Final presentation

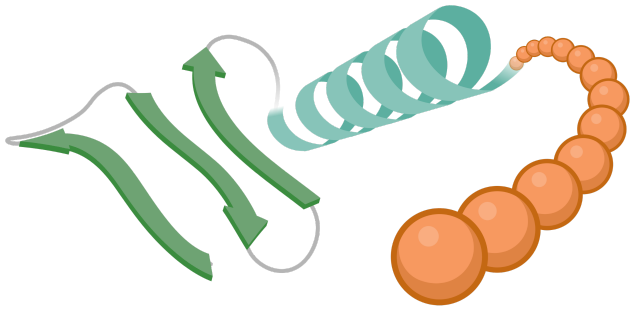
Research on the mechanism of protein folding dynamics

Reporter: **Zhenyu Wei**

Mentor: **Yunfei Chen**

School of mechanical engineering

Southeast University



Content

- Research background
- Multi-scale modeling
- Enhanced sampling
- Free energy of ion interaction in solution
- Free energy matrix
- OpenPD
- Future Proposal

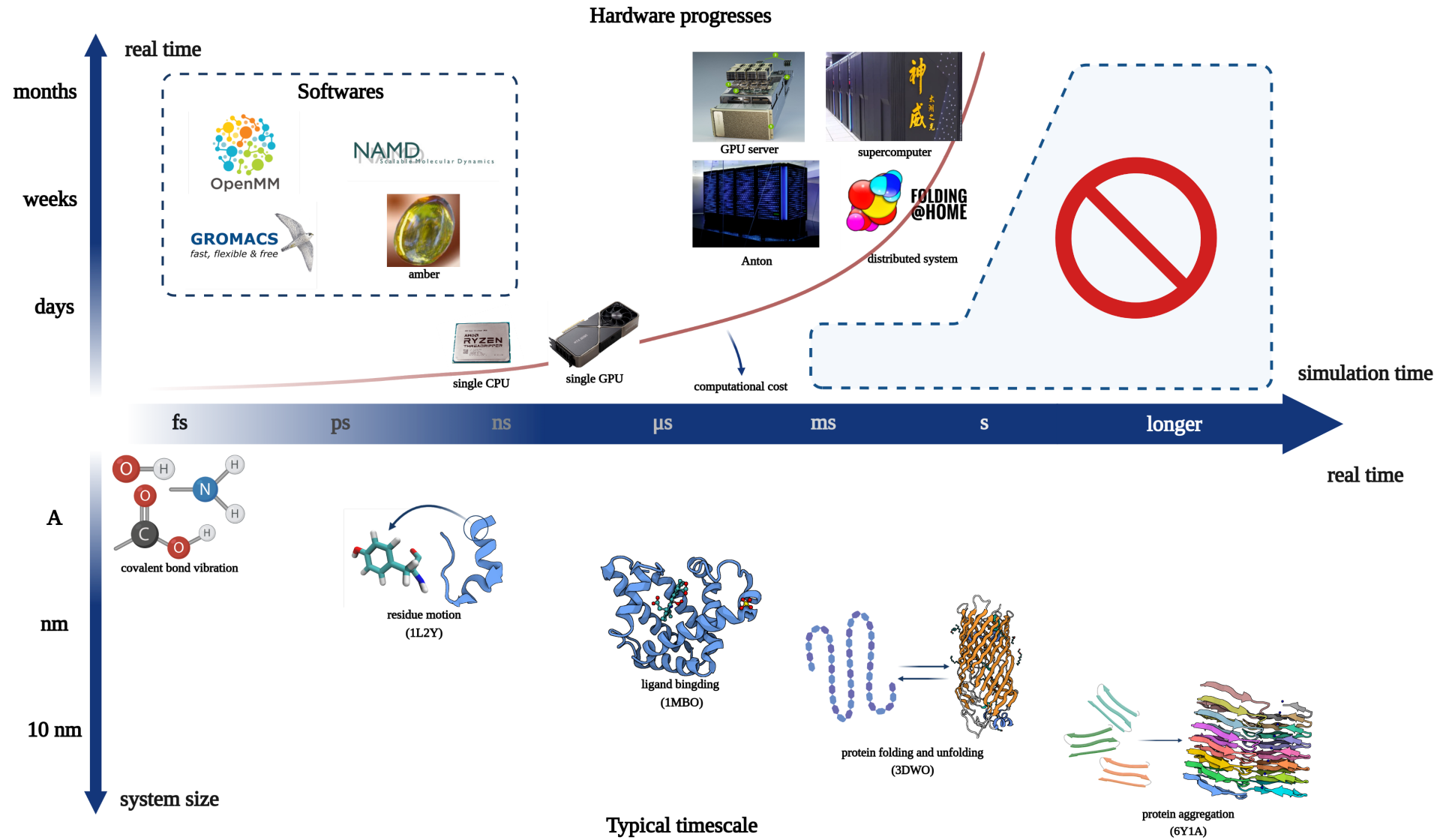
Research background

Importance of protein folding and misfolding

- Protein is essential in a wide range of fields.
  - **Membrane protein** paly a vital role in the inter-cellular substance exchange and communication
  - **Protein enzyme** is the most efficient catalyst, maintaining gazillion of chemical reaction in live system
- Protein misfolding leads to many diseases:
  - Creutzfeldt-Jakob disease (CJD)
  - Type 2 diabetes
  - Many neurodegenerative diseases

- The **sequence** of peptide chain determine the final structure of protein **completely**
- The protein can fold to the **native structure** rapidly and precisely
- Levinthal's paradox[1]:

> The protein with 100 peptides may misfold into a maximum of 3^{198} different conformations. Therefore, it would require a time longer than the age of the universe to arrive at its **correct native conformation** if protein **random searched** all the possible conformation.


Hardware progresses
real time
months
weeks
days
Softwares
OpenMM
NAMD
Scalable Molecular Dynamics
GROMACS
fast, flexible & free
amber
GPU server
supercomputer
Anton
FOLDING @HOME
distributed system
single CPU
single GPU
computational cost
simulation time
fs
ps
ns
μs
ms
s
longer
real time
A
nm
10 nm
system size
covalent bond vibration
residue motion
(1L2Y)
ligand bingding
(1MBO)
protein folding and unfolding
(3DWO)
protein aggregation
(6Y1A)
Typical timescale

Multi-scale modeling

Fundamentals

- When we sample in NVT ensemble, the probability density follows the **Boltzmann distribution**:

$$p\left(\mathbf{q}_a, \mathbf{q}_{a-}\right) = \frac{\exp\left[-\frac{U\left(\mathbf{q}_a, \mathbf{q}_{a-}\right)}{k_B T}\right]}{\int_\Omega \prod_{i=1}^{3N_a} \mathrm{d}q_a^{(i)} \prod_{j=1}^{3N_{a-}} \mathrm{d}q_{a-}^{(j)} \exp\left[-\frac{U\left(\mathbf{q}_a, \mathbf{q}_{a-}\right)}{k_B T}\right]}$$

- We can write **experimental observations** in the form of ensemble average:

$$\langle A \rangle = \int_\Omega \prod_{i=1}^{3N_a} \mathrm{d}q_a^{(i)} \prod_{j=1}^{3N_{a-}} \mathrm{d}q_{a-}^{(j)} A\left(\mathbf{q}_a\right) p\left(\mathbf{q}_a, \mathbf{q}_{a-}\right) = \int_{\Omega_a} \prod_{i=1}^{3N_a} \mathrm{d}q_a^{(i)} A\left(\mathbf{q}_a\right) \int_{\Omega_{a-}} \prod_{j=1}^{3N_{a-}} \mathrm{d}q_{a-}^{(j)} p\left(\mathbf{q}_a, \mathbf{q}_{a-}\right)$$

- Now we define a **marginal** probability density:

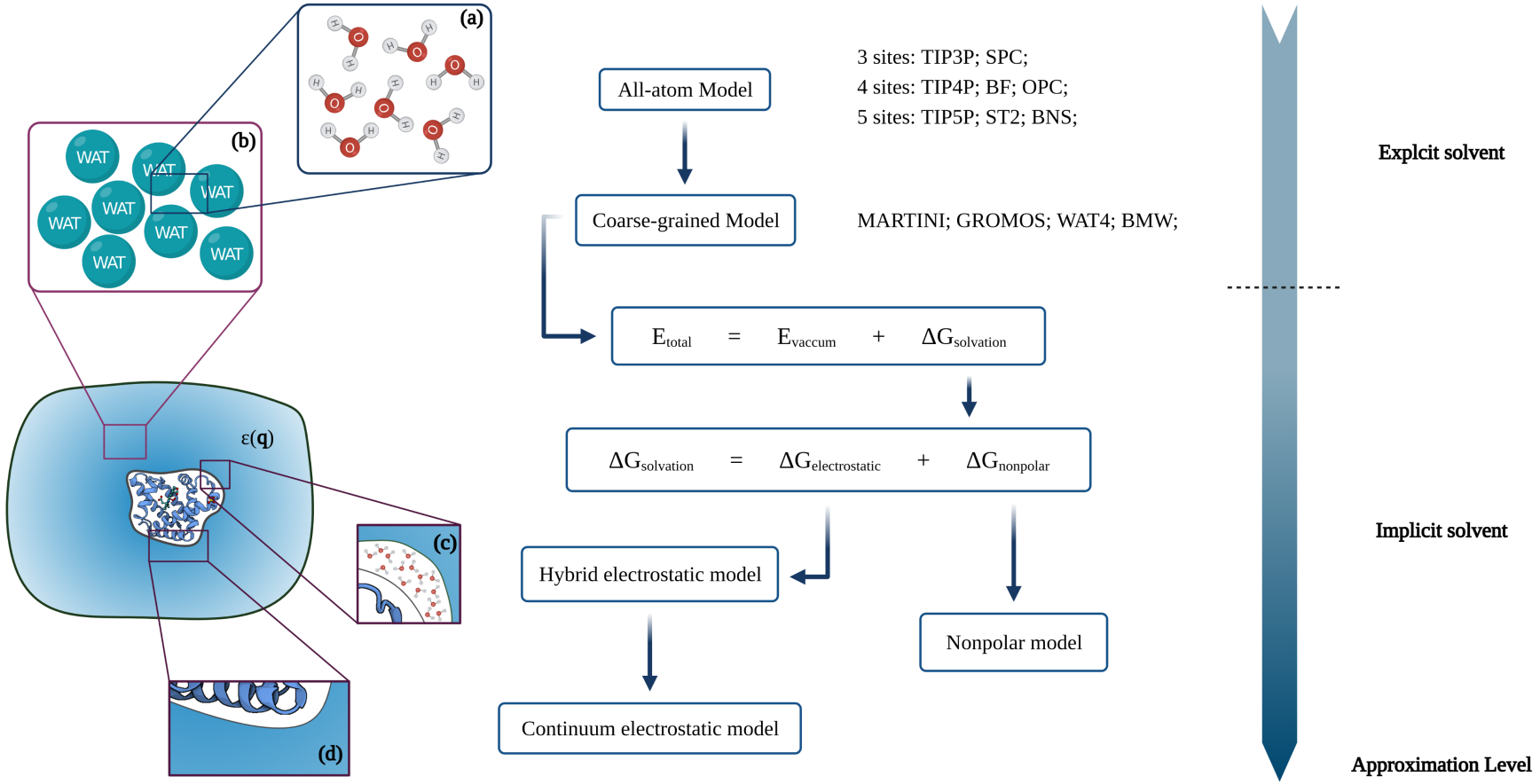
$$\bar{p}\left(\mathbf{q}_a\right)=\frac{\int_{\Omega_{a-}}\prod_{i=1}^{3N_{a-}}\mathrm{d}q_{a-}^{(i)}\exp\left[-\frac{U\left(\mathbf{q}_a,\mathbf{q}_{a-}\right)}{k_BT}\right]}{\int_{\Omega}\prod_{i=1}^{3N_a}\mathrm{d}q_a^{(i)}\prod_{j=1}^{3N_{a-}}\mathrm{d}q_{a-}^{(j)}\exp\left[-\frac{U\left(\mathbf{q}_a,\mathbf{q}_{a-}\right)}{k_BT}\right]}=\frac{\exp\left[-\frac{F\left(\mathbf{q}_a\right)}{k_BT}\right]}{\int_{\Omega_a}\prod_{i=1}^{3N_a}\mathrm{d}q_a^{(i)}\exp\left[-\frac{F\left(\mathbf{q}_a\right)}{k_BT}\right]}$$

- Then we can rewrite the ensemble average as:

$$\left\langle A\right\rangle=\int_{\Omega_a}\prod_{i=1}^{3N_a}\mathrm{d}q_a^{(i)}A\left(\mathbf{q}_a\right)\bar{p}\left(\mathbf{q}_a\right)$$

- , where the DoFs of $a-$ have been integrated out.

