



HAL
open science

User Guide for NOLB: Non-Linear Normal Mode Analysis

Sergei Grudinin

► **To cite this version:**

| Sergei Grudinin. User Guide for NOLB: Non-Linear Normal Mode Analysis. 2017. hal-01565129v1

HAL Id: hal-01565129

<https://inria.hal.science/hal-01565129v1>

Preprint submitted on 19 Jul 2017 (v1), last revised 26 Apr 2018 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright



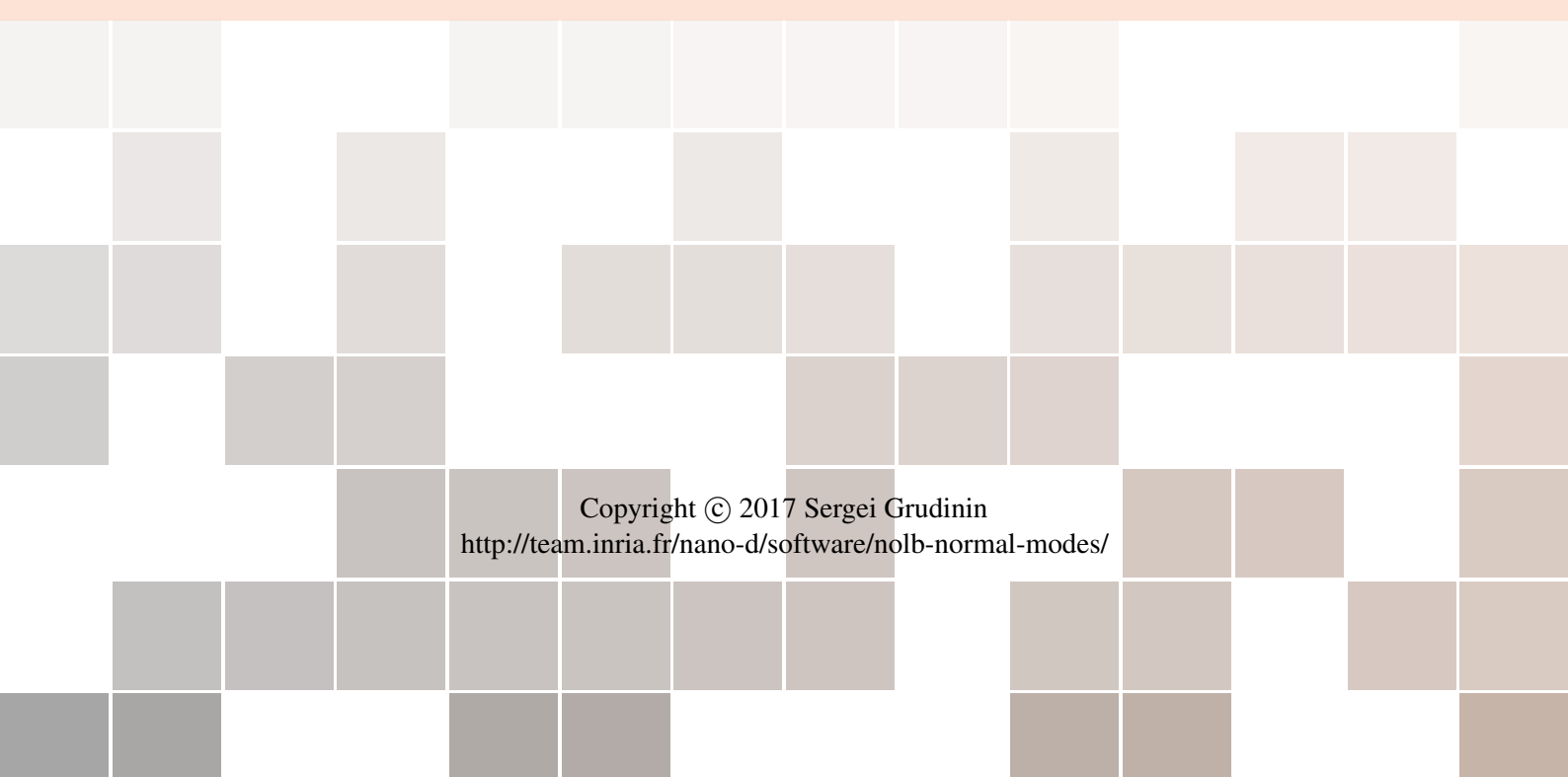
User Guide

NOLB : Non-Linear Normal Mode Analysis, version 0.6, July 2017

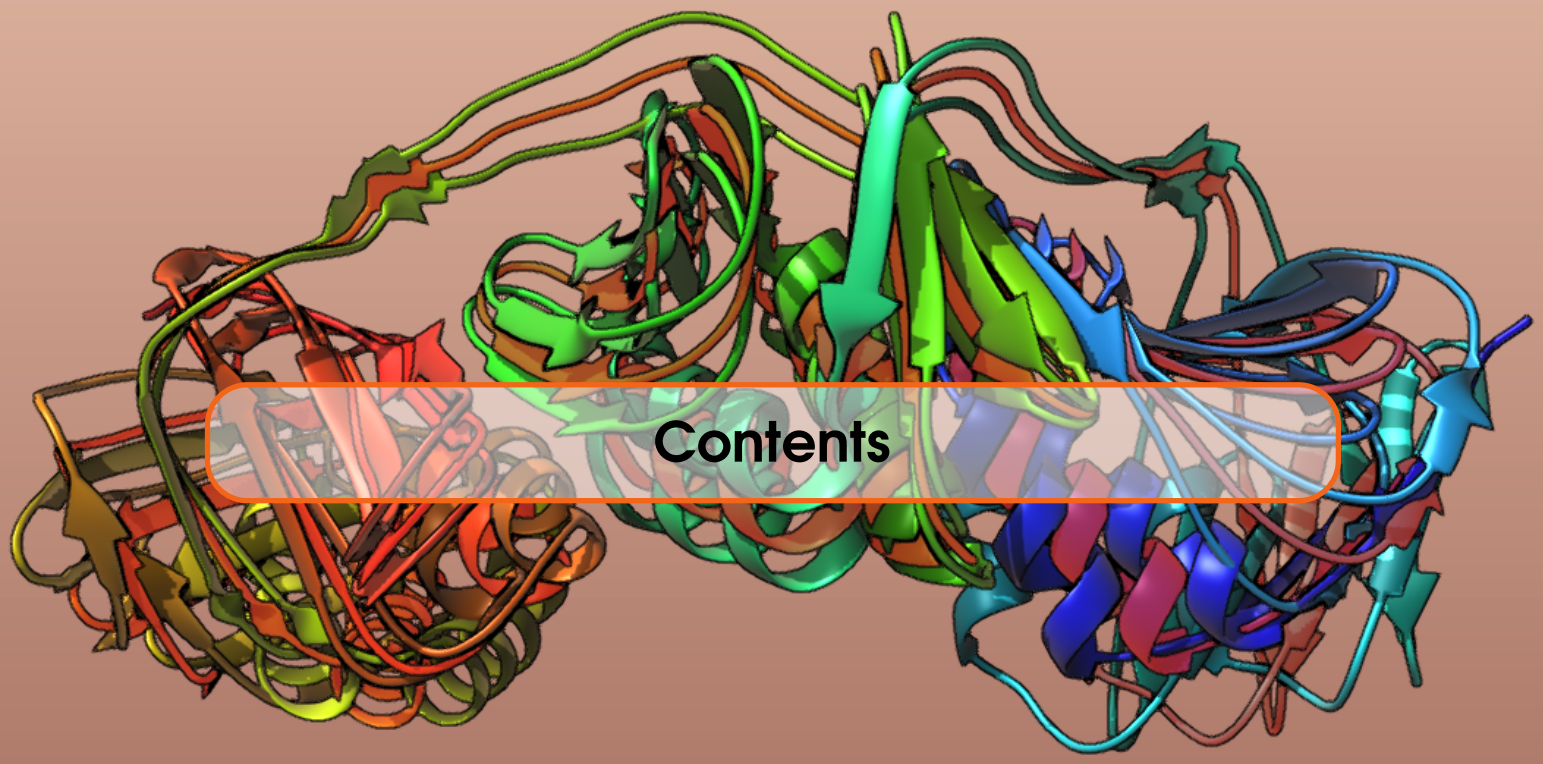
Sergei Grudinin

Inria/CNRS Grenoble, France

email : `sergei.grudinin@inria.fr`

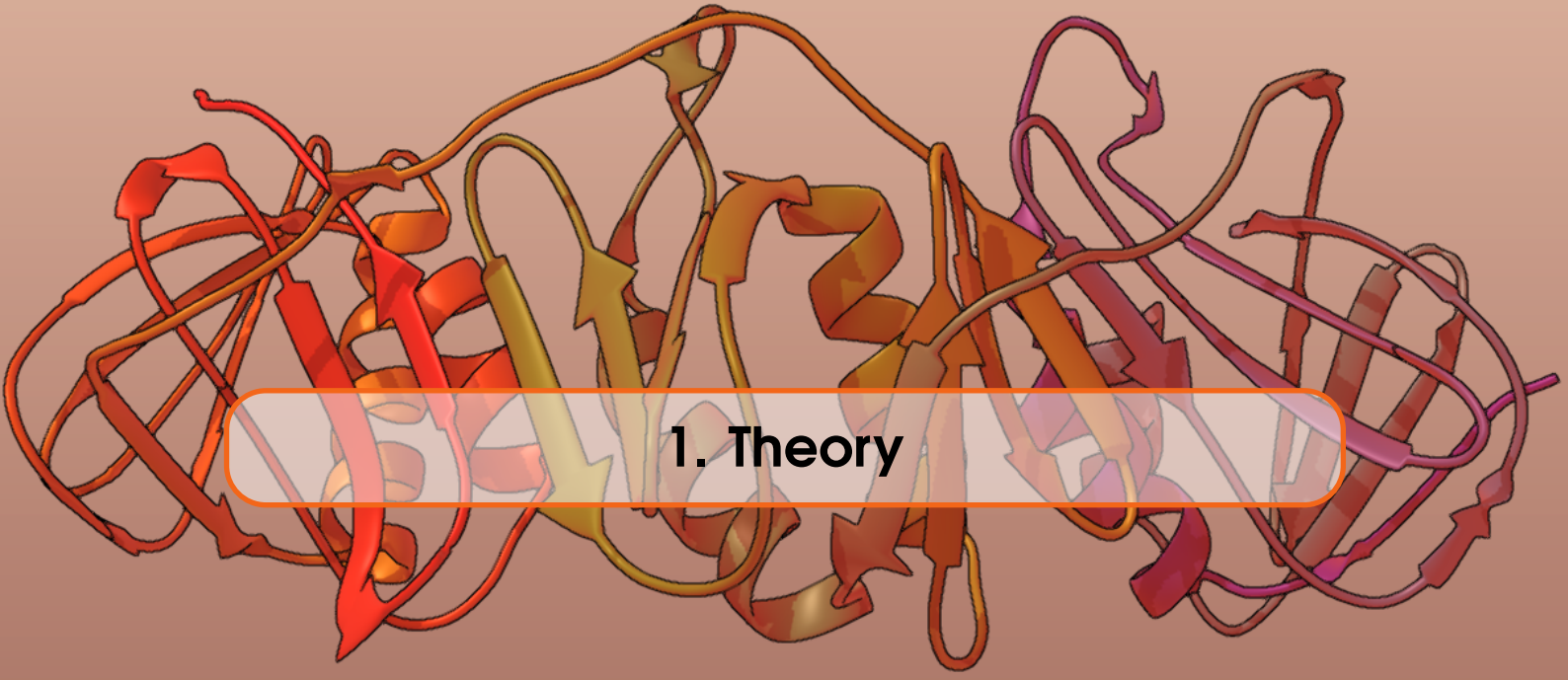


Copyright © 2017 Sergei Grudinin
<http://team.inria.fr/nano-d/software/nolb-normal-modes/>



Contents

| | | |
|----------|--|-----------|
| 1 | Theory | 3 |
| 1.1 | Introduction | 3 |
| 1.2 | NMA theory | 3 |
| 1.3 | RTB projection method | 4 |
| 1.4 | The NOLB method | 5 |
| 1.5 | Potential function | 6 |
| 2 | Program options | 7 |
| 2.1 | Usage | 7 |
| 2.2 | Main options | 9 |
| 2.3 | Experimental options | 11 |
| 2.4 | Currently suppressed options | 11 |
| 3 | Examples | 12 |
| 3.1 | Analysis of basic molecular motions | 12 |
| 3.2 | Comparison with linear NMA | 15 |
| 3.3 | Another example with a bigger system that contains heteroatoms | 15 |
| 3.4 | Finding the best transition between two protein conformations | 16 |
| 3.5 | Analysis of an MD trajectory or NMR ensemble | 17 |
| 3.6 | Analysis of protein docking poses | 18 |
| 3.7 | Structural ensembles | 19 |
| 4 | Related methods | 21 |
| 4.1 | GUI | 21 |
| 4.2 | Morphing paths | 21 |
| 4.3 | SAXS-guided structure optimization | 22 |
| | Index | 23 |



1. Theory

1.1 Introduction

Normal mode analysis (NMA) is an old and well established technique [1] that has recently found many new applications in the field of structural biology and structural bioinformatics [2]. NMA uses a quadratic approximation of the potential energy, and thus it produces linear deformations of the initial structure, which are accurate only for small-amplitude motions. Larger amplitudes can destroy, for example, the secondary structure and break interatomic bonds, when NMA is applied to a protein. An obvious circumvention for