Understanding function by

- Recognition of functional sequences or motifs
- Analysis of different structures in different forms
- Computational analyses and simulations
Molecular dynamics (MD) simulations

- A deterministic method based on the solution of Newton’s equation of motion

\[ F_i = m_i a_i \]

for the \( i \)th particle; the acceleration at each step is calculated from the negative gradient of the overall potential, using

\[ F_i = - \nabla V_i \]
\( V_i = \sum_k (\text{energies of interactions between } i \text{ and all other residues } k \text{ located within a cutoff distance of } R_c \text{ from } i) \)

\[ \nabla V_i = \text{Gradient of potential?} \]

- Derivative of \( V \) with respect to the position vector \( r_i = (x_i, y_i, z_i)^T \) at each step

\[
\begin{align*}
a_{xi} & \sim -\frac{\partial V}{\partial x_i} \\
a_{yi} & \sim -\frac{\partial V}{\partial y_i} \\
a_{zi} & \sim -\frac{\partial V}{\partial z_i}
\end{align*}
\]
Interaction potentials include;

**Non-Bonded Interaction Potentials**
- Electrostatic interactions of the form $E_{ik}(es) = q_i q_k / r_{ik}$
- Van de Waals interactions $E_{ij}(vdW) = -a_{ik} / r_{ik}^6 + b_{ik} / r_{ik}^{12}$

**Bonded Interaction Potentials**
- Bond stretching $E_i(bs) = (k_{bs}/2) (l_i - l_i^0)^2$
- Bond angle distortion $E_i(bad) = (k_{\theta}/2) (\theta_i - \theta_i^0)^2$
- Bond torsional rotation $E_i(tor) = (k_{\phi}/2) f(\cos \phi_i)$
Example 1: gradient of vdW interaction with \( k \), with respect to \( r_i \)

- \( E_{ik}^{(vdW)} = -a_{ik}/r_{ik}^6 + b_{ik}/r_{ik}^{12} \)
- \( r_{ik} = r_k - r_i \)
  - \( x_{ik} = x_k - x_i \)
  - \( y_{ik} = y_k - y_i \)
  - \( z_{ik} = z_k - z_i \)
  - \( r_{ik} = [(x_k - x_i)^2 + (y_k - y_i)^2 + (z_k - z_i)^2]^{1/2} \)
- \( \partial V/\partial x_i = \partial \left[ -a_{ik}/r_{ik}^6 + b_{ik}/r_{ik}^{12} \right] / \partial x_i \)

where \( r_{ik}^6 = [(x_k - x_i)^2 + (y_k - y_i)^2 + (z_k - z_i)^2]^3 \)
Example 2: gradient of bond stretching potential with respect to \( r_i \)

\[
E_i(\text{bs}) = \left( \frac{k_{bs}}{2} \right) (l_i - l_i^0)^2
\]

\[
l_i = r_{i+1} - r_i
\]

\[
l_{ix} = x_{i+1} - x_i
\]

\[
l_{iy} = y_{i+1} - y_i
\]

\[
l_{iz} = z_{i+1} - z_i
\]

\[
l_i = \sqrt{ (x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2 }^{1/2}
\]

\[
\frac{\partial E_i(\text{bs})}{\partial x_i} = - m_i a_{ix}(\text{bs}) \quad \text{(induced by deforming bond } l_i\text{)}
\]

\[
= \left( \frac{k_{bs}}{2} \right) \frac{\partial}{\partial x_i} \left\{ \left[ (x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2 \right]^{1/2} - l_i^0 \right\}^2 / \partial x_i
\]

\[
= k_{bs} (l_i - l_i^0) \frac{\partial}{\partial x_i} \left\{ \left[ (x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2 \right]^{1/2} - l_i^0 \right\} / \partial x_i
\]

\[
= k_{bs} (l_i - l_i^0) (1/2) (l_i^{-1}) \frac{\partial}{\partial x_i} (x_{i+1} - x_i)^2 / \partial x_i = - k_{bs} \left( 1 - l_i^0 / l_i \right) (x_{i+1} - x_i)
\]
The Verlet algorithm

Perhaps the most widely used method of integrating the equations of motion is that initially adopted by Verlet [1967]. The method is based on positions \(r(t)\), accelerations \(a(t)\), and the positions \(r(t-\delta t)\) from the previous step. The equation for advancing the positions reads as

\[
    r(t+\delta t) = 2r(t) - r(t-\delta t) + \delta t^2 a(t)
\]

There are several points to note about this equation. It will be seen that the velocities do not appear at all. They have been eliminated by addition of the equations obtained by Taylor expansion about \(r(t)\):

\[
    r(t+\delta t) = r(t) + \delta t v(t) + (1/2) \delta t^2 a(t) + ... \\
    r(t-\delta t) = r(t) - \delta t v(t) + (1/2) \delta t^2 a(t) -
\]

The velocities are not needed to compute the trajectories, but they are useful for estimating the kinetic energy (and hence the total energy). They may be obtained from the formula

\[
    v(t) = \frac{[r(t+\delta t) - r(t-\delta t)]}{2\delta t}
\]
Periodic boundary conditions
Initial velocities ($v_i$)

using the Boltzmann distribution at the given temperature

$$v_i = \left(\frac{m_i}{2\pi kT}\right)^{1/2} \exp\left(-\frac{m_i v_i^2}{2kT}\right)$$
How to generate MD trajectories?

