq.py pdb1pgb.cln > b1.seq
```

(the suffix "cln" just indicates that this PDB file contains only ATOM records, only one chain, only one of any alternate conformations, etc.) and where the sequence file has the following format,

```
% head b1.seq
DEFAULT ALF  127.0
DEFAULT TAU  -165.0
RES  END
RES  MET
RES  THR
RES  TYR
RES  LYS
RES  LEU
RES  ILE
RES  LEU
.
.
.
% tail b1.seq
RES  ALA
RES  THR
RES  LYS
RES  THR
RES  PHE
RES  THR
RES  VAL
RES  THR
RES  GLU
RES  END
```

The first two lines define the default α, τ angles for the model; these can be edited to your choice. The capping residues called "END" are necessary.

Then create the REDUX model,

```
python $REDUX/Build_pept.py b1.seq > b1_redux.pdb
```

where *b1_redux.pdb* is now of the form,

```
% head b1_redux.pdb
COMPND
ATOM      0  CA  END     0       1.000   1.000   1.000  1.00  0.00
ATOM      1  CA  MET     1       4.800   1.000   1.000  1.00  0.00
ATOM      2  CB  MET     1       6.139  -1.661   1.355  1.00  0.00
ATOM      3  HB  MET     1       5.943   2.517   1.000  1.00  0.00
ATOM      4  CA  THR     2       7.087   4.035   1.000  1.00  0.00
ATOM      5  CB  THR     2       6.471   5.281   1.161  1.00  0.00
ATOM      6  HB  THR     2       8.946   4.066   0.607  1.00  0.00
ATOM      7  CA  TYR     3      10.804   4.097   0.215  1.00  0.00
ATOM      8  CB  TYR     3      12.198   1.118  -0.982  1.00  0.00
.
.
.
% tail b1_redux.pdb
ATOM    157  CB  VAL    54     163.885  84.377 -10.226  1.00  0.00
ATOM    158  HB  VAL    54     166.678  82.665  -9.751  1.00  0.00
ATOM    159  CA  THR    55     168.576  82.659  -9.818  1.00  0.00
ATOM    160  CB  THR    55     169.199  81.406  -9.761  1.00  0.00
ATOM    161  HB  THR    55     169.728  84.169  -9.752  1.00  0.00
ATOM    162  CA  GLU    56     170.879  85.678  -9.686  1.00  0.00
ATOM    163  CB  GLU    56     169.544  88.412  -9.089  1.00  0.00
ATOM    164  HB  GLU    56     172.724  85.731 -10.140  1.00  0.00
ATOM    165  CA  END    57     174.568  85.783 -10.595  1.00  0.00
END
```

Note that both the N-terminus and C-terminus are capped with a residue called “END” which contains only a Cα atom.

To visualize the reduced residue structure in PyMOL get a copy of the atom radii,

`cp $REDUX/vdwrad.py .`

Then, after loading *b1_redux.pdb* into PyMOL run the radii script,
PyMOL> **`@vdwrad.py`** (and then show spheres)

To run a torsion angle MC simulation get a copy of the command file,

`cp $REDUX/Redux.py .`

and edit the following section appropriately,

```
####### EDIT THESE IF DESIRED #########

    # How large should the confinement cage be (2.8 is normal, 2.5 is small, 3.5 is big)
    # The cage will gradually shrink to the end size
    start_rg_prefac = 3.0
    end_rg_prefac = 3.0
    Rg_predict =  start_rg_prefac * (math.pow(numres, 0.34))


    # Scale the various scoring functions
    # Set to 0.0 to turn off that function
    #
    # Confinement
    rgscale = 0.1     # 0.1 default
    #
    # Soft Debump
    dbscale = 1.0     # 1.0 default
    #
    # Residue Specific Contact
    cnscale = 1.0     # 1.0 default
    #
    # Hydrogen bond
    hbscale = 1.0     # 1.0 default
    #
    # Backbone Mesostate Propensity
    msscale = 1.0     # 1.0 default
```

```
    # How many cycles: where a cycle is numres attempts to get a successful pivot move?
    ncycles = 100

    # Save a frame every how many cycles of pivot moves:
    save_int = 1

    # How many local move attempts for each successful pivot move?
    l_trials = 1

######## END EDIT ############
```

Then run the simulation with a command similar to the following,

```
python Redux.py b1_redux.pdb
```

The output should be in a directory called `sim/`

```
% ls sim/
b1_redux.pdb           min.pdb          traj.pdb
```

where *`b1_redux.pdb`* is the initial extended model, *`min.pdb`* is the minimum energy conformation sampled during the simulation and *`traj.pdb`* contains multiple conformations saved during the simulation. The `traj.pdb` is in the format of NMR multiple PDB files and can be loaded directly into PyMOL for viewing as a movie. This allows one to see the difference between a pivot move and a local move.

To collect minimal energy conformers from multiple independent simulations the following lines can be inserted into *Redux.p*y instead of the simple `run_sim(file)` command,

```
    for i in range(400):
        run_sim(file)
        os.rename('min.pdb', 'emin_' + `i` + '.pdb')
```

Utility scripts

To calculate the α, τ angles and associated mesostates from a PDB file:

```
python $REDUX/Torsion.py pdb1pgb.cln > pdb1pgb.ang
```

where *pdb1pgb.ang* looks like the following,

```
% head pdb1pgb.ang
   3 TYR LYS  126.28 -174.63  Ea
   4 LYS LEU  132.83 -146.26  Eb
   5 LEU ILE  122.41 -156.52  Da
   6 ILE LEU  109.70 -157.18  Ca
   7 LEU ASN  116.94 -155.74  Da
   8 ASN GLY  101.29  168.70  Bl
   9 GLY LYS  127.78  148.11  Ek
  10 LYS THR   90.01 -105.39  Ac
  11 THR LEU  108.97   13.87  Cg
  12 LEU LYS  119.07   55.23  Dh
```

To calculate the REDUX energies of a conformation:

From a REDUX model structure,

```
% cp $REDUX/Redux_energies.py .
% python Redux_energies.py min.pdb
debump    0.251166874474
confine   0.764243831542
contact   -3.97808618667
meso      -2.3057013828
hbond     -0.0
```

From a native PDB file,

```
% cp $REDUX/ Redux_energies_PDB.py .
% python Redux_energies_PDB.py pdb1pgb.cln
debump    2451572.11148
confine   0.0
contact   -7.46524261987
meso      -1.57870798552
hbond     -11.0
```