==========

The TEN Package

Tools for Elastic Networks

User Guide

Lars Ackermann, Silke A. Wieninger and G. Matthias Ullmann

==========

Contents

The TEN package implements a coarse-grained model for the analysis of biopolymer dynamics, the Elastic Network Model (ENM). It consists of three programs:

- EValVecC: calculates eigenvalues and eigenvectors (normal modes) based on protein structures
- NWAnalyzer: calculates covariances, correlations and theoretical B-factors (flexibility values)
- MovieGenerator: computes trajectories in dcd-format based on protein structure, eigenvalues and eigenvectors

Two versions of ENM, the Gaussian Network Model (GNM) and the Anisotropic Network Model (ANM) are implemented. The theory of Elastic Network Models is described elsewhere.[1–3]

1 Installation

TEN is available under the Gnu Affero Public License. It requires the libraries Boost, Lapack and Lapack++. For installation, untar the tar file, set the paths in Makefile.rules to your settings and type make.

To execute TEN, you need to set the environment variable LD_LIBRARY_PATH:
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:"/path_to_your_lapackpp.so_library"

2 The programs

2.1 EValVecC - calculate eigenvalues and eigenvectors

The program requires two input files:

- *prot*.pqrm
- settings

Computation GNM

- Build up the $N \times N$ Kirchhoff matrix. For all covalently bound atoms (those which have a connect entry in the pqrm-file) the matrix entry is set to the negative covalent force constant $-k_{\mathrm{cov}G}$
- Calculate the distance between all atoms which are not covalently bound

- Determine the noncovalent force constant $k_{\mathrm{ncov}G}$ according to the method chosen by *interacMode*
- The eigenvalues and -vectors are calculated using the LAPACK library.

Computation ANM

- Build up the Kirchhoff matrix as in GNM, but using $k_{\mathrm{cov}A}$ and $k_{\mathrm{ncov}A}$ instead of $k_{\mathrm{cov}G}$ and $k_{\mathrm{ncov}G}$
- Construct the $3N \times 3N$ Hessian matrix
- The eigenvalues and -vectors are calculated using the LAPACK library.

The output is written to the files *prot*_eigValsAndVecsGNM.out and *prot*_eigValsAndVecsANM.out.

2.2 NWAnalyzer: calculate covariances and theoretical B-factors

The program requires three input files:

- *prot*_eigValsAndVecsGNM.out or *prot*_eigValsAndVecsANM.out
- *prot*.pqrm
- settings

The pqrm file is needed for the option $bvalAlphaOnly$ to identify the C_α-atoms.

Computation

- In case of mass-weighted ANM, the Hessian matrix must be reweighted by the masses
- Calculate the pseudo inverse of the Kirchhoff or Hessian matrix using the eigenvalues and -vectors
- Calculate the covariances, correlations and theoretical B-factors

The following output files are written:
*prot*_covsGNM.out, *prot*_corrsGNM.out and *prot*_bValsGNM.out
*prot*_covsANM.out, *prot*_corrsANM.out and *prot*_bValsANM.out

2.3 MovieGenerator: generate trajectories

The program reads the following input files:

- *prot*_eigValsAndVecsGNM.out or *prot*_eigValsAndVecsANM.out
- *prot*.pqrm
- settings
- optional: *ReferenceProteinName*.pqrm

Computation

- Calculate amplitudes of the modes as $\alpha_k = \sqrt{2\mathrm{RT}/\lambda_k}$ with λ_k being the eigenvalue of mode k, R the universal gas constant and T the temperature in Kelvin
- The displacement vector of a single mode k is calculated as $D_k = M^{-1/2}\alpha_k L'_k \cos(t)$ with mass matrix M and eigenvectors L' if mass-weighting was used in EValVecC
- The displacement vector of the supertrajectory is calculated as $D_{\mathrm{super}} = \sum_k M^{-1/2}\alpha_k L'_k \cos(\omega_k t + \Phi_k)$
- Calculate new coordinates by adding the displacement vector to the previous coordinates
- If "*compareConfs* 1" is chosen, the snapshot conformation is compared to the reference structure by calculating the difference distance matrix and adding up the matrix elements.