this problem will be to take smaller amplitude steps and iteratively recompute and diagonalize the Hessian matrix from the updated positions. This procedure would indeed produce a more realistic deformation of the initial structure thanks to the non-linearity of the obtained deformation. However, such an approach requires multiple diagonalization steps, which may be computationally expensive for some of the applications. Thus, multiple attempts were made to introduce non-linear deformations without the need of multiple diagonalizations. Here we present a new scheme for *non-linear normal mode analysis* that extrapolates a motion computed from *instantaneous linear and angular velocities* to large amplitudes. The scheme can be considered as an evolution of the widely used rotations-translations of blocks (RTB) method.

1.2 NMA theory

Let us consider a molecular system with N_a atoms at an equilibrium position $q_0 \in \mathbb{R}^{3N_a}$. We call $V : \mathbb{R}^{3N_a} \mapsto \mathbb{R}$ the potential energy of our molecular system. We introduce $q : \mathbb{R}^+ \mapsto \mathbb{R}^{3N_a}$, a small time-dependent molecular displacement of our system around q_0 . The potential energy V in the vicinity of q_0 is given by its quadratic approximation, which allows to *analytically* solve the Newton's equation of motion,

$$M(\ddot{q} + \ddot{q}_0) + \nabla V(q_0 + q) \approx M\ddot{q} + Kq = 0, \quad (1.1)$$

where M is the diagonal mass matrix, and K is the *Hessian matrix* (or *stiffness matrix*) of the potential energy V evaluated at the equilibrium position q_0 . We then compute the square matrix of eigenvectors L and the diagonal matrix of eigenvalues Λ of the *mass-weighted stiffness matrix* $K_w = M^{-1/2}KM^{-1/2}$,

$$K_w = L\Lambda L^T. \quad (1.2)$$

We call $\eta : \mathbb{R}^+ \mapsto \mathbb{R}^{3N_a}$ the projection of q into the eigenspace of K_w and we call $(\lambda_i)_{i=0\dots 3N_a}$ the diagonal values in Λ . Then, left multiplication of eq. 1.1 by $L^T M^{1/2}$ gives the following system of

uncoupled equations,

$$\begin{aligned} \eta &= L^T M^{1/2} q \\ \dot{\eta}_i + \lambda_i \eta_i &= 0 \quad i = 1 \dots 3N_a \end{aligned} \quad (1.3)$$

which can be solved by the classical ordinary differential equation (ODE) theory.