Implicit solvent model



- Implicit solvent model treat solvent as a **continuum dielectrics**.

Electrostatic background

- Consider a electrostatic potential ϕ:

$$\Delta G_{\mathrm{ele}} = q\left[\phi - \phi_{\mathrm{vac}}\right]$$

- Combine the definition of ϕ and Gauss Law, we give the **Poisson equation**:

$$\nabla \cdot \left[\varepsilon_r \nabla \phi\right] = -\frac{\rho}{\varepsilon_0}$$

- And its special case for spherically symmetric system:

$$\frac{1}{r^2}\frac{\mathrm{d}}{\mathrm{d}r}\left[r^2\frac{\mathrm{d}\phi}{\mathrm{d}r}\right] = -\frac{\rho(r)}{\varepsilon_r\varepsilon_0}$$

- Consider mean-field theory and Debye-Huckle theory:

$$\frac{1}{r^2}\frac{\mathrm{d}}{\mathrm{d}r}\left[r^2\frac{\mathrm{d}\phi}{\mathrm{d}r}\right]=-\frac{e_0}{\varepsilon_r\varepsilon_0}\sum_i^{N_i}z_ic_i^0\exp\left[-\frac{z_ie_0\phi(r)}{k_BT}\right]$$

- This equation is refereed to as **Poisson-Boltzmann Equation**. In the limit of $k_BT\gg z_ie_0\phi(r)$, we have:

$$\frac{1}{r^2}\frac{\mathrm{d}}{\mathrm{d}r}\left[r^2\frac{\mathrm{d}\phi}{\mathrm{d}r}\right]=-\frac{e_0}{\varepsilon_r\varepsilon_0}\sum_i^{N_i}z_ic_i^0\left[1-\frac{z_ie_0\phi(r)}{k_BT}\right]=-\left[\frac{e_0^2}{\varepsilon_r\varepsilon_0k_BT}\sum_i^{N_i}z_i^2c_i^0\right]\phi(r)$$

- , which is called **Linearized Poisson-Boltzmann Equation**.

Generalized Born Model

- Still et al[1] invent Generalized Born Model follows:

$$G_{\mathrm{ele}} = \frac{1}{2\varepsilon} \sum_{i=1}^{N_c} \sum_{j=1, j\neq i}^{N_c} \frac{q_i q_j}{r_{ij}} - \frac{1}{2}\left(1 - \frac{1}{\varepsilon}\right) \sum_{i=1}^{N_c} \frac{q_i^2}{R_i}$$

- This equation can be rearranged as:

$$G_{\mathrm{ele}} = \frac{1}{2} \sum_{i=1}^{N_c} \sum_{j=1, j\neq i}^{N_c} \frac{q_i q_j}{r_{ij}} - \frac{1}{2}\left(1 - \frac{1}{\varepsilon}\right) \sum_{i=1}^{N_c} \sum_{j=1, j\neq i}^{N_c} \frac{q_i q_j}{r_{ij}} - \frac{1}{2}\left(1 - \frac{1}{\varepsilon}\right) \sum_{i=1}^{N_c} \frac{q_i^2}{R_i}$$

[1]: W Clark Still, Anna Tempczyk, Ronald C Hawley, and Thomas Hendrickson. Semianalytical treatment of solvation for molecular

mechanics anddynamics.Journal of the American Chemical Society, 112(16):6127–6129, 1990

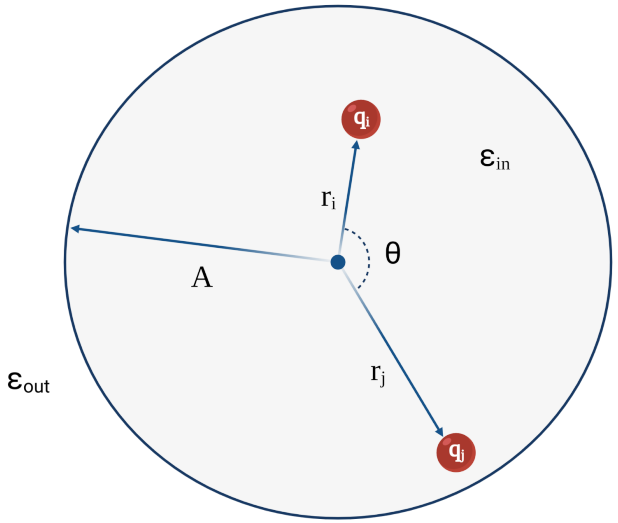
- As $\Delta G_{\mathrm{ele}}=q\left[\phi-\phi_{\mathrm{vac}}\right]$, we have:

$$\Delta G_{\mathrm{ele}}=-\frac{1}{2}\left(1-\frac{1}{\varepsilon}\right)\sum_{i=1}^{N_c}\sum_{j=1}^{N_c}\frac{q_i q_j}{f_{GB}}$$

- , where f_{GB} stands for the combination of last two terms. As this expression is similar to the Born energy, the model is refereed to as **generalized Born model**. And Still states that:

$$f_{GB}=\sqrt{r_{ij}^2+R_i R_j \exp\left[-\frac{r_{ij}^2}{\gamma R_i R_j}\right]}$$

- Consider a **two dielectric model** shown below:



- This model is spherically symmetric, describing several charges **embedded** in a dielectric with **low permittivity** ε_{in} which is surrounded by a different dielectric with **higher permittivity** ε_{out}

- For such a discrete distributed charge system, we can rewrite **Poisson equation** with **Green function**:

$$\nabla^2 \mathbf{G}\left(\mathbf{r}_i, \mathbf{r}_j\right) = -\frac{\delta\left(\mathbf{r}_i - \mathbf{r}_j\right)}{\varepsilon_{\text{in}}\varepsilon_0}$$

- , solution of which has the form:

$$\mathbf{G}\left(\mathbf{r}_i, \mathbf{r}_j\right) = \frac{1}{4\pi\varepsilon_{\text{in}}\varepsilon_0}\frac{1}{||\mathbf{r}_i - \mathbf{r}_j||} + \mathbf{F}\left(\mathbf{r}_i, \mathbf{r}_j\right)$$

- Obviously, $\mathbf{F}\left(\mathbf{r}_i, \mathbf{r}_j\right)$ stands for the complex solute-solvent interaction:

$$\Delta G_{\text{ele}} = \frac{1}{2}\sum_{i=1}^{N_c}\sum_{j=1}^{N_c}\mathbf{F}\left(\mathbf{r}_i, \mathbf{r}_j\right) q_i q_j$$

- Analytical solution of Poisson Equation for this model exists[1]:

$$\mathbf{F}\left(\mathbf{r}_i, \mathbf{r}_i\right) = -\frac{1}{A}\left(\frac{1}{\varepsilon_{\text{in}}} - \frac{1}{\varepsilon_{\text{out}}}\right)\sum_{n=0}^{\infty}\frac{t_{ii}^{n}}{1+\frac{n}{n+1}\beta}$$

$$\mathbf{F}\left(\mathbf{r}_i, \mathbf{r}_j\right) = -\frac{1}{A}\left(\frac{1}{\varepsilon_{\text{in}}} - \frac{1}{\varepsilon_{\text{out}}}\right)\sum_{n=0}^{\infty}\frac{t_{ij}P_n(cos\theta)}{1+\frac{n}{n+1}\beta}$$

- , where $t_{ij} = r_i r_j / A^2$, $\beta = \varepsilon_{in}/\varepsilon_{out}$, and $P_n(cos\theta)$ are associated Legendre functions.

[1]: John G Kirkwood. Theory of Solutions of Molecules Containing Widely Separated Charges with Special Application to Zwitterions. The Journal of Chemical Physics, 2(7):351–361, 1934.

- First we consider the solution of $\mathbf{F}\left(\mathbf{r}_i, \mathbf{r}_i\right)$. In the limit of $\beta \to 0$:

$$\Delta G_{\mathrm{ele}}^{ii} = \frac{q_i^2}{2}\mathbf{F}\left(\mathbf{r}_i, \mathbf{r}_i\right) = -\frac{1}{2}\left(\frac{1}{\varepsilon_{\mathrm{in}}} - \frac{1}{\varepsilon_{\mathrm{out}}}\right)\frac{q_i^2}{A(1-t_{ii})} = -\frac{1}{2}\left(\frac{1}{\varepsilon_{\mathrm{in}}} - \frac{1}{\varepsilon_{\mathrm{out}}}\right)\frac{q_i^2}{A - \frac{r_i^2}{A}}$$

- As the final result is similar to the Born energy, we name the denominator $R_i = A - r_i^2/A$ after **effective Born radius**.

- Next consider $\mathbf{F}\left(\mathbf{r}_i, \mathbf{r}_j\right)$, we first replace $n/(n+1)$ with a constant α, for all $n \geq 1$, rewriting the series as:

$$1+\frac{1}{1+\alpha\beta}-\frac{1}{1+\alpha\beta}+\sum_{n=1}^{\infty}\frac{t_{ij}^{n}P_n(cos\theta)}{1+\alpha\beta}=\frac{1}{1+\alpha\beta}\left[\sum_{n=0}^{\infty}\left[\frac{t_{ij}^{n}P_n(cos\theta)}{1+\alpha\beta}\right]+\alpha\beta\right]$$

- , where $\sum_{n=0}^{\infty} t^n P_n(cos\theta)$ can be simplified as $1/\sqrt{1-2t_{ij}cos\theta+t_{ij}^2}$:

$$\Delta G_{\mathrm{ele}}^{ij}=\frac{q_i q_j}{2}\mathbf{F}\left(\mathbf{r}_i, \mathbf{r}_i\right)=-\frac{1}{2}\left(\frac{1}{\varepsilon_{\mathrm{in}}}-\frac{1}{\varepsilon_{\mathrm{out}}}\right)\frac{q_i q_j}{1+\alpha\beta}\left[\frac{1}{\sqrt{r_{ij}^2+R_i R_j}}+\frac{\alpha\beta}{A}\right]$$

- , which yields $f_{GB}=\sqrt{r_{ij}^2+R_i R_j}$

- Now we need to estimate the effective Born radius. In classical electrostatic:

$$G_{\mathrm{ele}} = \frac{1}{2}\int_{\Omega} \rho\phi dV = \frac{1}{8\pi}\int_{\Omega} \mathbf{E}\cdot\mathbf{D} dV$$

- Then we introduce a essential approximation, which state $\mathbf{D}$ has the same form as Coulomb field $\mathbf{D}_i \approx \frac{q_i}{r^2}\hat{\mathbf{r}}_i$:

$$G^{ii}_{\mathrm{ele}} = \frac{1}{8\pi}\left[\int_{\Omega_{\mathrm{in}}} \frac{q_i q_i}{\varepsilon_{\mathrm{in}} r^4} dV + \int_{\Omega_{\mathrm{out}}} \frac{q_i q_i}{\varepsilon_{\mathrm{out}} r^4} dV\right]$$

$$\Delta G^{ii}_{\mathrm{ele}} = G_{ii} - \frac{1}{8\pi}\int_{\Omega} \frac{q_i^2}{\varepsilon_{\mathrm{in}} r^4} dV = -\frac{1}{8\pi}\left(\frac{1}{\varepsilon_{\mathrm{in}}} - \frac{1}{\varepsilon_{\mathrm{out}}}\right)\int_{\Omega_{\mathrm{out}}} \frac{q_i^2}{r^4} dV$$

- Recall the expression of $\Delta G_{\text{ele}}^{ii}$, we have:

$$\frac{1}{R_i} = \frac{1}{4\pi} \int_{\Omega_{\text{out}}} \frac{1}{r^4} dV$$

- As the boundary of Ω_{in} is usually complex, we rewrite $\Delta G_{\text{ele}}^{ii}$ as:

$$\frac{1}{R_i} = \frac{1}{4\pi} \left[\int_0^k \frac{1}{r^2} 4\pi r^2 dr - \int_{\Omega_{\text{in}} - \Omega_s} \frac{1}{r^4} dV \right] = \frac{1}{k} - \frac{1}{4\pi} \int_{\Omega_{\text{in}} - \Omega_s} \frac{1}{r^4} dV$$

- , where Ω_s is a sphere volume with radius k.