The single mode trajectory files are called *prot*_traj*.dcd or *prot*_traj*.pdb. The supertrajectory file is called *prot*_superTraj.dcd or *prot*_superTraj.pdb. For loading pqrm and dcd files in VMD, first load the pqrm file using *Determine File Type:* PDB, then load the dcd files into it.

3 Some setup options

3.1 Non-covalent force constants

The non-covalent force constants can be set to a constant value for all node pairs with distance smaller than the cutoff radius (*interacMode*=0), or distance-dependent force constants can be calculated by:[4]

$$k_{\mathrm{ncov}}(\mathbf{r}) = \frac{1}{2} k_{\mathrm{cov}} \exp\left(-\frac{|\mathbf{r}|^2}{r_\circ^2}\right) \qquad \text{for } interacMode = 1 \tag{1}$$

3.2 Two conformations

In MovieGenerator, the snapshot conformations can be compared to a reference conformation, provided by a second pqrm-file. To compare different protein conformations, we calculate the distance matrix D, i.e. the matrix of distances between all atom pairs, for each conformation. The difference distance matrix is given by

$$\Delta D = D_1 - D_2$$

A scalar value for comparing the conformations is obtained by summing over the absolute values of all entries of ΔD.

4 File formats

4.1 Input files

4.1.1 Structure file (PQRM)

The pqrm (PDB charge radius mass) file consists of two parts. The coordinate section has the following format:
[field name] ... specifies the entry type (ATOM or HETATM)
[atom number] ... atom index
[atom name] ... short name of atom
[residue name] ... amino acid name in 3-letter code
[chain id] ... optional, not used in the programs; chain index
[x y z] ... 3 float values specifying the atom coordinates
[charge] ... optional, not used in the programs; atomic charge
[radius] ... atomic radius in Å, not used in the programs
[mass] ... atomic molecular mass in u After the string TER follows the connect entry section:
[field name] ... specifies the entry type (CONECT)
[reference atom] ... index of the reference atom
[c1 c2 c3 ...] ... indices of atoms which are covalently bound to the reference atom

4.1.2 Settings file

A single settings file can be used for all programs. Settings which do not apply to the program in use are ignored.

General settings:

- *names* ... comma-separated protein names. MUST NOT BE EMPTY. pqrm-files with corresponding names must exist

- *protPaths* ... comma-separated paths of the pqrm-files; must have the same order as *names*. If empty, the pqrm files must lie in the same directory as the settings file.

EValVecC settings:

- *kcovA* ... force constant for covalent bonds in ANM; default value: 100
- *kncovA* ... force constant for non-covalent bonds in ANM; default value: 1
- *kcovG* ... force constant for covalent bonds in GNM; default value: 100
- *kncovG* ... force constant for non-covalent bonds in GNM; default value: 1
- *rcutA* ... cutoff distance in ANM; default value: 12Å
- *rcutG* ... cutoff distance in ANM; default value: 6Å
- *interacMode* ... determination of interaction constants
  0: directly use the given force constants
  1: weighted: weight non-covalent bonds according to the atom distance
- *massWeightedHessian* ... defines whether the Hessian matrix is weighted by the masses of the atoms
  0: not weighted
  1: weighted (default)
- *models* ... ENM type used in calculation, MUST BE ASSIGNED
  GNM: only Gaussian Network Model
  ANM: only Anisotropic Network Model
  GNM,ANM: both Network Models

NWAnalyzer

- *massWeightedHessian* ... defines whether the Hessian matrix was weighted by the atom masses in EValVecC
  0: not weighted
  1: weighted (default)
- *temp* ... defines the temperature; default: 298K
- *bvalAlphaOnly* ... defines whether B-factors of all atoms are printed
  0: print all B-factors
  1: print B-factors of C_α atoms only (default)

- *models* ... ENM type used in calculation, MUST BE ASSIGNED
  GNM: only Gaussian Network Model
  ANM: only Anisotropic Network Model
  GNM,ANM: both Network Models

MovieGenerator

- *massWeightedHessian* ... defines whether the Hessian matrix was weighted by the atom masses in EValVecC
  0: not weighted
  1: weighted (default)

- *temp* ... defines the temperature; default: 298K

- *traj_modes* ... defines for which modes (eigenvectors) trajectories should be computed. Format: "1-3,5,8-20". The eigenvectors are sorted by their eigenvalues in ascending order. The first modes provide the major movements of the protein. KEEP IN MIND: the first six modes are ignored because their eigenvalues are zero. If you specify *traj_modes* 1, eigenvector seven is used.

