Here we provide a stepwise description of *HYCUD* calculations for the monomeric L12 system as our tutorial example. Monomeric L12 is a single-chain protein containing 120 residues. The protein comprises of two rigid domains, the N-terminal domain (NTD: residues 3-30) and C-terminal domain (CTD: residues 53-120), connected by a flexible linker (residues 31-52).

System requirements:

Python 3.0 and Perl should be installed on your system. Create an alias for Python (*python3*), if necessary, so that you can call it from any directory.

A. Generation of structural ensemble:

HYCUD calculation requires a structural ensemble of the protein of interest. Here, for the purpose of this tutorial, 500 structural models of monomeric L12 were generated using EOM software. The NTD (3-30) and CTD (53-120) were treated as rigid bodies, while residues 1-2 and 31-52 were allowed to sample conformational space according to their amino acid conformational propensities.

Required input files:

L12mono.seq

NTDmono.pdb

CTDmono.pdb

For academic users, the EOM software is available here:

http://www.embl-hamburg.de/biosaxs/download.html

The log file (Rancheom.log) contains information on how the EOM program was run. The generated models are stored in the “pdbs” directory under step_A_eom.

B. Generation of a single PDT file

A Python script (*makePDT.py*) was used to merge the 500 PDB files generated in step A into a single PDT file. In the PDT file, each model starts with a REMARK line specifying the source PDB file.

Required input files:

All PDB structures of monomeric L12 generated in step A.

Command:

```
python3  makePDT.py  --inDir  ../step_A_eom/pdbs  --outFile L12monomer_500.pdt
```

To ensure that residue numbering at each protein model within the PDT file starts at 1 and increments consecutively, we used another Python script *fixResidueNumbering.py*.

Command:

```
python3  fixResidueNumbering.py L12monomer_500.pdt L12monomer_500_fixed.pdt
```

Consult the help option for further information:

```
python3  makePDT.py  --help
```

```
python3  fixResidueNumbering.py --help
```

C. Hydrodynamic calculation for isolated domains

Using HYDROPRO10, the rotational correlation time (τ_c) and intrinsic viscosity ($[\eta_0]$) of isolated NTD and CTD domains were calculated. The executive file, *hydropro10-lnx.exe*, takes the *hydropro.dat* as its parameter file. In *hydropro.dat*, parameters temperature, corresponding viscosity and molecular weight of the related protein domain should be specified. Special care should be taken to parameters AER, SIGMIN and SIGMAX. The AER value of 2.9 Angstrom is recommended for rigid domains. Parameters SIGMIN and SIGMAX are adjusted in order to ensure that the number of mini-beads used to build a shell model of the interested protein domain ranges between ~ 300 and 2000.

Required input files:

NTDmono.pdb

hydropro.dat (with parameters of NTDmono)

CTDmono.pdb

hydropro.dat (with parameters of CTDmono)

For academic users, the HYDROPRO10 software is available here:

http://leonardo.inf.um.es/macromol/programs/hydropro/hydropro.htm

The obtained rotational correlation times ($\tau_{c,0}$) and intrinsic viscosities ($[\eta_0]$) of isolated NTD (mono) and CTD domains will be used for the next step of HYCUD calculations.

D. HYCUD calculations

At the core of HYCUD calculations lies a Python script, called *hycud.py*. First, open the script and modify the default values accordingly. The usage of *hycud.py* script requires the followings:

D.1. The PDT file generated in step B; The PDT file name is entered as an argument for “--pdt" option. Make sure that residue numbering within the PDT file is already fixed (as described in section B).

D.2. The EOM-generated ensemble contains only CA traces for disordered parts. It is therefore essential that the backbone and side chain coordinates are reconstructed before hydrodynamic calculations are performed. To this end, a third-party program called REMO is called within HYCUD. This is done via “-R" option, which takes as its argument the path to REMO directory. The REMO program can be freely downloaded from here:

http://zhanglab.ccmb.med.umich.edu/REMO/

The REMO-reconstructed PDT file can be exported by “--REMOout” option for future use. If the input PDT file already contains full coordinates, the REMO step will be skipped by “--NoREMO" flag.

D.3. The user should define a fragmentation scheme which determines the boundaries between rigid and disordered parts and inside the disordered parts as well. This is done through “-D” or “--detailedFrag" option. The boundary between rigid and disordered regions is decided on the basis of existing experimental evidence, e.g. NMR spin relaxation rates. Inside the disorder regions, the fragments are defined with ~12-14 residues each. This is nearly twice the persistence length of unfolded polypeptide chains, and represents the so-called statistical Kuhn’s length. For monomeric L12, the fragmentation scheme was defined as residues (1-2),(3-30),(31-41),(42-52) and (53-120).

D.4. Since globular domains, as defined in step A, keep their same size/shape across the protein ensemble, one can fix their initial rotational correlation time $\tau_{c,0}$ and intrinsic viscosity $[\eta_0]$ to the values known from experiment or obtained through hydropro calculations at step C. In this way, during HYCUD calculations, the hydropro calculations will be skipped for these domains. In case of L12 monomer, we fixed $\tau_{c,0}$ and $[\eta_0]$ of NTD domain at 2.161 ns and 5.025 cm^3/g and of CTD domain at 3.526 ns and 3.817 cm^3/g, as obtained from hydropro calculations at step C. For the short fragment consisted of residues 1-2, we fixed its $[\eta_0]$ at 0 and thereby switched off its effect on the τ_c of other fragments.