1.3 RTB projection method

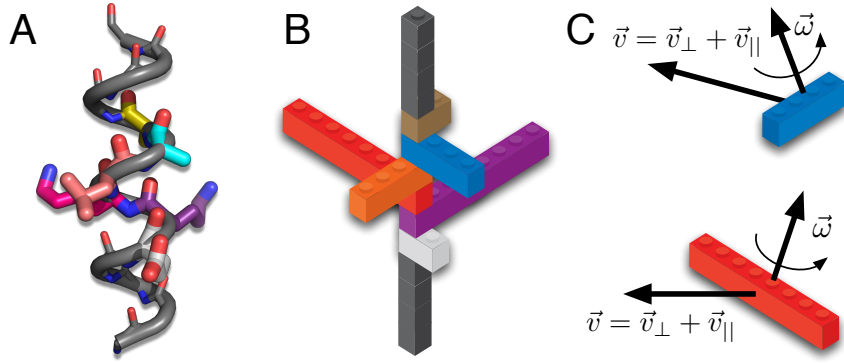


Figure 1.1: (A) All-atom representation of a protein. (B) RTB representation of the same protein. (C) Each rigid block has six degrees of freedom.

The main idea of the RTB projection method is to approximate the system as n rigid blocks. Figure 1.1.A-B schematically shows the all-atom and the RTB representations of the same system. The transition from the RTB coordinate system with $6n$ DOFs to the all-atom coordinate system with $3N$ DOFs is performed by an orthogonal projection matrix $P \in \mathbb{R}^{3N \times 6n}$. The conservation laws of the linear momentum and the angular momentum of a rigid block consisting of N_b atoms written in mass-weighted coordinates are

$$\begin{aligned} \sqrt{M_b} \dot{\tilde{q}} &= \sum_{k=1}^{N_b} \sqrt{m_k} \dot{q}_k \quad \text{for a translation} \\ \mathbf{I}^{1/2} \dot{\tilde{q}} &= \sum_{k=1}^{N_b} \sqrt{m_k} (q_k \times \dot{q}_k) \quad \text{for a rotation} \end{aligned} \quad (1.4)$$

where M_b is the total mass of the rigid block, \mathbf{I} is the rigid block's inertia tensor, \tilde{q} is the blocks's displacement, m_k is the mass of the k^{th} atom of the block, and q_k is the displacement of the k^{th} atom of the block. The elements constituting P^T , the matrix projecting an all-atom motion q into a motion of rigid block \tilde{q} are then obtained by differentiating (1.4) with respect to \dot{q}_k [3]. This leads to translation P_t and rotation P_r matrices of size $3N_b \times 3$ each, computed for each of the rigid blocks and written through their k 3×3 square components,

$$\begin{aligned} P_{tk} &= \sqrt{\frac{m_k}{M_b}} \mathbf{E}_3 \quad \text{for a translation} \\ P_{rk} &= -\sqrt{m_k} (\mathbf{I})^{-1/2} [r_k - r^{COM}]_{\times} \quad \text{for a rotation} \end{aligned} \quad (1.5)$$

where k is one of N_b atom indices, r_k is the position of the corresponding atom in the block, and r^{COM} the position of the block's center of mass (COM). The rigid block's displacement (δ, θ)

6-vector is then obtained by summing up the displacements in the RTB coordinate frame,

$$\begin{aligned}\delta &= \sum_{k=1}^{N_b} \mathbf{P}_{t_k}^T q_k && \text{for a translation} \\ \theta &= \sum_{k=1}^{N_b} \mathbf{P}_{r_k}^T q_k && \text{for a rotation}\end{aligned}\quad (1.6)$$

Having written these equations, we can write the projection matrix P as a diagonal block matrix,

$$P = \begin{pmatrix} \mathbf{P}_t^1 & \mathbf{P}_r^1 & & \\ \cdot & \cdot & \cdot & \\ & & & \mathbf{P}_t^n & \mathbf{P}_r^n \end{pmatrix}. \quad (1.7)$$

The normal modes are then computed by the diagonalization of the RTB-projected mass-weighted stiffness matrix,

$$P^T K_w P = \tilde{L} \tilde{\Lambda} \tilde{L}^T, \quad (1.8)$$

where \tilde{L} is the matrix composed of the RTB normal modes with the corresponding diagonal eigenvalue matrix $\tilde{\Lambda}$. This equation can be rewritten as

$$K_w = (P\tilde{L})\tilde{\Lambda}(P\tilde{L})^T. \quad (1.9)$$

The all-atom normal modes L^w (in mass-weighted coordinates) are then obtained as a projection of RTB normal modes \tilde{L} according to

$$L^w = P\tilde{L}. \quad (1.10)$$

1.4 The NOLB method

Molecular vibrations in a multi-dimensional harmonic oscillator are all uncoupled and can be found by solving (1.3). Diagonalization of the RTB-projected mass-weighted stiffness matrix gives a set of eigenvectors that are composed of *instantaneous linear velocities* and *instantaneous angular velocities* of individual rigid blocks $\{\vec{v}_w, \vec{\omega}_w\}$, see Fig. 1.1.C. For a rigid block with mass M_b and inertia tensor \mathbf{I} , we first compute these in the non-mass weighted coordinates as follows,

$$\begin{aligned}\vec{v} &= M_b^{-1/2} \vec{v}_w \\ \vec{\omega} &= \mathbf{I}^{-1/2} \vec{\omega}_w\end{aligned}\quad (1.11)$$

Then, given a deformation amplitude a , the translational increment in the rigid block's position $\Delta\vec{x}$ and the angular increment in its orientation $\Delta\phi$ can be computed as

$$\begin{aligned}\Delta\vec{x} &= a\vec{v} \\ \vec{n} &= \vec{\omega} / \|\vec{\omega}\|_2, \\ \Delta\phi &= a\|\vec{\omega}\|_2\end{aligned}\quad (1.12)$$

where the rigid block's rotation is described with a unit axis \vec{n} passing through its COM and an angle ϕ . Finally, we rewrite the increment in the rigid block's position $\Delta\vec{x}$ as a sum of two orthogonal vectors,

$$\Delta\vec{x} = \Delta\vec{x}_\perp + \Delta\vec{x}_\parallel, \quad (1.13)$$

where $\Delta\vec{x}_\perp$ is orthogonal to \vec{n} and $\Delta\vec{x}_\parallel$ is collinear to \vec{n} , and represent the $\Delta\vec{x}_\perp$ related motion as a pure rotation about a new center \vec{r} , such that the final rigid blocks's position is

$$\vec{A}' = R(\Delta\phi, \vec{n})(\vec{A} - \vec{r}_0) + \vec{r}_0 + \Delta\vec{x}_\parallel, \quad (1.14)$$

where $R(\Delta\phi, \vec{n})$ is the rotation matrix describing rigid block's rotation about an axis \vec{n} by an angle $\Delta\phi$. See more details in [4].

1.5 Potential function

In principle, our approach can be used with any potential function. Currently, to omit the need of initial system's energy minimization, we have only implemented an all-atom Anisotropic Network Model (ANM) [5, 6], which is a type of elastic network model where the initial structure is always at equilibrium. Figure 1.2 shows a schematic representation of this model. The all-atom ANM has the following potential function,

$$V(q) = \sum_{d_{ij}^0 < R_c} \frac{\gamma}{2} (d_{ij} - d_{ij}^0)^2, \quad (1.15)$$

where d_{ij} is the distance between the i^{th} and the j^{th} atoms, d_{ij}^0 is the reference distance between these atoms, as found in the original structure, γ is the stiffness constant, and R_c is a cutoff distance, typically between 5 Å and 15 Å. The stiffness matrix corresponding to this potential function is composed of the following blocks [2, 5, 6],

$$H_{ij} = -\frac{\gamma}{(d_{ij}^0)^2} \begin{pmatrix} (x_{ij}^0)^2 & x_{ij}^0 y_{ij}^0 & x_{ij}^0 z_{ij}^0 \\ y_{ij}^0 x_{ij}^0 & (y_{ij}^0)^2 & y_{ij}^0 z_{ij}^0 \\ z_{ij}^0 x_{ij}^0 & z_{ij}^0 y_{ij}^0 & (z_{ij}^0)^2 \end{pmatrix} \quad i \neq j, \quad (1.16)$$

$$H_{ii} = -\sum_{j \neq i} H_{ij}$$

where $x_{ij} = x_i - x_j$, $y_{ij} = y_i - y_j$, and $z_{ij} = z_i - z_j$. To rapidly compute this matrix, we use an efficient neighbor search algorithm [7].