- *traj_tstep* ... defines the time step between single mode snapshots; default: $1/30$

- *print_traj* ... defines the file format of the single mode trajectory files

  - 0: do not print any single mode trajectory
  - 1: create dcd file; default
  - 2: create pdb file

- *supertraj_modes* ... defines for which modes the supertrajectory should be computed. The supertrajectory is the combination of the single modes with frequencies according to their eigenvalues. Default: all modes are taken.

- *phases* ... time offsets (phases) for the different modes in the calculation of the supertrajectory. It is possible to set only a couple of phases. The rest gets random values between 0 and 2π. There are three possibilites to set phases:

  - "0.1,0.3,...": The first phase value is set to 0.1, the second to 0.3 etc.
  - "0.1[4]": The phase of mode 4 is set to 0.1
  - "0.1[2-8]": The phases of modes 2 to 8 are set to 0.1

  If you specify *phases* 0.1[1-9],0.3, the phase of the tenth mode is set to 0.3

- *supertraj_tstep* ... defines the time step between snapshots of the supertrajectory, calculated as $(traj\_tstep)/\sqrt{\lambda_{\min}}$, where $\lambda_{\min}$ gives the smallest non-zero eigenvalue

- *supertraj_steps* ... defines the total number of supertrajectory steps; default: 50
- *print_super* ... defines the file format of the supertrajectory file
  - 0: do not print supertrajectory
  - 1: create dcd file; default
  - 2: create pdb file
- *compareConfs* ... compare the molecule snapshots to a reference conformation for each time step if *compareConfs* 1. Default is 0 (no comparison). Only C_α atoms are used for the comparison. The two structures must have the same number of C_α atoms.
- *refNames* ... comma-separated list of names of reference protein structures. Can only be emtpy if *compareConfs* 0. For each reference protein, the program computes the distances between reference and trajectory conformation for each time step.
- *refPaths* ... comma-separated list of paths of the reference pqrm files
- *writeConfDistances* ... defines the number of smallest distances given for the comparison between snapshots and the reference conformation; default: 10

4.2 Output files

4.2.1 Eigenvalues and eigenvectors

*prot*_eigValsAndVecsGNM.out and *prot*_eigValsAndVecsANM.out:
The eigenvectors are written to the output file in ascending order. The output files should not be manipulated manually because they are read by the other programs. The file is formatted in three paragraphs:

- settings which were used to calculate the eigenvalues and -vectors
- atom masses
- eigenvalue (first line) and associated eigenvector (second line)

4.2.2 Covariance and correlation

*prot*_covs*NM.out and *prot*_corrs*NM.out:
The first two columns give the atom indices, the third column the covariance or correlation

4.2.3 Theoretical B-factors

*prot*_bVals*NM.out:
B-factors of all atoms or C_α atoms only; atom index in the first column and B-factor in the second column

References

1. I Bahar, A R Atilgan, and B Erman. Direct evaluation of thermal fluctuations in proteins using a single-parameter harmonic potential. *Fold Des*, 2:173–181, 1997.

2. A R Atilgan, S R Durell, R L Jernigan, M C Demirel, O Keskin, and I Bahar. Anisotropy of fluctuation dynamics of proteins with an elastic network model. *Biophys. J.*, 80:505–515, 2001.

3. S A Wieninger, E H Serpersu, and G M Ullmann. ATP binding enables broad antibiotic selectivity of aminoglycoside phosphotransferase(3')-IIIa: an elastic network analysis. *J. Mol. Biol.*, 409:450–465, 2011.

4. K Hinsen. Analysis of domain motions by approximate normal mode calculations. *Proteins*, 33:417–429, 1998.