D.5. The executory file of HYDROPRO10 program and the parameter file *hydropro.dat* are required. The path to their directories is declared by options “--exe" and “--in", respectively, or it is taken as default values from *hycud.py* file. In the parameter file, one should set temperature and corresponding viscosity at which the rotational correlation time is concerned. Molecular weight for each fragment will be calculated within the *hycud* script, so it is not required to be specified. Parameters SIGMAX and SIGMIN in the *hydropro.dat* parameter file should be adjusted in order to ensure that the number of mini-beads used to build a shell model of the typical fragments ranges between ~ 300 and 2000. We’ve set SIGMAX and SIGMIN at 1.5 and 0.5, respectively.

D.6. It is necessary to specify a directory for temporary files (via the “--tmpDir" option).

D.7. The results of *HYCUD* calculations are exported through the “--outData” option into a .res file.

The *HYCUD* was run with the following options (do not forget to first open the *hycud.py* and modify default values accordingly):

```
python3 hycud.py --tmpDir ./tmp1 --exe ./hydropro10-lnx.exe --in ./hydropro.dat --pdt ./step_B_mergingPDBs/L12monomer_500_fixed.pdt –D “(1-2 v0.0 t1.0e-9)(3-30 v5.025 t2.161e-9)(31-41)(42-52)(53-120 v3.817 t3.526e-9)” --outData L12monomer_500.res --REMOout L12monomer_500_afterREMO.pdt -v
```

The level of verbosity can be increased by using -vv, -vvv or –vvvv.

As an additional application, *HYCUD* enables ensemble-based calculation of translational diffusion coefficient (therefore, hydrodynamic radius through Stokes-Einstein equation). For this, use the following command:

```
python3 hycud.py --tmpDir ./tmp1 --exe ./hydropro10-lnx.exe --in ./hydropro.dat –pdt L12monomer_500_afterREMO.pdt --translationOnly –D “(1-120)” --outData L12monomer_500_translation.res --NoREMO -v
```

Since the HYDROPRO calculation time increases very quickly by the number of mini-beads in the shell model, special care should be taken in setting SIGMAX and SIGMIN in *hydropro.dat* file. Here, we’ve used SIGMAX and SIGMIN of 2.5 and 1.5, respectively, to keep calculation time at a reasonable level.

For further information about different options and their arguments, consult

```
python3  hycud.py --help
```

E. Analysis of *HYCUD* results

The results of *HYCUD* calculations can be viewed by the following command:

```
python3  hycud.py --inData L12monomer_500.res -v
```

The detailed information on *HYCUD* calculations can be obtained through “--inInfo” flag:

```
python3  hycud.py --inData L12monomer_500.res --inInfo
```

To remove outliers, the fragment-specific τ_c distribution can be trimmed from the rightmost side (largest rotational correlation time) until the relative change in the standard deviation of distribution upon removal of the last point falls below a certain threshold value, e.g. 5%. In other words, models are sorted on the basis of their fragment-specific τ_c. The standard deviation of τ_c is calculated with and without the inclusion of the model with longest τ_c. If relative change in standard deviation (i.e. $(SD_{with}-SD_{without})/SD_{with}$) exceeds the specified cut-off value, the related model will be excluded. The same procedure is then iterated until the relative change in standard deviation falls below the cut-off value. The command is:

```
python3  hycud.py --inData L12monomer_500.res --filterOutliers 0.05 -v
```

We recommend starting from a cut-off value of 0.9. This will exclude models only if their removal decreases the standard deviation of at least one fragment by one order of magnitude. Then, try a cut-off value of 0.5 and decrease it gradually until the average τ_c gets nearly stable.

To display the harmonic average of τ_c instead of arithmetic average, use the following option:

```
python3  hycud.py --inData L12monomer_500.res --displayHarmonicMean  -v
```

To include only N (e.g. 100) models in statistical calculations, starting from the first model, one can use the following command:

```
python3  hycud.py --inData L12monomer_500.res --inCount 100
```

To include only 100 models in statistical calculations, but starting from model 11:

```
python3  hycud.py --inData L12monomer_500.res --inCount 100:10
```

One can also use a combination of --filterOutliers and --inCount options:

```
python3  hycud.py --inData L12monomer_500.res --inCount 100 --filterOutliers 0.01 -v
```

For a more detailed analysis, the *HYCUD* results can be exported as a text file to be opened by spreadsheet applications (like MS-Excel):

```
python3  hycud.py --inData L12monomer_500.res  --dumpDataTable L12monomer_500.txt
```

To evaluate the precision of *HYCUD* predictions, the averaging of τ_c could be performed over 10 non-overlapping sub-ensembles, and the standard deviation among predicted τ_c is considered as the uncertainty of *HYCUD*-predicted τ_c.