- Finally we can consider the influence of mobile ion based on the Debye-Huckle theory by replacement:

$$\left(\frac{1}{\varepsilon_{\text{in}}}-\frac{1}{\varepsilon_{\text{out}}}\right) \rightarrow \left(\frac{1}{\varepsilon_{\text{in}}}-\frac{\exp\left[-\kappa f_{GB}\right]}{\varepsilon_{\text{out}}}\right)$$

- , where κ is the Debye screening radius:

$$\kappa=\frac{e_0^2}{\varepsilon_r \varepsilon_0 k_B T} \sum_i^{N_i} z_i^2 c_i^0$$

Enhanced sampling

Fundamentals

- Consider a mapping:

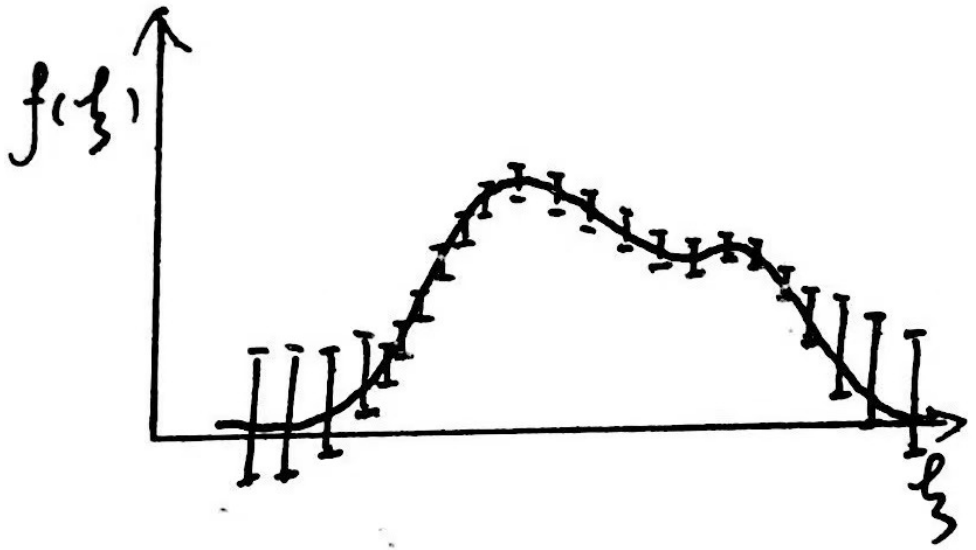
$$\theta : \Omega \in \mathbb{R}^{3N} \rightarrow \Theta \in \mathbb{R}^{n}, n \ll 3N$$

- Then we can analyze free energy based on the distribution of θ:

$$F\left(\theta'\right) = -k_B T \ln\left[H\left(\theta'\right)\right]$$

- Advantages:
  - Sampling as a $3N - n$ dimensional hyper-sphere with high efficiency;
  - It's easier to manipulate CV, overcoming the energy barrier;
  - It's easier to compare to the experimental observations;

Umbrella sampling



- In simulation, PDF can be derived from histogram on collective variables. However, the estimation on the **rare events** is accompanied with **high statistical error**

- Simulation with bias potential:

$$\tilde{U}(\mathbf{R}) = U(\mathbf{R}) + \Delta U(\mathbf{R})$$

  where three terms corresponds to new potential, origin potential and bias potential, respectively.

- So we have:

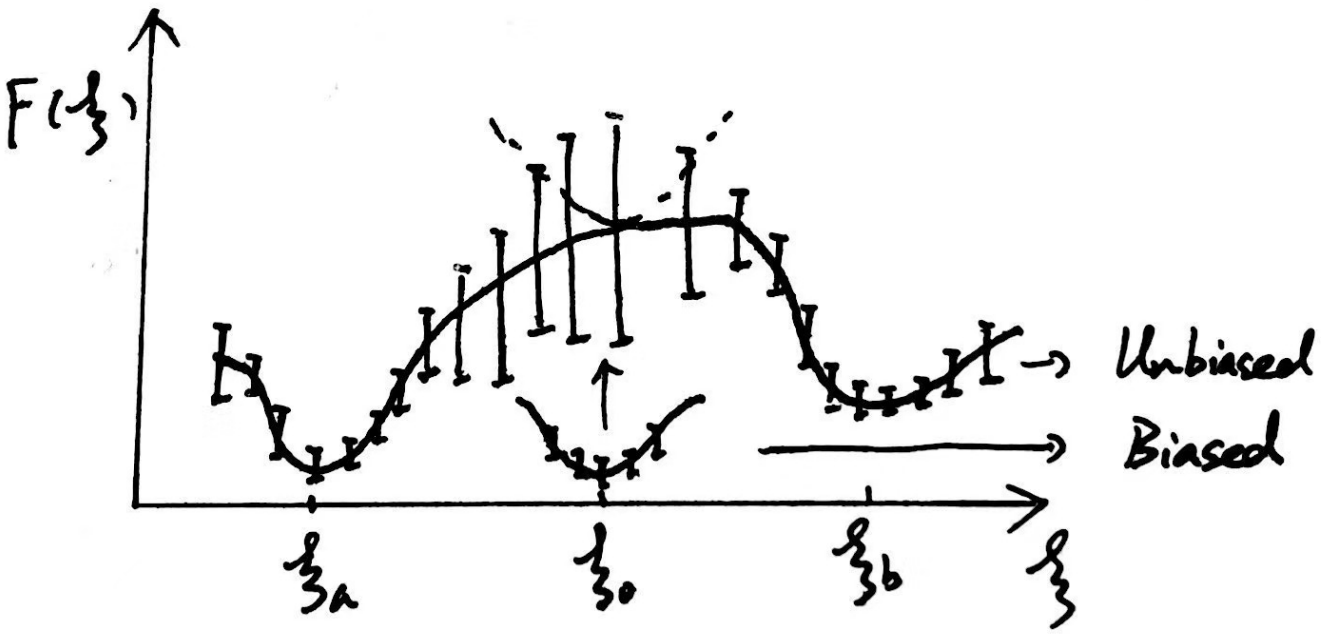
$$\begin{aligned} \tilde{F}(\xi') &= -k_B T ln \left(\frac{1}{Z} \int d^{3N} R e^{-\beta U(\mathbf{R})} e^{-\beta \Delta U(\mathbf{R})} \delta \left(\xi(\mathbf{R}) - \xi' \right) \right) \\ &= F(\xi') + \Delta U(\xi') + C \end{aligned}$$

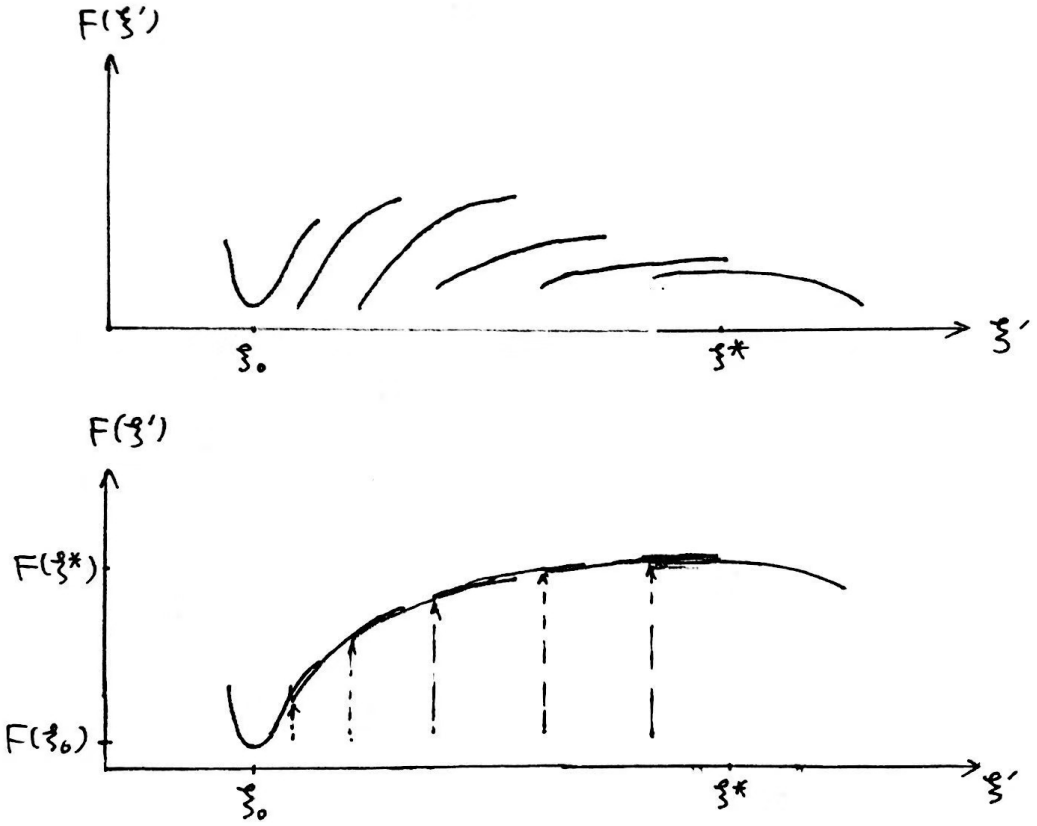
- Usually, a potential like harmonic oscillator is a good choice:

$$\Delta U(\mathbf{R}) = \frac{1}{2}\,\mathrm{K}_{bias}\;\left(\xi(\mathbf{R}) - \xi_0\right)^2$$

  where K_{bias} and ξ_0 are two hyper parameters for umbrella sampling

-





- Suppose we perform a series of simulation over different bias potential
- Each histogram is overlapped with others (like umbrella)
- We can reconstruct the free energy with a series curve with little error

Weighted Histogram Analysis Method

- Initial data:
  - A collective variable $\xi : \Omega \to \Xi \in \mathbb{R}^d, d \ll 6N$
  - M independent trajectories (usually with a biased potential), $S_i, i \in [1, M]$
  - The histogram and probability distribution (or probability density) of each trajectories on reaction coordinate: $h_i(\xi')$ and $p_i(\xi'),\ i \in [1, M]$
- Final goal:
  - The probability distribution (or probability density) on reaction coordinate of unbiased simulation

- **Initial Assumption**: the biased probability distribution $p_i(\xi')$ is related to the unbiased one $p_0(\xi')$:

$$p_i(\xi') = f_i c_i(\xi') p_0(\xi')$$

where $c_i(\xi')$ is the biasing factor corresponds to the influence of bias potential:

  - $c_i(\xi') = exp\left[-(\beta_i - \beta_0)E_0(\xi')\right]$ for temperature biasing
  - $c_i(\xi') = exp\left[-\beta \Delta U(\xi')\right]$ for coordinate biasing