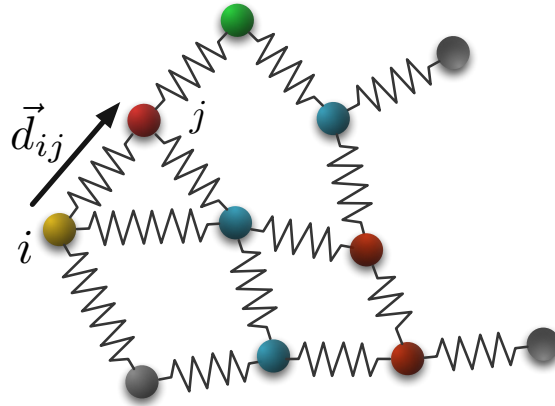


Figure 1.2: Schematic representation of the elastic network model.



2. Program options

2.1 Usage

Typing the '--help' or '-h' flag produces the brief and more detailed description of the program options,

```
NOLB --help
```

```
*****
*-----*
*-----NOLB : a Non-Linear Rigid Block NMA method-----*
*-----Authors: Alexandre Hoffmann & Sergei Grudinin-----*
*-----Copyright (c): Nano-D team, Inria/CNRS Grenoble, France, 2017.---*
*-----e-mail: sergei.grudinin@inria.fr -----*
*----- http://team.inria.fr/nano-d/software/nolb-normal-modes/ -----*
*-----*
*****

USAGE:

NOLB <pdb filename> <pdb reference file> [-o <output name>] [-n <number
of modes>] [-c <cutoff distance>] [-s <number of output frames>]
[-a <maximum amplitude>] [-p <potential function>] [-m] [--linear]
[--trajectory] [--hetatm] [--zdock <Zdock transform filename>]
[--hex <Hex transform filename>] [--frames <max number of input
trajectory frames>] [--covScaling <scaling of covalent
interactions>] [--coords <input traj file>] [--weights <input
weights file>] [-r <maximum rmsd>] [--dist <RMSD distribution>]
[--nSteps <number of minimization steps>] [-t <minimization
tolerance>] [-h] [--version] [-log]

Where:

<pdb filename>
  (required) Input PDB file, can be a PDB multi-model trajectory file

<pdb reference file>
  Reference PDB file. Only to compute the least RMSD motion. Disabled by
  default.

-o <output name>, --output <output name>
  Output file name. If nothing provided, a default name will be chosen.

-n <number of modes>, --modes <number of modes>
  Number of modes, 10 by default
```

```
-c <cutoff distance>, --cutoff <cutoff distance>
  Interactions cutoff distance in angstroms, 5 by default

-s <number of output frames>, --solutions <number of output frames>
  Number of frames in the output NMA trajectory, 10 by default

-a <maximum amplitude>, --amplitude <maximum amplitude>
  Maximum amplitude, 1.0 by default. The absolute amplitude value is
  chosen automatically according to the system size. This option only
  adjusts the absolute value.

-p <potential function>, --potential <potential function>
  Potential function, 0 = Elastic Network, 1 = Gaussian Network.
  Currently disabled.

-m, --minimize
  Minimize obtained structures, off by default.

--linear
  Additionally compute the linear modes. Off by default.

--trajectory
  Save the output trajectory at equidistant points. Off by default.

--hetatm
  Read heteroatoms from the input file. False by default.

--zdock <Zdock transform filename>
  Zdock transforms file name. Experimental option.

--hex <Hex transform filename>
  Hex transforms file name. Experimental option.

--frames <max number of input trajectory frames>
  Max number of trajectory frames to use for trajectory analysis, 100 by
  default.

--covScaling <scaling of covalent interactions>
  Scaling of covalent interactions, 1 by default.

--coords <input traj file>
  Name of the text trajectory coordinate file that should correspond to
  input pdb. Experimental option.

--weights <input weights file>
  Name of the trajectory weights text file, where weights for different
  frames are provided. Experimental option.

-r <maximum rmsd>, --rmsd <maximum rmsd>
  Maximum rmsd for decoy generation.

--dist <RMSD distribution>
  Power dependence of RMSD distribution of generated decoys on the input
  RMSD, 0 for constant RMSD value (default), 1 for linear dependence,
  0.5 for quadratic dependence, etc.

--nSteps <number of minimization steps>
  Number of minimization steps, 500 by default

-t <minimization tolerance>, --tol <minimization tolerance>
  Minimization tolerance, 0.1 by default

-h, --help
  Displays usage information and exits.

--version
  Displays version information and exits.

-log, --changelog
  Displays ChangeLog information and exits.
```

Typing '`--version`' prints the current version of the program,

```
NOLB --version
```

```
NOLB version: 0.6, July 2017
```

Typing the '`-log`' or '`--log`' prints the changelog of the program,

```
NOLB -log
```

```
NOLB ChangeLog:
Version 0.1 from Feb 2016:
  Initial release.
Version 0.2 from March 2016:
  Added initial support of PDB trajectories.
Version 0.3 from Feb 2017:
  Added reading of hetero atoms. Adapted for SAMSON GUI. Multiple tests performed.
  Compiled on Linux, Win32 and MacOS.
Version 0.4 from May 2017:
  Added automatic tests for convergence. More of nonlinear PCA support.
Version 0.5 from June 2017:
  Added output of linear modes.
Version 0.6 from July 2017:
  Added generation of NMA decoys.
```

2.2 Main options

```
<pdb filename> (required)
```

There is only one required argument, the path to the input PDB file. This can alternatively be a PDB multi-model trajectory file. In this case, the NMA will be performed based on the covariance matrix of the trajectory.

```
<pdb reference file>
```

This options provides the path to the reference PDB file. It is only used to compute the least-RMSD transition path. Disabled by default.

```
-o <output name>, --output <output name>
```

This options provides the output file name. If nothing provided, a default name based on the name of the input file will be chosen.

```
-n <number of modes>, --modes <number of modes>
```

This options provides the number of non-trivial modes to compute, 10 by default. If this number exceeds the size of the Hessian matrix, it will be adapted accordingly.

```
-c <cutoff distance>, --cutoff <cutoff distance>
```

This options specifies the interaction cutoff distance for the elastic network models (in angstroms), 5 by default. The Hessian matrix is constructed according to this interaction distance. Some artifacts should be expected for too short distances ($< 5 \text{ \AA}$).


```
-s <number of output frames>, --solutions <number of output frames>
```

This options provides the number of frames in the output NMA trajectory, 10 by default. The output trajectory is written in the PDB file with multiple models. This flag specifies the number of models in the output file. It also controls the number of models if pseudo-random decoys are generated.

```
-a <maximum amplitude>, --amplitude <maximum amplitude>
```

This options sets the maximum relative amplitude of NMA motion, 1.0 by default. The absolute amplitude value is chosen automatically according to the system size. This flag only adjusts the absolute value.

```
--linear
```

Flag to additionally compute the linear modes. Off by default. If computed, these are stored separately.

```
--hetatm
```

Flag to read heteroatoms from the input file. False by default. Should be used for lipids, small molecules, etc.

```
--trajectory
```

Flag to save the output trajectory at equidistant points. Off by default. If this option is disabled, then the trajectory is written according to the natural harmonic motion of the oscillator.

```
--frames <max number of input trajectory frames>
```

This is the option to choose the maximum number of trajectory frames to use for trajectory analysis, 100 by default. Only takes the effect if the input file contains multiple models, or if the output of docking programs is analyzed.

```
-r <maximum rmsd>, -rmsd <maximum rmsd>
```

If pseudo-random decoys are needed, this option specifies the maximum RMSD of the generated decoys.

```
-dist <RMSD distribution>
```

This option defines the distribution of the decoys RMSD, 0 for constant RMSD (default), 0.5 for quadratic dependence, 1 for linear dependence, etc.

```
-m, -minimize
```

Flag to minimize the obtained structures, off by default.

```
-nSteps <number of minimization steps>
```

This option sets the number of minimization steps, 500 by default.

```
-t <minimization tolerance>, -tol <minimization tolerance>
```

This option sets the minimization tolerance, 0.1 by default.

```
-h, --help
```

Flag to display usage information and exit.

```
--version
```

Flag to display version information and exit.

```
-log, --changelog
```

Flag to display the ChangeLog information and exit.

