

Adaptive Anisotropic Network Model Program Tutorial

Step 1. Prepare the PDB files.

Two PDB files of the ending structures are required. The structures need to be aligned and share common sequence. Kabsch algorithm is suitable for the structural alignment. To implement it on PyMOL, please refer to: <http://www.pymolwiki.org/index.php/Kabsch>

Rename the aligned PDB files to “xxxx.pdb” and “yyyy.pdb” (xxxx and yyyy could be any alphanumeric 4-letters). *Tip: use PyMOL to output the aligned structures.*

Important: Make a copy of the ending structure PDB files into the folder “PathGenerate”, since after each iteration, the “xxxx.pdb” and “yyyy.pdb” will be replaced by a new pair of intermediate structures. In the folder of “PathGenerate”, the original ending structure files will be used later to generate the pathway trajectories.

Step 2. Set up the setting files.

The setting files have “*.inp” file extension.

Setting File Name	Description *	Called by codes
Energy_Function.inp	Parameters of the double minima quadratic energy function. For details see Eq. (7) (8) and Ref. 49.	path.x
Intermediate_States_Generator.inp	Parameters for the α ANM iteration. This setting file will update itself automatically after running each successful iteration. Just follow the instructions to setup for the first iteration (e.g. set the iteration number to 1). For details see Eq. (4) & (5).	inter.x
MEIG_seq.inp	The approximated number of eigen-modes needed to be solved in each iteration. Just make an approximation. If the [MEIG] too small, the program can increase it automatically. Each line specify one iteration.	inter.x
PDB_Nodes_Atoms.inp	The nodes atoms name from PDB file. By default, only the CA atom is needed (e.g. the first line.)	anm.x
PDBList.inp	List the PDB id(s). Must be alphanumeric 4-letters.	anm.x
runANM.inp	Parameters for the ANM program. For details see Eq. (8) and Ref. 55.	
Transition_Pathway.inp	Parameters for combining the pathways. User can setup it after running the “inter.x” and knowing the number of total iterations.	path.x

*All the reference and equation numbers are referred to the paper in *PLoS Comput Biol* (2009) 5(4): e1000360.

Tip: Once the setting files are set, no need to update them after each iteration.

Step 3. Generate the intermediate structures.

Requirement: Matlab installed on the Linux server.

Put all the PDB files (ending structures), setting files (*.inp files), executable file (*.x files) and the Matlab source codes (*.m files) in one directory.

Write a shell scripts to repeat the following task:

```
./anm.x  
matlab -nodisplay <anmEigenMulti.m  
./inter.x
```

Each successful running represents one iteration, and generates a new pair of intermediate states. As an example see file “inter.cmd”, which was designed for the serial jobs (see “gserial” on Linux) with 8 possible iteration. User can write a loop for higher iterations.

A log file will be generated with extension “*.log”. keep tracing the log file, user can get the information of current iteration number, number of eigen-modes used, and RMSD between pair of intermediate states.

Step 4. Generate the pathway trajectories.

- Copy all the newly generated “*.xyz” into the folder “PathGenerate”.
- Identify the total iteration number from previous step and setup the “Transition_Pathway.inp” file.
- Then submit the job (once).

```
./path.x
```

The pathway trajectories can be illustrated with PyMOL by open the “*.pml” file. Each intermediate structures can be found in the pathway PDB files.

The source codes and Makefiles can be found in the “sourceCode” folder.

For questions, bugs or suggestion: Please feel free to contact the codes author, Zheng Yang, Email: proteinyang@gmail.com

For citation: Zheng Yang, Peter Májek, and Ivet Bahar (2009) Allosteric Transitions of Supramolecular Systems Explored by Network Models: Application to Chaperonin GroEL. *PLoS Comput Biol* **5(4)**: e1000360.

Thank you for using aANM!