- And f_i is a normalization factor:

$$
\begin{aligned}
f_i^{-1} &= \int_{\Xi} c_i(\xi') p_0(\xi')\, d\xi' \\
&= \int_{\Xi} e^{-\beta \Delta U(\xi')} \frac{1}{Q_0} \int_{\Omega} e^{-\beta U(\mathbf{q})} \delta\left[\xi(\mathbf{q}) - \xi\right] d\mathbf{q} d\xi \\
&= \frac{1}{Q_0} \int_{\Xi} \int_{\Omega} e^{-\beta [U(\mathbf{q}) + \Delta U(\mathbf{q})]} \delta\left[\xi(\mathbf{q}) - \xi\right] d\mathbf{q} d\xi \\
&= \frac{1}{Q_0} \int_{\Omega} e^{-\beta [U(\mathbf{q}) + \Delta U(\mathbf{q})]} d\mathbf{q} \; = \frac{Q_i}{Q_0}
\end{aligned}
$$

- As we know:

$$\Delta A_{ij} = -k_B T ln \frac{Q_j}{Q_i} = -k_B T ln \frac{\int_{\Gamma_j} exp\left[-\frac{U_j(\mathbf{q})}{k_B T}\right] d\mathbf{q}}{\int_{\Gamma_i} exp\left[-\frac{U_i(\mathbf{q})}{k_B T}\right] d\mathbf{q}}$$

- So we have:

$$\beta^{-1} ln f_i = \Delta A_{0i}$$

- Based on the initial assumption:

$$p_i(\xi') = f_i c_i(\xi') p_0(\xi')$$

- We can define the estimation of unbiased result from i trajectory:

$$p_0^{(i)}(\xi') = \frac{p_i(\xi)}{f_i c_i(\xi')} = \frac{h(\xi')}{N_i f_i c_i(\xi')}$$

  where $h(\xi')$ is the histogram value and N_i is the total samples of trajectory i.

- And we give the final estimation of $p_0(\xi')$ as a weighted average:

$$p_0^{est}(\xi') = \sum_{i=1}^{M} w_i p_0^{(i)}(\xi')$$

- Obviously, w_i satisfy the normalization condition: $\sum_{i=1}^{M} w_i = 1$.
- Clearly, if we want to get a good estimation, we want the variance of each data point to be the minimum:

$$
\begin{aligned}
var(p_0^{est}(\xi')) &= \left\langle (p_0^{est}(\xi') - \langle p_0^{est}(\xi') \rangle)^2 \right\rangle \\
&= \left\langle \left(\sum_{i=1}^{M} w_i p_0^{(i)}(\xi') - \left\langle \sum_{i=1}^{M} w_i p_0^{(i)}(\xi') \right\rangle \right)^2 \right\rangle \\
&= \left\langle \left(\sum_{i=1}^{M} w_i \left(p_0^{(i)}(\xi') - \left\langle p_0^{(i)}(\xi') \right\rangle \right) \right)^2 \right\rangle
\end{aligned}
$$

We define $\delta p_0^{(i)}(\xi') = \left(p_0^{(i)}(\xi') - \left\langle p_0^{(i)}(\xi') \right\rangle \right)$. Then we have:

$$
\begin{aligned}
var\left(p_0^{est}(\xi')\right) &= \left\langle \left(\sum_{i=1}^{M} w_i \delta p_0^{(i)}(\xi') \right)^2 \right\rangle \\
&= \left\langle \sum_{i=1}^{M} w_i^2 (\delta p_0^{(i)}(\xi'))^2 + \sum_{j=1}^{M} \sum_{k!=j}^{M} w_j w_k \delta p_0^{(j)}(\xi') \cdot \delta p_0^{(k)}(\xi') \right\rangle \\
&= \sum_{i=1}^{M} w_i^2 \left\langle \left(\delta p_0^{(i)}(\xi') \right)^2 \right\rangle + \sum_{j=1}^{M} \sum_{k!=j}^{M} w_j w_k \left\langle \delta p_0^{(j)}(\xi') \cdot \delta p_0^{(k)}(\xi') \right\rangle
\end{aligned}
$$

- Always, we can make the assumption of independent sampled trajectory, which means $\left\langle \delta p_0^{(j)}(\xi') \cdot \delta p_0^{(k)}(\xi') \right\rangle = 0$ and we have $\left\langle \left(\delta p_0^{(i)}(\xi') \right)^2 \right\rangle = var\left(p_0^{(i)}(\xi') \right)$, so we can rewrite equation as:

$$var\left(p_0^{est}(\xi') \right) = \sum_{i=1}^{M} w_i^2 var\left(p_0^{i}(\xi') \right)$$

- As we know $var(ax) = \left\langle a^2 x^2 \right\rangle - \left\langle ax \right\rangle^2 = a^2 var(x)$, so recall the definition of $p_0^{(i)}(\xi') = \dfrac{p_i(\xi)}{f_i c_i(\xi')} = \dfrac{h(\xi')}{N_i f_i c_i(\xi')}$, we have:

$$var\left(p_0^{est}(\xi') \right) = \sum_{i=1}^{M} \frac{w_i^2 var\left(h(\xi) \right)^2}{N_i^2 c_i^2(\xi) f_i^2}$$

- If we have N independent samples, the probability and variance of having n counts in one bin will follow the **binomial distribution**
  - Mean: $np_i(\xi')$
  - Variance: $np_i(\xi')(1-p_i(\xi'))$
- In the limit of large N and small $p_i(\xi')$ (exactly what happened for a long simulation and a detailed histogram), the binomial distribution can be treated as a **Poisson distribution**:

$$P(n) = exp(-Np_i(\xi'))\frac{(Np_i(\xi'))^n}{n!}$$

- Means and variance: $Np_i(\xi')$

- Substituting the variance so we have:

$$var\left(p_0^{est}(\xi')\right) = \sum_{i=1}^{M} \frac{w_i^2 N_i f_i c_i(\xi') p_0(\xi')}{N_i^2 c_i^2(\xi) f_i^2} = \sum_{i=1}^{M} \frac{w_i^2 p_0(\xi')}{N_i c_i(\xi) f_i}$$

- We want to minimize equation with the constraint $\sum_{i=1}^{M} w_i = 1$, so we define a Lagrange multiplier λ, giving a goal function:

$$G = \sum_{i=1}^{M} \frac{w_i^2 p_0(\xi')}{N_i c_i(\xi) f_i} + \lambda \sum_{i=1}^{M} w_i$$

- To minimize the goal function, first take the partial derivative:

$$\frac{\partial G}{\partial w_i} = \frac{2w_i p_0(\xi')}{N_i c_i(\xi') f_i} + \lambda$$

- And setting it to zero, obtain:

$$w_i = -\frac{N_i c_i(\xi') f_i}{2p_0(\xi')}\lambda$$

- Substitute to the constraints:

$$\sum_{i=1}^{M} -\frac{N_i c_i(\xi') f_i}{2p_0(\xi')}\lambda = 1$$

- so we have:

$$\lambda = -\frac{2p_0(\xi')}{\sum_{i=1}^{M} N_i c_i(\xi') f_i}$$

- and obtain the weight:

$$w_i = \frac{N_i c_i(\xi') f_i}{\sum_{i=1}^{M} N_i c_i(\xi') f_i}$$

Substitute this weight and finally we get:

$$p_0^{est}(\xi') = \sum_{i=1}^{M} w_i \frac{h(\xi')}{N_i f_i c_i(\xi')} = \frac{\sum_{i=1}^{M} h_i(\xi')}{\sum_{i=1}^{M} N_i f_i c_i(\xi')}$$

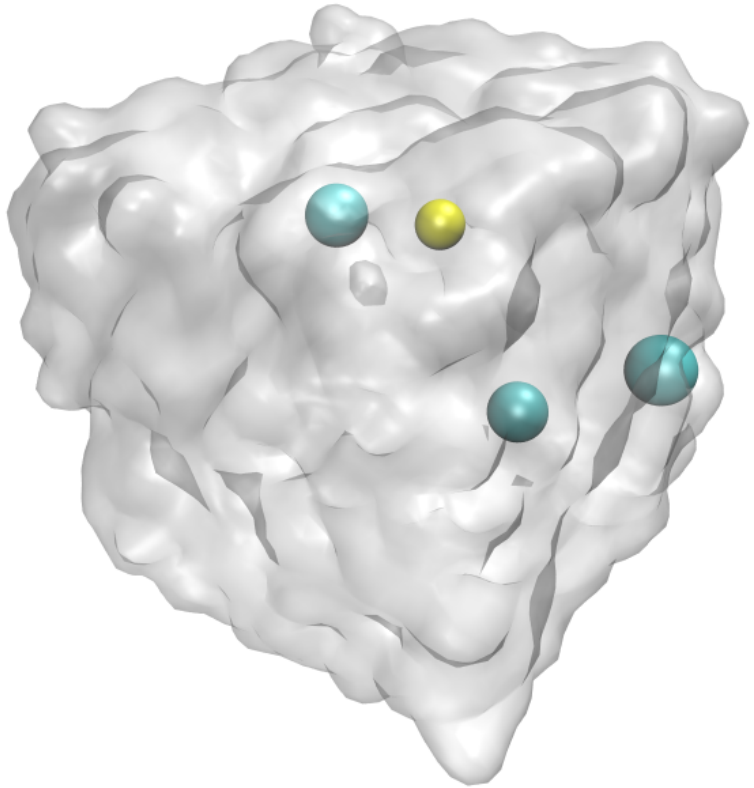
This equation and the normalization condition:

$$f_i^{-1} = \int_{\Xi} c_i(\xi') p_0^{est}(\xi') \, d\xi'$$

are **collectively known as the Weighted Histogram Analysis Method (WHAM) equations**.