- Known initial conformation, i.e. $r_i(0)$ for all atom $i$
- Assign $v_i(0)$, based on Boltzmann distribution at given $T$
- Calculate $r_i(\delta t) = r_i(0) + \delta t v_i(0)$
- Using new $r_i(\delta t)$ evaluate the total potential $V_i$ on atom $I$
- Calculate negative gradient of $V_i$ to find $a_i(\delta t) = -\nabla V_i / m_i$
- Start Verlet algorithm using $r_i(0)$, $r_i(\delta t)$ and $a_i(\delta t)$
- Repeat for all atoms (including solvent, if any)
- Repeat the last three steps for $\sim 10^6$ successive times (MD steps)
Limitations of MD simulations

- Full atomic representation → noise
- Empirical force fields → limited by the accuracy of the potentials
- Time steps constrained by the fastest motion (bond stretching of the order of femptoseconds)
- Inefficient sampling of the complete space of conformations
- Limited to small proteins (100s of residues) and short times (subnanoseconds)
MC simulations of protein dynamics

- Unfolding dynamics
- Folding dynamics
- Binding dynamics
Folding/unfolding energy landscapes

Thermodynamic equilibrium → native state has lowest free energy

Reference
Methodology for unfolding dynamics

- Start with known structure
- Randomly perturb all virtual bond torsional angles (softest degrees of freedom)
- Complete one cycle (i.e. change all virtual bonds)
- Compute the new conformation
- Evaluate its conformational energy
- Accept/reject the new conformation on the basis of Metropolis criterion, i.e. accept if exp (-βΔE) > A, where A is a random number. 0 < A < 1. Readjust Metropolis criteria parameter β so as to accept about ½ of the moves.
- Save trajectory for further analysis
Methods of trajectory analysis:

- Normal mode analysis (NMA)
- Essential dynamics analysis (EDA) (Amadei et al., 1993).

“Principal component analysis (PCA)” (*)

(diagonalization of the covariance matrix of atomic fluctuations to yield the collective variables that are sorted according to their contribution to the total ms fluctuations)


**Covariance matrix**

(directly found from MD or MC trajectories)

\[ \Delta R_i = \text{instantaneous fluctuation in the position vector } R_i \text{ of atom } i = R_i - \langle R_i \rangle \]

\[ \langle \Delta R_1 . \Delta R_1 \rangle = \text{ms fluctuation of site 1 averaged over all snapshots.} \]
Eigenvalue decomposition of \( C \)

\[
C = U \Lambda U^{-1}
\]

\( U \) is the matrix of eigenvectors, \( \Lambda \) is the diagonal matrix of eigenvalues. The \( i \)th column (eigenvector) of \( U \) is given by a linear combination of Cartesian coordinates and represents the axis of the \( i \)th collective coordinate (principal axis) in the conformational space.

The \( i \)th eigenvalue represents the mean-square fluctuation along the \( i \)th principal axis. The motion along the \( i \)th principal axis is the \( i \)th mode.
In NMA, the covariance (also called second moment) matrix is given by

\[ C = k_B T H^{-1} \]

Where \( k_B \) is the Boltzmann constant, \( T \) is the absolute temperature and \( H \) is the (Hessian) matrix of the second derivatives of the potential.

**GNM** is a simplified form of PCA in which \( H \) reduces to \( \Gamma \).
**Aim:** to understand the long-time dynamics, to remove the ‘uninteresting’ fast modes.

**Method:** to map the trajectory onto a new multidimensional space, the axes of which refer to motions along principal coordinates.

**Frame transformation:** From the 3N-dimensional space defining ‘conformations’ in Cartesian coordinates to the 3N-6 dimensional space of conformations in collective coordinates.
Essential dynamics analysis (EDA)

- Based on singular value decomposition of trajectory matrix \( A \)
- More effective on a coarse-grained scale
- Snapshots provided by MD trajectories, or different crystal structures
- Provide information on collective dynamics, and in particular the low frequency modes

Ref: Amadei, Linssen & Berendsen, Proteins 17, 412, 1993
Consider a molecule composed of 19 interaction sites. We define a conformation vector for each step $i$, composed of the instantaneous position vectors of the 19 sites relative to their mean positions:

$$\Delta r_i = \begin{bmatrix} r_{1i} - \bar{r}_1 \\ r_{2i} - \bar{r}_2 \\ \vdots \\ r_{19i} - \bar{r}_{19} \end{bmatrix}$$

Suppose the trajectory is comprised of $n$ steps.