2.3 Experimental options

```
--zdock <Zdock transform filename>
```

This flag provides the path to the Zdock transforms file. Zdock protein docking poses are analyzed then using the covariance matrix.

```
--hex <Hex transform filename>
```

This flag provides the path to the Hex transforms file. Hex protein docking poses are analyzed then using the covariance matrix.

```
--coords <input traj file>
```

Name of the trajectory coordinate file that should correspond to the input pdb file. This one is used for the analysis of the covariance matrix.

```
--weights <input weights file>
```

Name of the trajectory weights text file, where weights for different frames are provided.

2.4 Currently suppressed options

```
-p <potential function>, --potential <potential function>
```

Potential function, 0 = Elastic Network, 1 = Gaussian Network. Currently, the ANM model is only used.

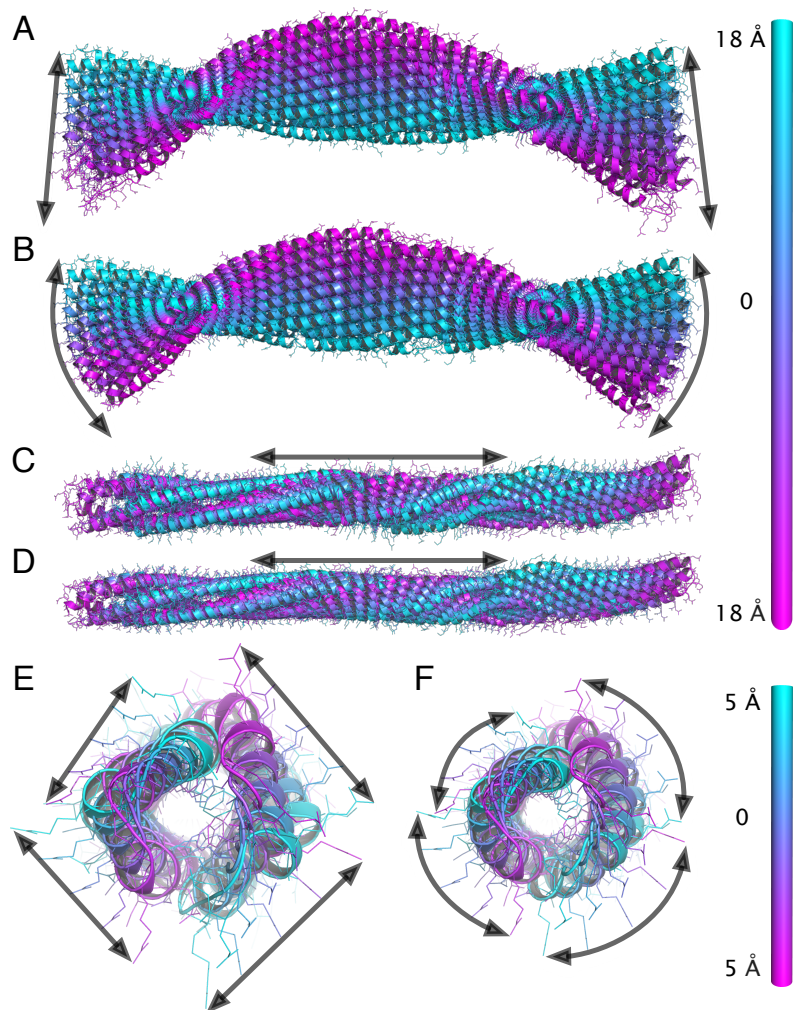
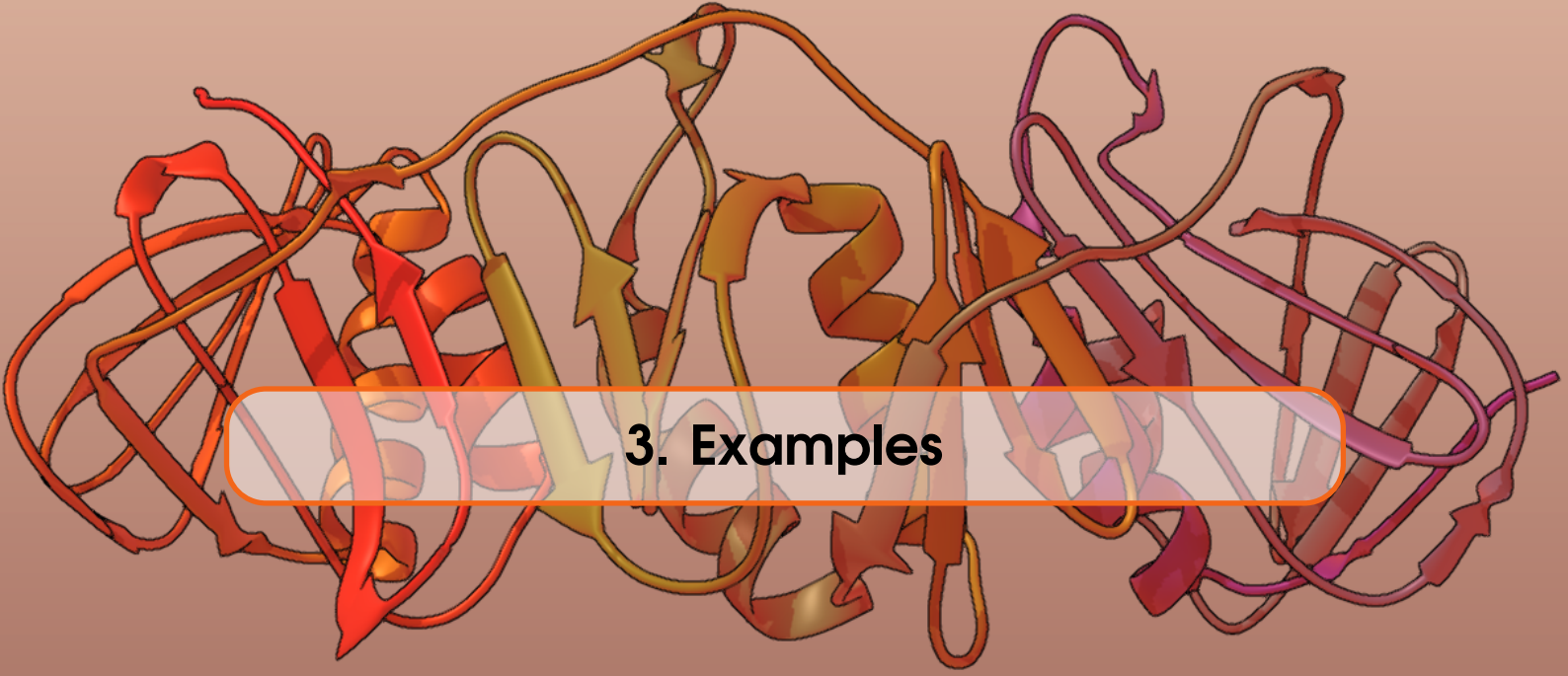


Figure 3.1: Comparison of linear (A, C, E) and non-linear (B, D, F) normal modes computed for a coiled coil protein (pdb code 2ch7). Three types of motions are shown, bending (A, B), stretching (C, D), and twisting (E, F). Several snapshots at different deformation amplitudes are superposed to each other. These are colored according to the values of the overall deformation, as measured by the RMSD. The colorbars show the RMSD with respect to the initial position. The arrows follow the trajectories of individual atoms.