Free energy of ion interaction in solution

Model detail

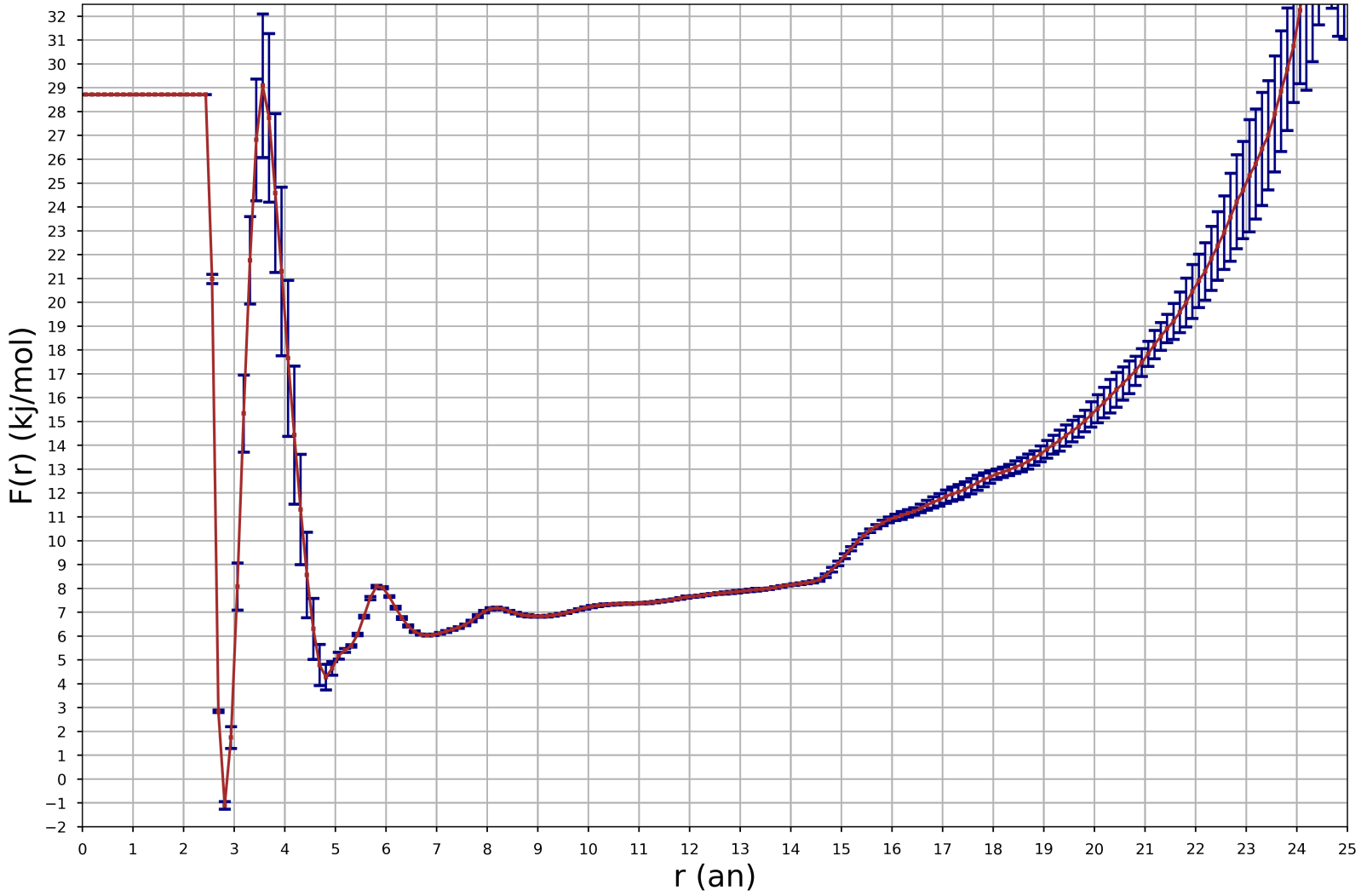


- Simulation tool: OpenMM
- Number of atoms: 2072
- Thermostat: Andersen Thermostat
- Barostat: MonteCarlo Barostat
- Time step: 1 femtosecond
- Equilibrium temperature: 300K
- Equilibrium pressure: 1 Bar

Umbrella sampling settings

- **10** replicas
- Each has **200** trajectories
- Each run for **200ns**
- Each executed for **10 hours**

SCHM Result

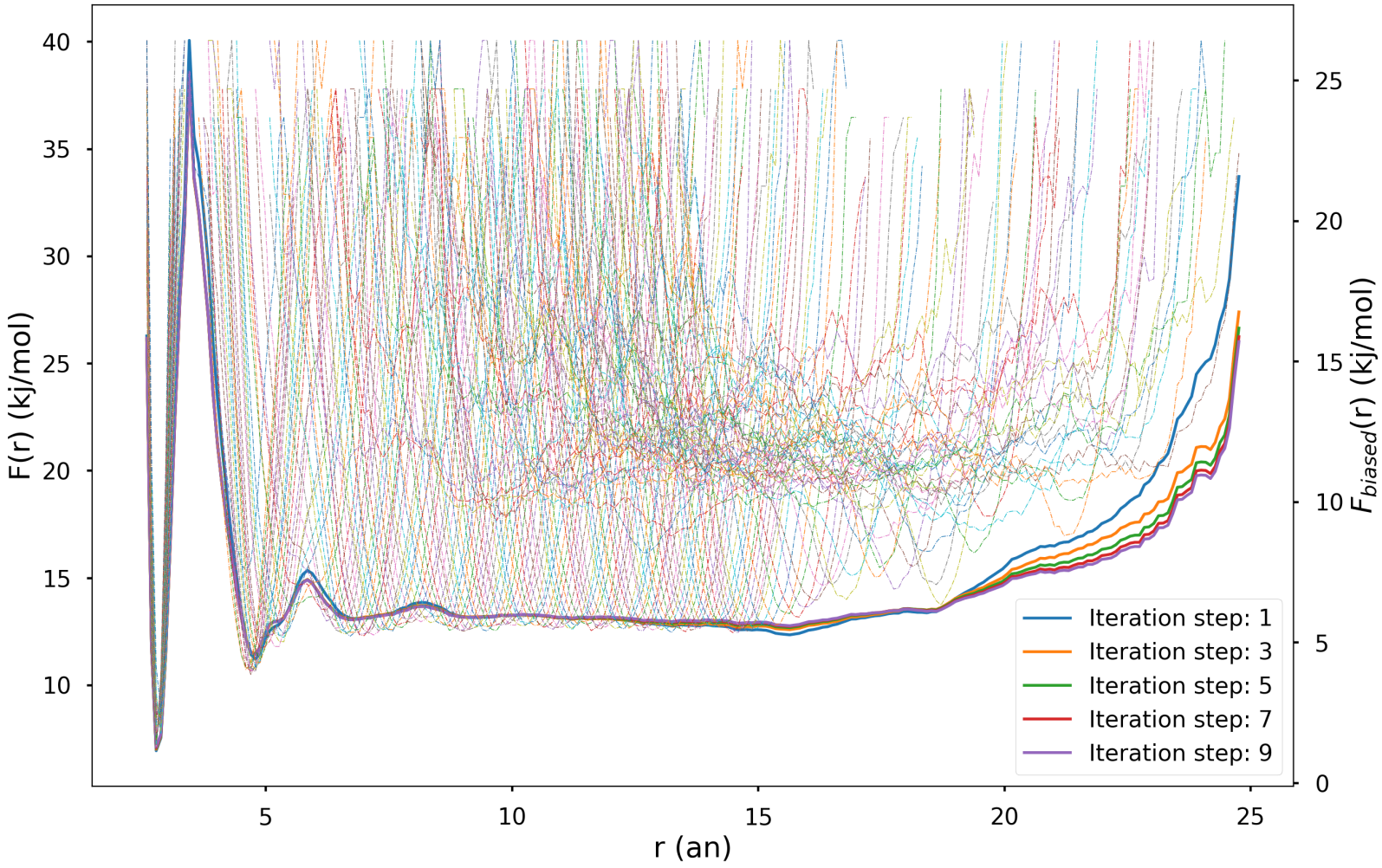

F(r) (kj/mol)
r (an)
32
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
−1
−2
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

MDAnalyser Demo

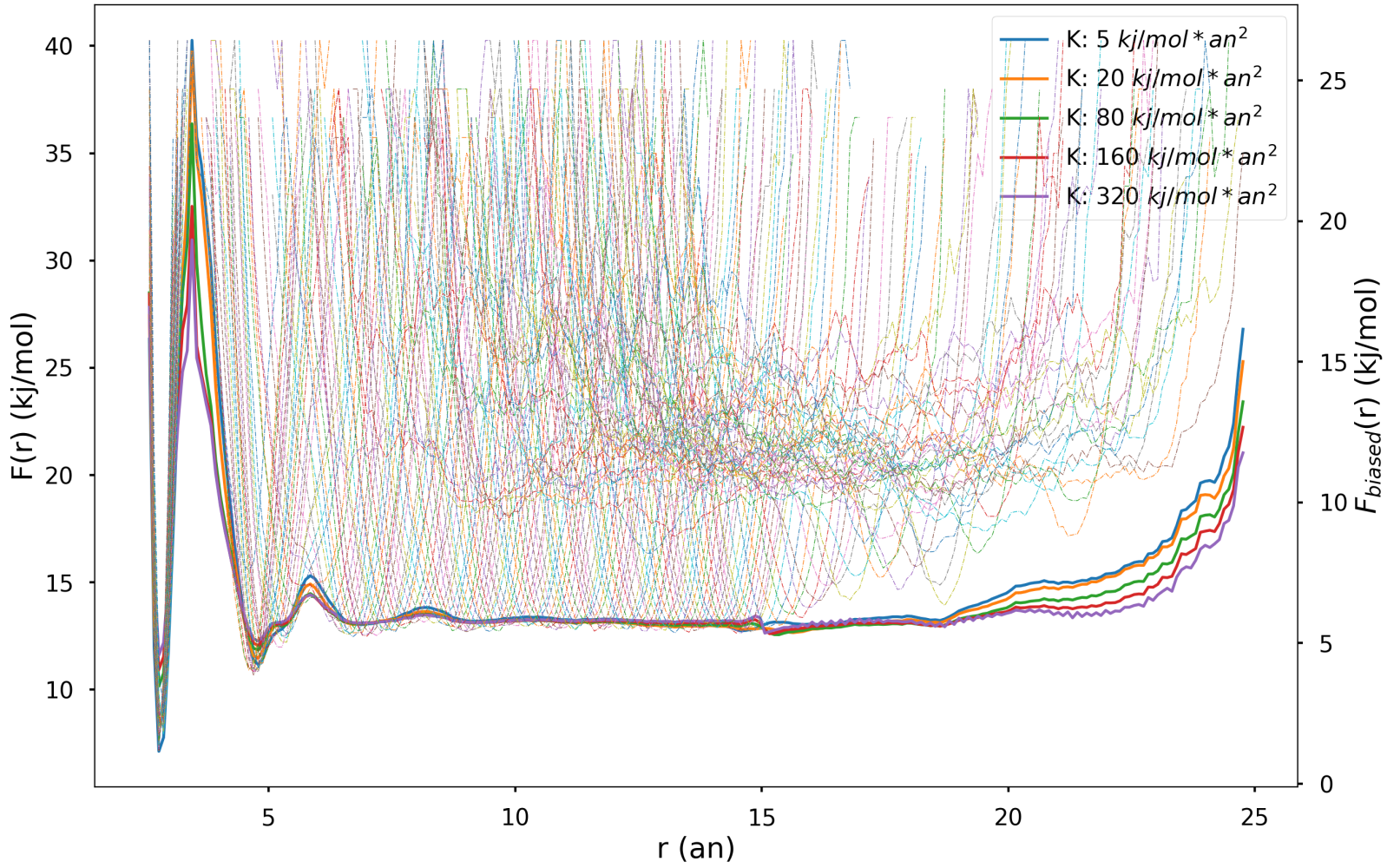
```
from mdanalyser.analyser import WHAMUmbrellaSamplingAnalyser

path_current = sys.path[0]
path_out_cv = path_current + '/../output/cv_files'

analyzer = WHAMUmbrellaSamplingAnalyser()

analyzer.setDirPath(path_out_cv)
analyzer.setFileIdRangeFromStartToEnd(0, 99)
analyzer.setBinRangeFromStartToEnd(1, 15, 250)
analyzer.setConstantK(100)
analyzer.setTemperature(300)

analyzer.loadData()
analyzer.iterativeSolver(num_steps=500)
```

Validation:

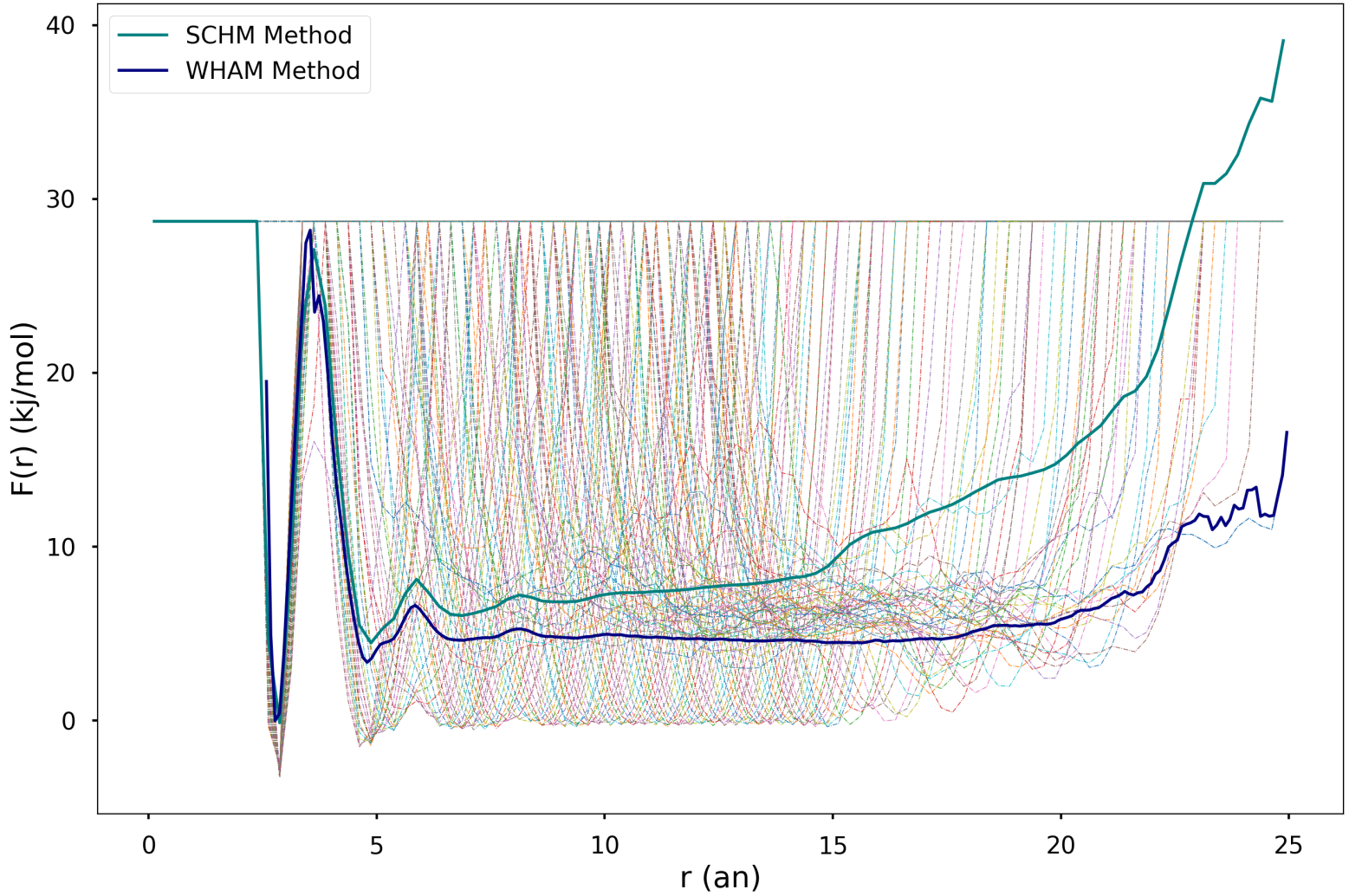
- Quickly convergence



- Sensitive to the constant K of harmonic oscillator bias potential

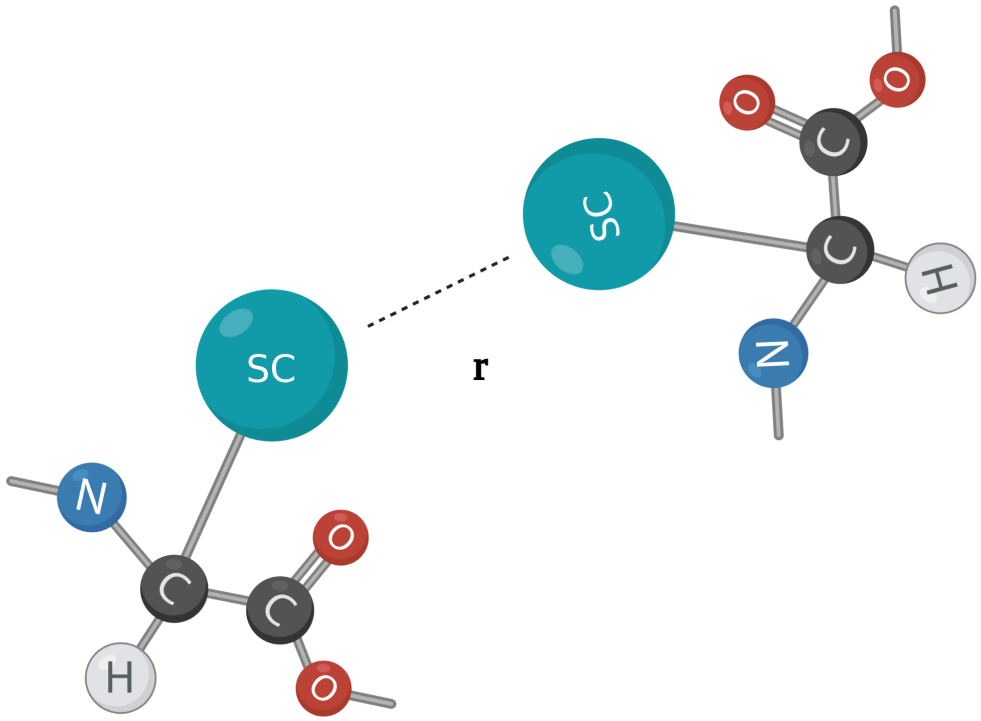


- Better performance than **Self-Consistent Histogram Method** (SCHM) in sparse sampling part



Free energy matrix

Model details



- Two amino acid
- number of atoms: around 5000
- NaCl concentration: 0.15mol/L
- time step: 1 fs
- thermostat: Langevin
- barostat: MonteCarlo 1bar
- integrator: Langevin Integrator
- forcefield: 'amber14/tip3pfb.xml'

Collective variable details

- **Defination**:

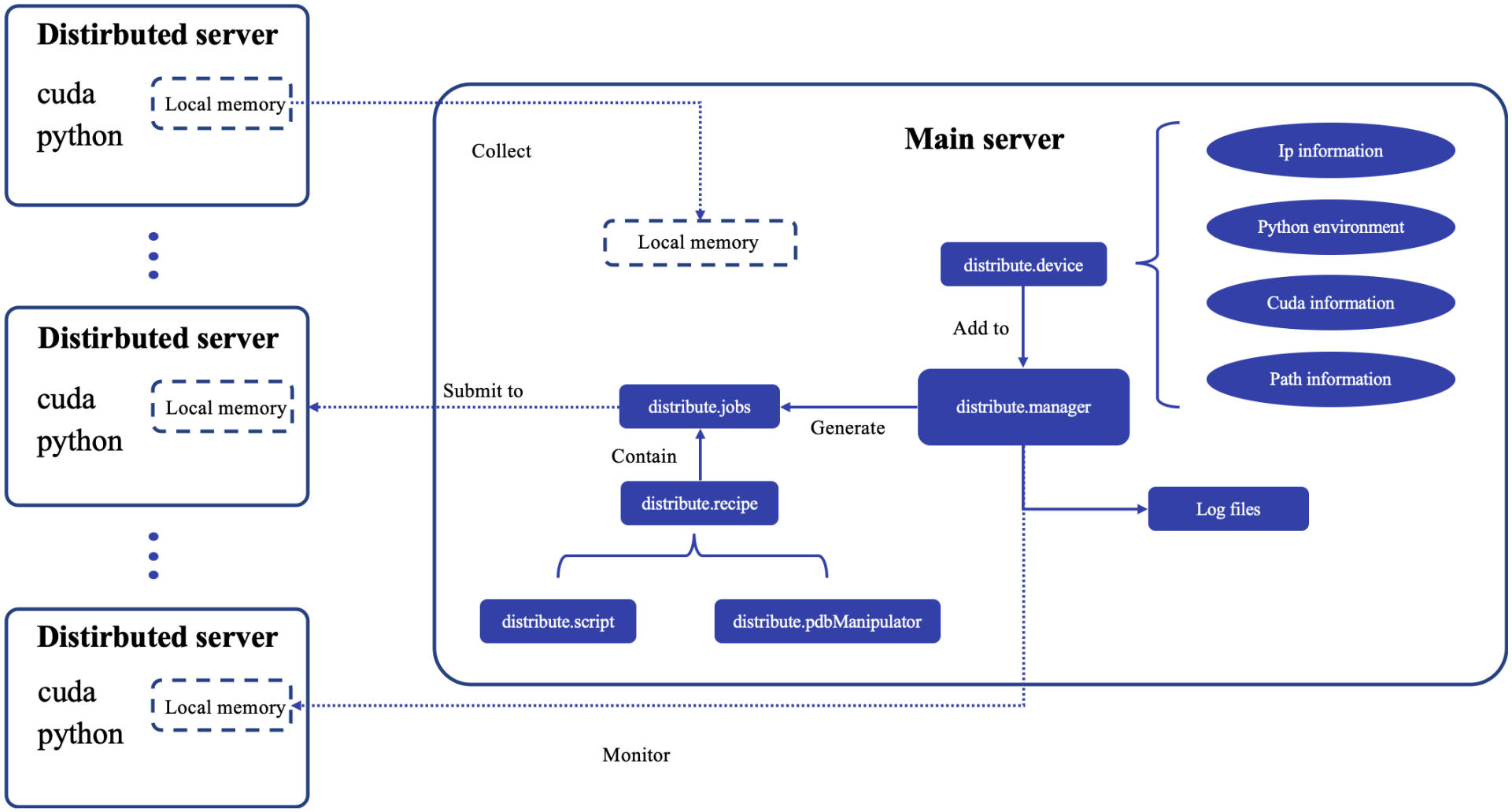
  The distance between the centroid center of two amino acid

$$\Delta U = \frac{1}{2} K (\mathbf{c}_{dist} - \mathbf{c}_0)^2,\ \mathbf{c}_{dist} = \frac{1}{M_{tot}} \sum_{i=1}^{N} m_i \mathbf{q}_i$$

$$F_{x_i} = \frac{\partial \Delta U}{\partial c_{dist,x}} \frac{\partial c_{dist,x}}{\partial x_i} = \frac{m_i}{M_{tot}} K c_{dist,x}$$

- **Range**: $r \in [1, 20] A$
- **Grid**: 200 trajectories: 150 $\in [1, 15)$, 50 $\in [10, 20)$

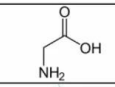
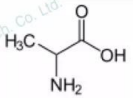
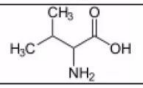
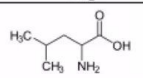
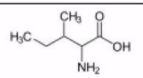
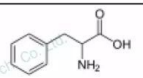
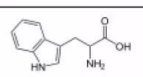
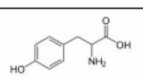
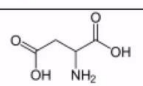
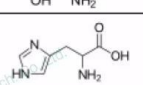
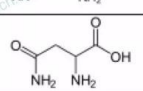
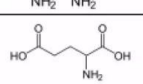
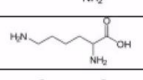
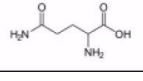
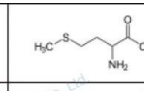
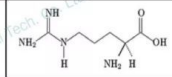
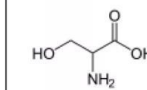
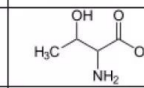
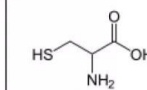
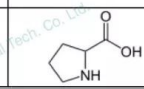
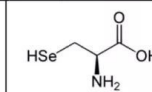
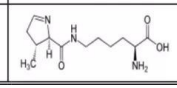
Pdff-distribute