Conformation vectors are organized in a trajectory matrix $A$ of size $19 \times n$ as:

$$A_{[19 \times n]} = [ \Delta r_1 \ \Delta r_2 \ \Delta r_3 \ \ldots \ \Delta r_n ]$$

$$= U_{[19 \times 19]} \Lambda_{[19 \times 19]} V^T_{[19 \times n]}$$

$U$ is the matrix of the left singular vectors (or principal axes) $(u_i)$ of $A$, $\Lambda$ is the diagonal matrix of the singular values $\lambda_i$. The columns of $\Lambda V^T$ are the conformation vectors expressed in the singular frame spanned by the axes $u_i$, $1 \leq i \leq 19$. Thus, the matrix $U^T$ when operated on $A$, rotates each of the $n$ original conformation vectors into their representation in the new space.
SVD method for genome-wide expression data processing & modeling

Organized in a matrix $e$ of size $[8,000 \times 10]$ to be decomposed as

$$e = u \varepsilon \, v^T$$

(Matrix to be decomposed)

$e_{i,8,000}$

genome-wide expression of the same sample $i$, expressed in the singular space

$\alpha_i = \begin{bmatrix} \alpha_{i1} \\ \alpha_{i2} \\ \alpha_{i3} \\ \alpha_{i4} \\ \alpha_{i5} \\ \alpha_{i,8,000} \end{bmatrix}$

$\alpha_i$ = ith column of $V^T$

$mRNA$ levels for all genes for the $i$th sample
(e.g. a 8800-dimensional vector for each brain tissue)

$M = 10$ arrays
$N = 8,000$ genes
Original A matrix for the time evolution of 3N coordinates

\[
A = \begin{bmatrix}
  r_{11} & r_{21} & r_{31} & r_{M1} \\
  r_{12} & r_{22} & r_{32} & r_{M2} \\
  \text{boxed} & r_{23} & r_{33} & r_{M3} \\
  r_{14} & r_{24} & r_{34} & r_{M4} \\
  r_{15} & r_{25} & r_{35} & r_{M5} \\
  r_{1,N} & r_{2,N} & r_{3,N} & r_{M,N}
\end{bmatrix}
\]

3N coordinates define the multidimensional conformational space (M = # steps)

Snapshot at a given time step

Complete trajectory of the 3\textsuperscript{rd} residue (or the time evolution along the 3\textsuperscript{rd} coordinate)
Conformation in SVD space

<table>
<thead>
<tr>
<th>$\alpha_{11}$</th>
<th>$\alpha_{21}$</th>
<th>$\alpha_{31}$</th>
<th>Eigenvector $\gamma_1$</th>
<th>$\alpha_{M1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_{12}$</td>
<td>$\alpha_{22}$</td>
<td>$\alpha_{32}$</td>
<td>eigenvector $\gamma_2$</td>
<td>$\alpha_{M2}$</td>
</tr>
<tr>
<td>$\alpha_{13}$</td>
<td>$\alpha_{23}$</td>
<td>$\alpha_{33}$</td>
<td></td>
<td>$\alpha_{M3}$</td>
</tr>
<tr>
<td>$\alpha_{14}$</td>
<td>$\alpha_{24}$</td>
<td>$\alpha_{34}$</td>
<td></td>
<td>$\alpha_{M4}$</td>
</tr>
<tr>
<td>$\alpha_{15}$</td>
<td>$\alpha_{25}$</td>
<td>$\alpha_{35}$</td>
<td></td>
<td>$\alpha_{M5}$</td>
</tr>
</tbody>
</table>

$V^T = \begin{bmatrix}
\alpha_{1,1} & \alpha_{2,1} & \alpha_{3,1} & \cdots & \alpha_{M,1} \\
\alpha_{1,2} & \alpha_{2,2} & \alpha_{3,2} & \cdots & \alpha_{M,2} \\
\alpha_{1,3} & \alpha_{2,3} & \alpha_{3,3} & \cdots & \alpha_{M,3} \\
\alpha_{1,4} & \alpha_{2,4} & \alpha_{3,4} & \cdots & \alpha_{M,4} \\
\alpha_{1,5} & \alpha_{2,5} & \alpha_{3,5} & \cdots & \alpha_{M,5} \\
\end{bmatrix}$

First row: displacement along the first PA (a total of $M$ steps)

Snapshot at a given time step
Projection of the motion onto the space of the two first principal axes

Comparison with essential modes from MD

Doruker, Atilgan & Bahar, Proteins 40, 512, 2000
The motions driven by the essential modes are

- Robust (invariant; do not depend on the details of the model)
- Relevant to biological function
General strategy:

- Decompose the trajectory into a collection of modes
- Reconstruct the trajectory based on dominant modes
Proteins are microscopic machines

Two fundamental features in modeling/designing these machines

- Mechanics
- Chemical properties
Gaussian Network Model

Global motions of HIV-1 RT
Computational problems/challenges

- Trade off between computational efficiency and accuracy
- Loss of information at atomic scale for acquiring information at the global scale
- Multiscale representations of dynamics
- Extracting rules from microscopic simulations, for macroscopic approaches
- Bridging continuous and discrete models