3.1 Analysis of basic molecular motions

We will start with the NOLB NMA of a mid-size coiled-coil protein from the cytoplasmic domain of a bacterial chemoreceptor (pdb code 2ch7), which clearly demonstrates the three basic types of molecular vibration motions. These are bending, stretching and twisting.

Example 1 To compute the first 10 non-trivial normal modes for a 2ch7.pdb protein structure, type

```
NOLB 2ch7.pdb
```

This will produce the following output in the terminal (we suppose you are starting the program from the terminal!),

```
*****
*-----NOLB : a Non-Linear Rigid Block NMA method-----*
*-----Authors: Alexandre Hoffmann & Sergei Grudinin-----*
*--Copyright (c): Nano-D team, Inria/CNRS Grenoble, France, 2017.--*
*----- e-mail: sergei.grudinin@inria.fr -----*
*--- http://team.inria.fr/nano-d/software/nolb-normal-modes/ ---*
*-----*
*****
=====Reading input PDB file=====
Input PDB file..... : 2ch7.pdb
Number of chains read..... : 3
Number of atoms read..... : 4630
Found..... : 4641 covalent bonds
..... : 6264 angles
=====Constructing Hessian=====
Number of disconnected atoms..... : 0
Number of interactions..... : 631042
Memory required for the Hessian matrix..... : 129.991 Mb
=====Doing NMA=====
Memory required for the RTB projection matrix..... : 1.27167 Mb
Reduced Hessian size : ..... : 3702 x 3702
Number of computed modes..... : 10
Null-space size..... : 1
Lowest mode frequency..... : 0.000535409
Highest mode frequency..... : 0.00412211
Frequencies :
  [ 1]..... : 0.000535409
  [ 2]..... : 0.000542863
  [ 3]..... : 0.00135327
  [ 4]..... : 0.00153196
  [ 5]..... : 0.00167521
  [ 6]..... : 0.00261288
  [ 7]..... : 0.00278308
  [ 8]..... : 0.00347724
  [ 9]..... : 0.003539
  [10]..... : 0.00412211
=====Writing output PDB files=====
Using output mask..... : 2ch7
Using output mask for nonlinear modes..... : 2ch7_nlb_
=====
===== Timing : =====
=====
Parsing arguments..... : 0.001036 s
Reading input PDB file..... : 0.015923 s
Constructing Hessian..... : 0.549772 s
Doing NMA..... : 0.797914 s
Writing output PDB files..... : 0.775731 s
.....
Total time : ..... : 2.14038 s
=====
```

The first section lists the structural parameters, the number of atoms, chains, and the number of

bonds and angles in the structure. These last two can be used during energy relaxation, which is currently disabled. Water molecules are ignored and not listed, heteroatoms can be read with an additional flag '--hetatm':

```
=====Reading input PDB file=====
Input PDB file..... : 2ch7.pdb
Number of chains read..... : 3
Number of atoms read..... : 4630
Found..... : 4641 covalent bonds
..... : 6264 angles
```

The second section lists parameters of the Hessian matrix. These are the number of interacting atom pairs, which can be controlled with the '--cutoff' flag, the number of disconnected atoms, which are stayed static, and the total memory required for the Hessian:

```
=====Constructing Hessian=====
Number of disconnected atoms..... : 0
Number of interactions..... : 631042
Memory required for the Hessian matrix..... : 129.991 Mb
```

The next section lists parameters of the RTB basis and the diagonalization procedure. It provides the size of the projection matrix, the size of the RTB Hessian, the size of the detected null-space, the number of modes and the corresponding frequencies. These are the square roots of the corresponding eigenvalues. The null-space is usually not detected fully, it should be 5 or 6 for non-symmetric systems, and skipped from the further analysis:

```
=====Doing NMA=====
Memory required for the RTB projection matrix..... : 1.27167 Mb
Reduced Hessian size : ..... : 3702 x 3702
Number of computed modes..... : 10
Null-space size..... : 1
Lowest mode frequency..... : 0.000535409
Highest mode frequency..... : 0.00412211
Frequencies :
[ 1]..... : 0.000535409
[ 2]..... : 0.000542863
[ 3]..... : 0.00135327
[ 4]..... : 0.00153196
[ 5]..... : 0.00167521
[ 6]..... : 0.00261288
[ 7]..... : 0.00278308
[ 8]..... : 0.00347724
[ 9]..... : 0.003539
[10]..... : 0.00412211
```

The next section lists the masks of the output files:

```
=====Writing output PDB files=====
Using output mask..... : 2ch7
Using output mask for nonlinear modes..... : 2ch7_nlb_
```

The output pdb trajectory files have the following names, '2ch7_nlb_1.pdb', '2ch7_nlb_2.pdb', etc. The number in the name refers to the normal mode id starting from 1. The output mask can be changed with the '-o newMask' flag. Finally, the last section lists the timings of the program, split into individual contributions:

```
===== Timing : =====
=====
Parsing arguments..... : 0.001036 s
Reading input PDB file..... : 0.015923 s
Constructing Hessian..... : 0.549772 s
Doing NMA..... : 0.797914 s
Writing output PDB files..... : 0.775731 s
```

```
.....
Total time : ..... : 2.14038 s
=====
```

Here, the most time consuming operation is typically the diagonalization step. Writing trajectories can be very long as well, depending on the number of required modes and frames in the trajectories. Figure 3.1 shows some of the computed modes.

3.2 Comparison with linear NMA

Example 2 In this example we will compare the NOLB normal modes with the linear normal modes. For this, we will enable the '--linear' flag, and will also output more trajectory frames for a better visual perception of the motions using the '-s 20' flag. Finally, we will increase the motion amplitude by a factor of 2 with the '-a 2' flag:

```
NOLB 2ch7.pdb -s 20 -a 2 --linear
```

The output will be only different when listing the output files,

```
=====Writing output PDB files=====
Using output mask..... : 2ch7
Using output mask for linear modes..... : 2ch7_linear_
Using output mask for nonlinear modes..... : 2ch7_nlb_
```

Now, we have also computed the linear NMA trajectories stored in '2ch7_linear_1.pdb', '2ch7_linear_2.pdb', etc. Figure 3.1 shows the visual difference between the computed linear and non-linear modes.

3.3 Another example with a bigger system that contains heteroatoms

Example 3 In this example we compute the first 50 normal modes for a larger system, the photosystem II complex (pdb code 5b5e) with about 50,000 of atoms,

```
NOLB 5b5e.pdb -n 50
```

This will produce the following output, which we will cut for brevity,

```
=====Reading input PDB file=====
Input PDB file..... : 5b5e.pdb
Number of chains read..... : 39
Number of atoms read..... : 40908
Found..... : 42093 covalent bonds
..... : 57363 angles
=====Constructing Hessian=====
Number of disconnected atoms..... : 0
Number of interactions..... : 6424422
Memory required for the Hessian matrix..... : 1323.39 Mb
=====Doing NMA=====
Lowest mode frequency..... : 0.00156711
Highest mode frequency..... : 0.00784686
Frequencies :
    [ 1]..... : 0.00156711
    [ 2]..... : 0.00175741
===== Timing : =====
=====
Parsing arguments..... : 0.000678 s
Reading input PDB file..... : 0.099499 s
Constructing Hessian..... : 7.44606 s
Doing NMA..... : 29.754 s
Writing output PDB files..... : 28.5004 s
.....
```



```

=====Aligning Trajectory Frames=====
=====Constructing the Covariance matrix=====
Memory Required for the Covariance Matrix..... : 13.4146 Mb
Memory required for the RTB projection matrix..... : 0.121399 Mb
===== Timing : =====
=====
Parsing arguments..... : 0.000613 s
Reading input PDB file..... : 0.003657 s
Reading Trajectory..... : 0.005014 s
Aligning Trajectory Frames..... : 0.000192 s
Constructing the Covariance matrix..... : 0.00419 s
Doing NMA..... : 0.000838 s
Writing output PDB files..... : 0.112159 s
.....
Total time : ..... : 0.126667 s
=====

```

You may see that the trajectory analysis is very fast, the only noticeable time here is actually the output of the computed motions.