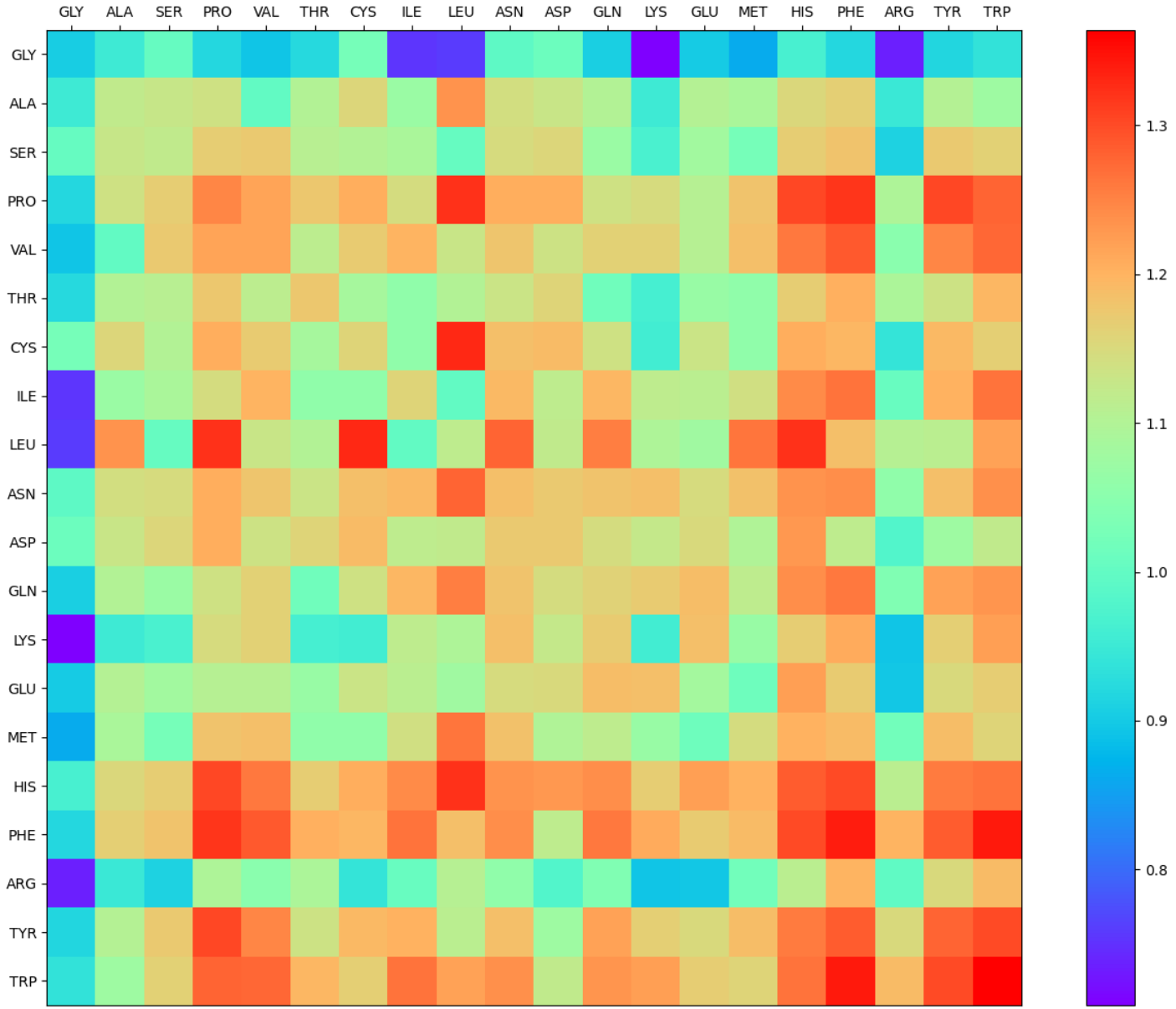
Result

	ALA	ARG	ASN	ASP	CYS	GLN	GLU	GLY	HI
ALA	1.119	0.949	1.142	1.129	1.156	1.103	1.103	0.954	1.15
ARG	0.949	0.996	1.058	0.98	0.942	1.039	0.897	0.735	1.11
ASN	1.142	1.058	1.186	1.174	1.185	1.18	1.148	0.995	1.23
ASP	1.129	0.98	1.174	1.172	1.192	1.145	1.151	1.013	1.22
CYS	1.156	0.942	1.185	1.192	1.157	1.137	1.134	1.026	1.20
GLN	1.103	1.039	1.18	1.145	1.137	1.16	1.188	0.908	1.24
GLU	1.103	0.897	1.148	1.151	1.134	1.188	1.083	0.901	1.21

氨基酸性质表

	缩写	名称	结构	分子量	等电点	溶解度 水 (0,20°C)g/L	分类
G	Gly	甘氨酸 Glycine		75.07	6.06	141.8 225.2	脂肪族类
A	Ala	丙氨酸 Alanine		89.09	6.11	127.3 157.8	脂肪族类
V	Val	缬氨酸 Valine		117.15	6	59.6 68.1	脂肪族类
L	Leu	亮氨酸 Leucine		131.17	6.01	22.70 23.74	脂肪族类
I	Ile	异亮氨酸 Isoleucine		131.17	6.05	37.91 40.25	脂肪族类
F	Phe	苯丙氨酸 Phenylalanine		165.19	5.49	19.83 27.35	芳香族类
W	Trp	色氨酸 Tryptophan		204.23	5.89	8.23 10.57	芳香族类
Y	Tyr	酪氨酸 Tyrosine		181.19	5.64	0.196 0.384	芳香族类
D	Asp	天冬氨酸 Aspartic acid		133.1	2.85	2.62 6.33	酸性氨基酸类
H	His	组氨酸 Histidine		155.16	7.6	41.9 (25℃)	碱性氨基酸类
N	Asn	天冬酰胺 Asparagine		132.12	5.41	8.49 23.51	酰胺类
E	Glu	谷氨酸 Glutamic acid		147.13	3.15	8.55 17.22	酸性氨基酸类
K	Lys	赖氨酸 Lysine		146.19	9.6	400 630	碱性氨基酸类
Q	Gln	谷氨酰胺 Glutamine		146.15	5.65	7.2 (37℃)	酰胺类
M	Met	甲硫氨酸 Methionine		149.21	5.74	18.18 29.95	含硫类
R	Arg	精氨酸 Arginine		174.2	10.76	855.6 (25℃)	碱性氨基酸类
S	Ser	丝氨酸 Serine		105.09	5.68	22.04 42.95	羟基类
T	Thr	苏氨酸 Threonine		119.12	5.6	13.2 (25℃)	羟基类
C	Cys	半胱氨酸 Cysteine		121.16	5.05	0.11 (25℃)	含硫类
P	Pro	脯氨酸 Proline		115.13	6.3	1620 (25℃)	亚氨基酸
U	Sec	硒半胱氨酸 Selenocysteine		168.07			
O	Pyl	吡咯赖氨酸 Pyrrolysine		255.31			

Sort by molecular weight

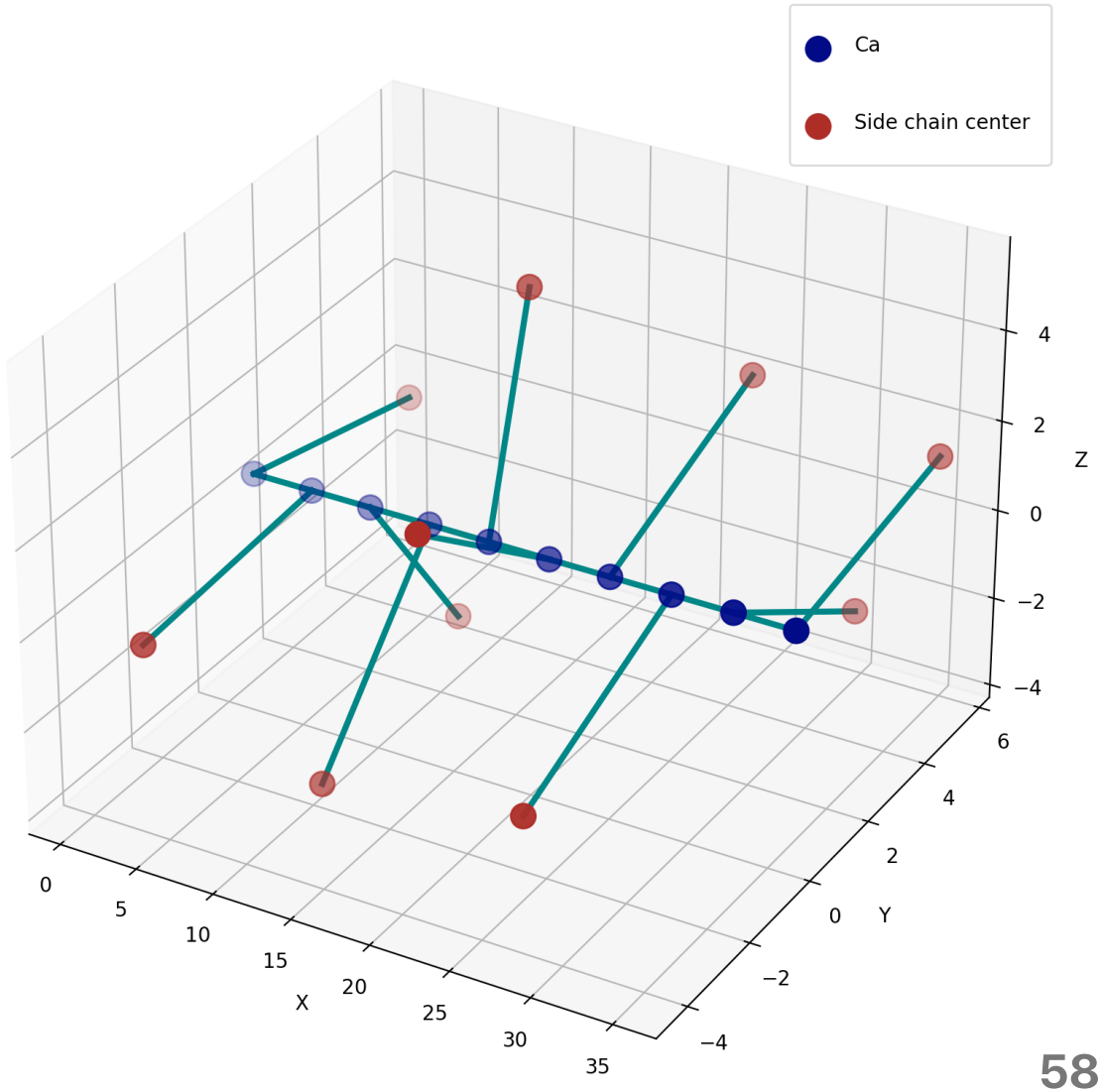


OpenPD

Coarse-Grained Model

The protein is treated as a peptides chain, each of which consists two beads:

- **CA** bead represents the backbone part of the peptide;
- **SC** bead represents the side chain part of the peptide;