3.6 Analysis of protein docking poses

In our next example, we will compute principal components of protein-protein docking poses. We have chosen a standard example from the protein docking benchmark (pdb code 1d6r, with the receptor 1D6R_r_b.pdb and the ligand 1D6R_l_b.pdb). We have computed the first 1,000 of docking poses with the popular zDock package, Hex can be used as well here.

Example 7 To visualize the principal components of the computed docking poses type

```
NOLB outName.pdb --zdock ./1D6R.zdock.out
```

This produces the following output,

```

=====Reading Zdock Transforms=====
=====Reading PDB files=====
Reading receptor PDB file..... : ./1D6R_r_b.pdb
Number of receptor chains read..... : 1
Number of receptor atoms read..... : 1629
Reading ligand PDB file..... : ./1D6R_l_b.pdb
Number of ligand chains read..... : 1
Number of ligand atoms read..... : 427
=====Reading Docking Trajectory=====
Number of frames read..... : 100
=====Doing NMA=====
Number of computed modes..... : 5
Lowest mode frequency..... : 2.39176e-05
Highest mode frequency..... : 0.00031764
Frequencies :
    [ 1]..... : 2.39176e-05
    [ 2]..... : 5.74957e-05
    [ 3]..... : 8.65304e-05
    [ 4]..... : 0.000198992
    [ 5]..... : 0.00031764
=====Writing output PDB files=====
Using only..... : 5 modes
Using output mask..... : outName
Using output mask for nonlinear modes..... : outName_nlb

```

We should mention that this is just an example how to use the NOLB method for a simple PCA analysis, which probably cannot be used for practical applications.

3.7 Structural ensembles

Another application of the NOLB method is the generation of structural ensembles (also called 'decoys') with either a fixed deformation RMSD, or a certain distribution of these. Figure 3.2 shows a sample output of the method.

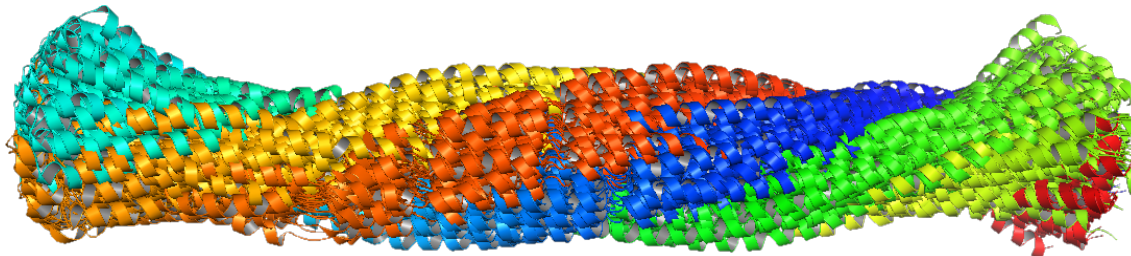


Figure 3.2: Illustration of 100 decoys of a coiled-coil system 2ch7 protein with RMSD of 10 Å from the starting structure.

Example 8 In the first example we will generate 100 decoys of a coiled-coil system at 10 Å from the starting structure.

```
NOLB 2ch7.pdb -s 100 --rmsd 10
```

This creates a '2ch7_nlb_decoys.pdb' pdb trajectory file with 100 models with the following output,

```
=====-Writing output Decoy files=====
Number of files..... : 100
Output RMSD..... : 10 A
Using outputfilename..... : 2ch7_nlb_decoys.pdb
=====-
=====- Timing : =====
=====-
Parsing arguments..... : 0.000178 s
Reading input PDB file..... : 0.006173 s
Constructing Hessian..... : 0.0752 s
Doing NMA..... : 0.605818 s
Writing output Decoy files..... : 0.848708 s
.....
Total time : ..... : 1.53608 s
=====-
```

Example 9 In the second example we will generate 100 decoys of a the same system with RMSD distributed linearly from 0 to 10 Å compared to the starting structure. We will also minimize the obtained decoys.

```
NOLB 2ch7.pdb -s 100 --rmsd 10 --dist 1 -m
```

This creates a '2ch7_nlb_decoys.pdb' pdb trajectory file with 100 models with the following output,

```
=====-Writing output Decoy files=====
Number of files..... : 100
Output RMSD..... : 10 A
Using outputfilename..... : 2ch7_nlb_decoys.pdb
Current energy..... : 0.272494
N minimization steps..... : 0
Current Gradient norm..... : 0.0210788
```

```
Displacement ..... : 0.00167775
RMSD after minimization..... : 2.45655
---
Current energy..... : 6.27796
N minimization steps..... : 1
Current Gradient norm..... : 0.0738622
Displacement ..... : 0.0135281
RMSD after minimization..... : 7.13781
=====
===== Timing : =====
=====
Parsing arguments..... : 0.000224 s
Reading input PDB file..... : 0.009928 s
Constructing Hessian..... : 0.069928 s
Doing NMA..... : 0.603946 s
Writing output Decoy files..... : 2.1154 s
.....
Total time : ..... : 2.79943 s
=====
```

4. Related methods

4.1 GUI

A graphical user interface created for the SAMSON software platform is available at <https://www.samson-connect.net>. Figure 4.1 shows the current version of this GUI.

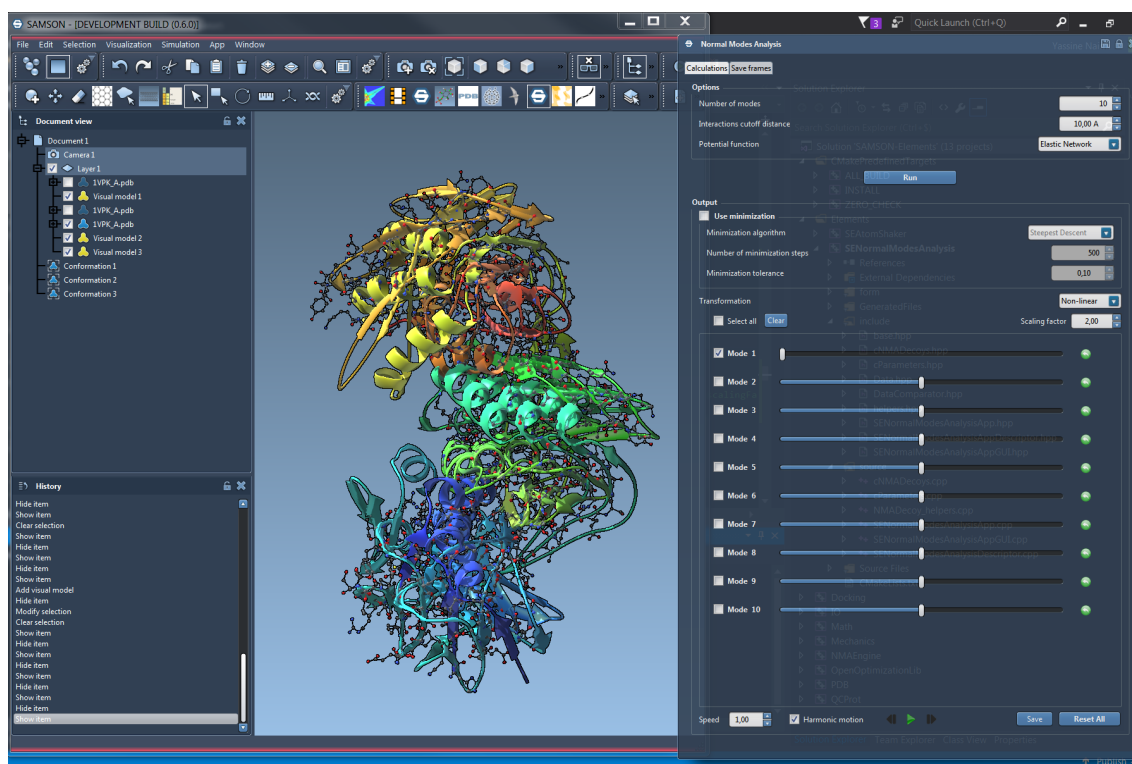


Figure 4.1: NOLB graphical user interface for the SAMSON software platform.

4.2 Morphing paths

The NOLB normal modes can be used to produce morphing pathways between multiple conformations of biological molecules. It is not a well tested approach yet. However, in the future it can be merged with the NOLB program.

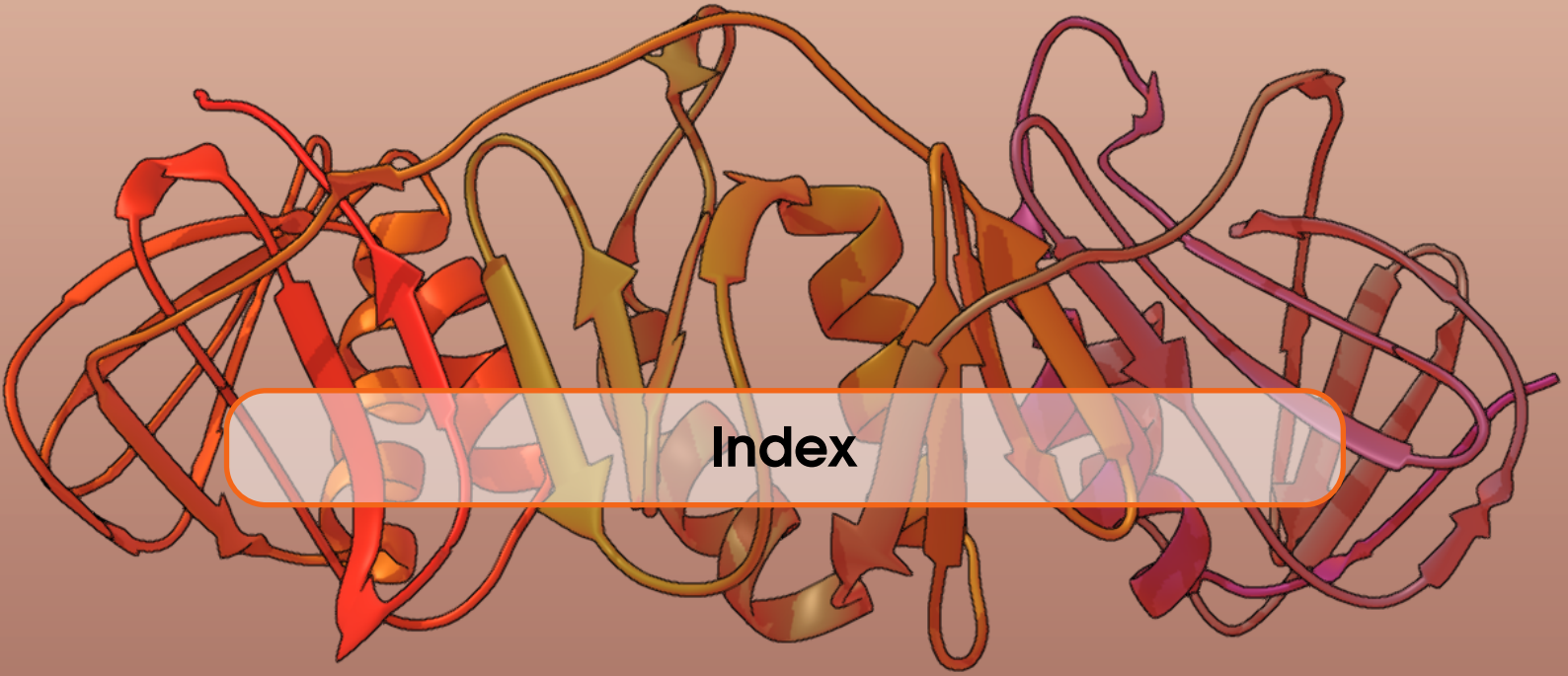
4.3 SAXS-guided structure optimization

We have been also using the NOLB normal modes as a low-dimensional representation of the protein motion subspace in the recent CASP12 blind structure prediction exercise. More precisely, we used for in the SAXS-assisted protein targets, where we numerically computed the gradient of the SAXS loss function and optimized the structures towards the gradients in this low-dimensional space. This method will be distributed separately as a part of our Pepsi-SAXS scattering package at <https://team.inria.fr/nano-d/software/pepsi-saxs/>.



Bibliography

- [1] E Bright Wilson, J C Decius, and Paul C Cross. *Molecular Vibrations: The Theory of Infrared and Raman Spectra*. McGraw-Hill, 1955 (cited on page 3).
- [2] Ivet Bahar et al. “Normal Mode Analysis of Biomolecular Structures: Functional Mechanisms of Membrane Proteins”. In: *Chem. Rev.* 110.3 (Mar. 2010), pages 1463–1497. DOI: 10.1021/cr900095e. URL: <https://doi.org/10.1021%7B%7D2Fcr900095e> (cited on pages 3, 6).
- [3] Timothy R Lezon et al. “Elastic Network Models for Biomolecular Dynamics: Theory and Application to Membrane Proteins and Viruses”. In: *Handbook on Biological Networks*. World Scientific Pub Co Pte Lt, Dec. 2009, pages 129–158. DOI: 10.1142/9789812838803_0007. URL: http://dx.doi.org/10.1142/9789812838803%7B%5C_%7D0007 (cited on page 4).
- [4] Alexandre Hoffmann and Sergei Grudinin. “NOLB: Nonlinear Rigid Block Normal-Mode Analysis Method”. In: *J. Chem. Theory Comput.* 13.5 (2017), pages 2123–2134 (cited on page 5).
- [5] Pemra Doruker, Ali Rana Atilgan, and Ivet Bahar. “Dynamics of Proteins Predicted by Molecular Dynamics Simulations and Analytical Approaches: Application to α -Amylase Inhibitor”. In: *Proteins: Struct., Funct., Bioinf.* 40.3 (2000), pages 512–524 (cited on page 6).
- [6] A R Atilgan et al. “Anisotropy of Fluctuation Dynamics of Proteins with an Elastic Network Model”. In: *Biophys. J.* 80.1 (Jan. 2001), pages 505–515. DOI: 10.1016/s0006-3495(01)76033-x. URL: <https://doi.org/10.1016%7B%7D2Fs0006-3495%7B%7D2801%7B%7D2976033-x> (cited on page 6).
- [7] Svetlana Artemova, Sergei Grudinin, and Stephane Redon. “A Comparison of Neighbor Search Algorithms for Large Rigid Molecules”. In: *J. Comput. Chem.* 32.13 (2011), pages 2865–2877 (cited on page 6).



Index

A

Analysis of basic molecular motions 15
 Analysis of protein docking poses 20
 Analysis of trajectories 19

C

Comparison with linear NMA 17
 Currently suppressed options 12

D

Decoys 21

E

Example with a bigger system that contains heteroatoms 17
 Experimental options 12

F

Finding the best transition between two protein conformations 18

G

GUI 21

H

Heteroatoms 18

I

Introduction 4

M

Main options 11
 Morphing paths 22

N

NMA theory 5

R

RTB projection method 5

S

SAXS-guided structure optimization 22
 Structural ensembles 21

T

The NOLB method 6

U

Usage 9