Force field expression

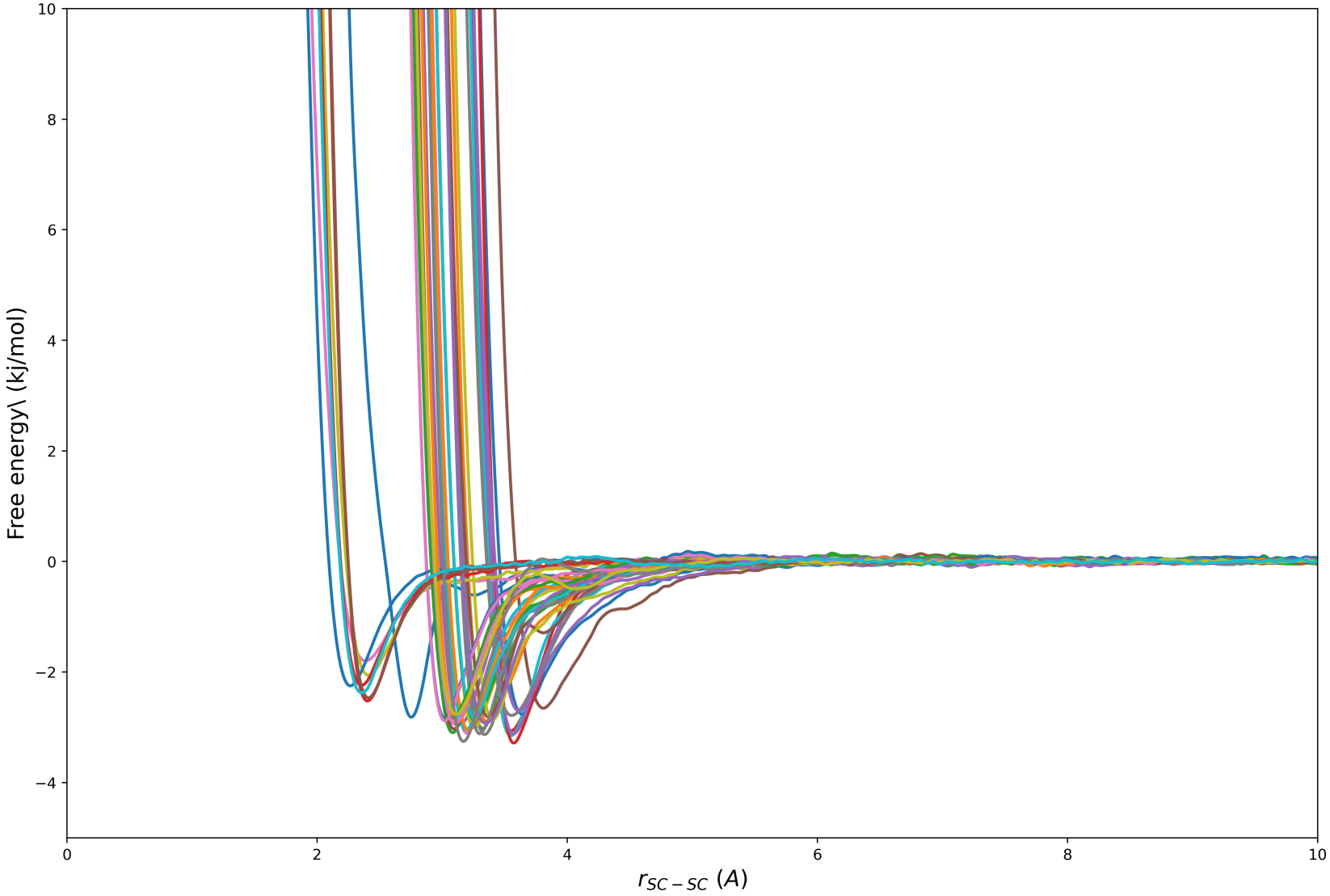
$$
\begin{aligned}
U_{tot} &= U_{Non-Bonded} + U_{Bond} + U_{Torsion} \\
&= \sum_{i=1}^{N_A} \sum_{j=i+1}^{N_A} U_{Non-Bonded}^{(ij)} \left(\mathbf{r}_{ij}\right) + \sum_{i=1}^{N_B} U_{Bond}^{(i)} \left(l_i\right) + \sum_{i=1}^{N_T} U_{Torsion}^{(i)} \left(\theta_i\right)
\end{aligned}
$$

- $U_{Non-Bonded}$ describes the interaction between **SC** and **SC** (no neighbor) bead.
- U_{Bond} describes the bond that constraint the connected bead.
- $U_{Torsion}$ describes the torsion angle preference between neighbor **SC** bead.

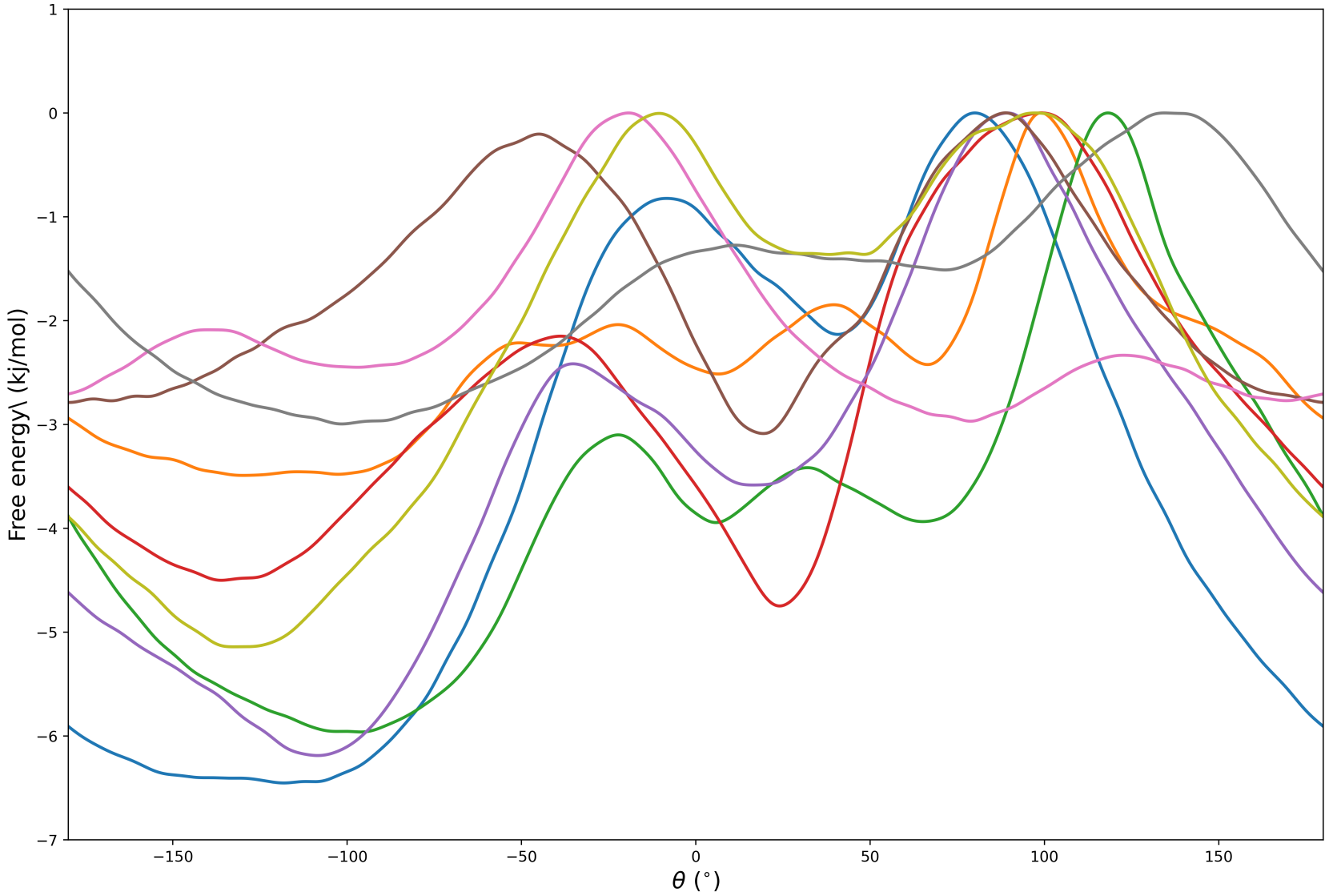
Computational detail

- Non bonded interaction:
  - CV: the distance between centroid of two peptides' side chain;
  - Sampling algorithm: **Umbrella sampling**; 500ns;
- Bond interaction:
  - CV: the distance between two target beads;
  - Sampling algorithm: **Umbrella sampling**; 500ns;
- Torsion interaction:
  - CV: $^{2}\angle SC-CA-CA-SC$;
  - Sampling algorithm: **Well-tempered Metadynamics**; 200ns;

Non Bonded Result



Torsion Result



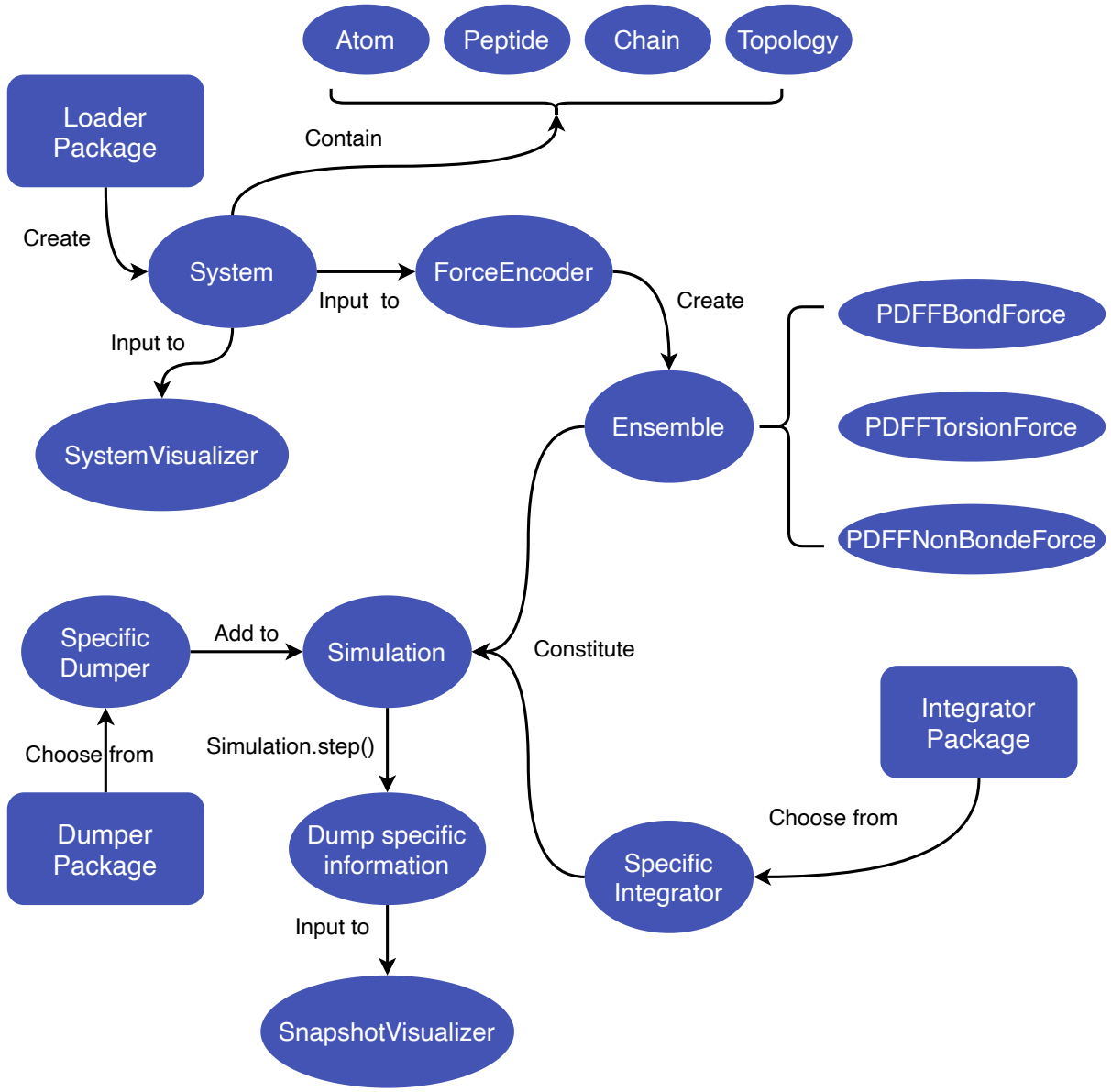
Difference from ordinary coarse grained force field

- PDFF does not contain solvent term;
- PDFF is not fitting from experimental data or other result;
- PDFF is, theoretically, generalized enough for **all** protein, no matter whether exists homologous pair in the database

Overview

OpenPD, standing for **Open P**eptide **D**ynamics, is a **python** package, distributed freely under the terms of **GPLv3**, for peptide dynamics simulation.

- https://github.com/openpd-dev/openpd
- https://openpd.net



Extensible packages

- **openpd.loader** package:
  - PDBLoader, SequenceLoader
- **openpd.force** package:
  - PDFFNonBondedForce, PDFFBondForce, PDFFTorsionForce
- **openpd.integrator** package:
  - VerletIntegrator, VelocityVerletIntegrator, BrownianIntegrator ...
- **openpd.dumper** package:
  - LogDumper, SnapshotDumper, PDBDumper, XYZDumper

Current

Numba

NumPy

Planed

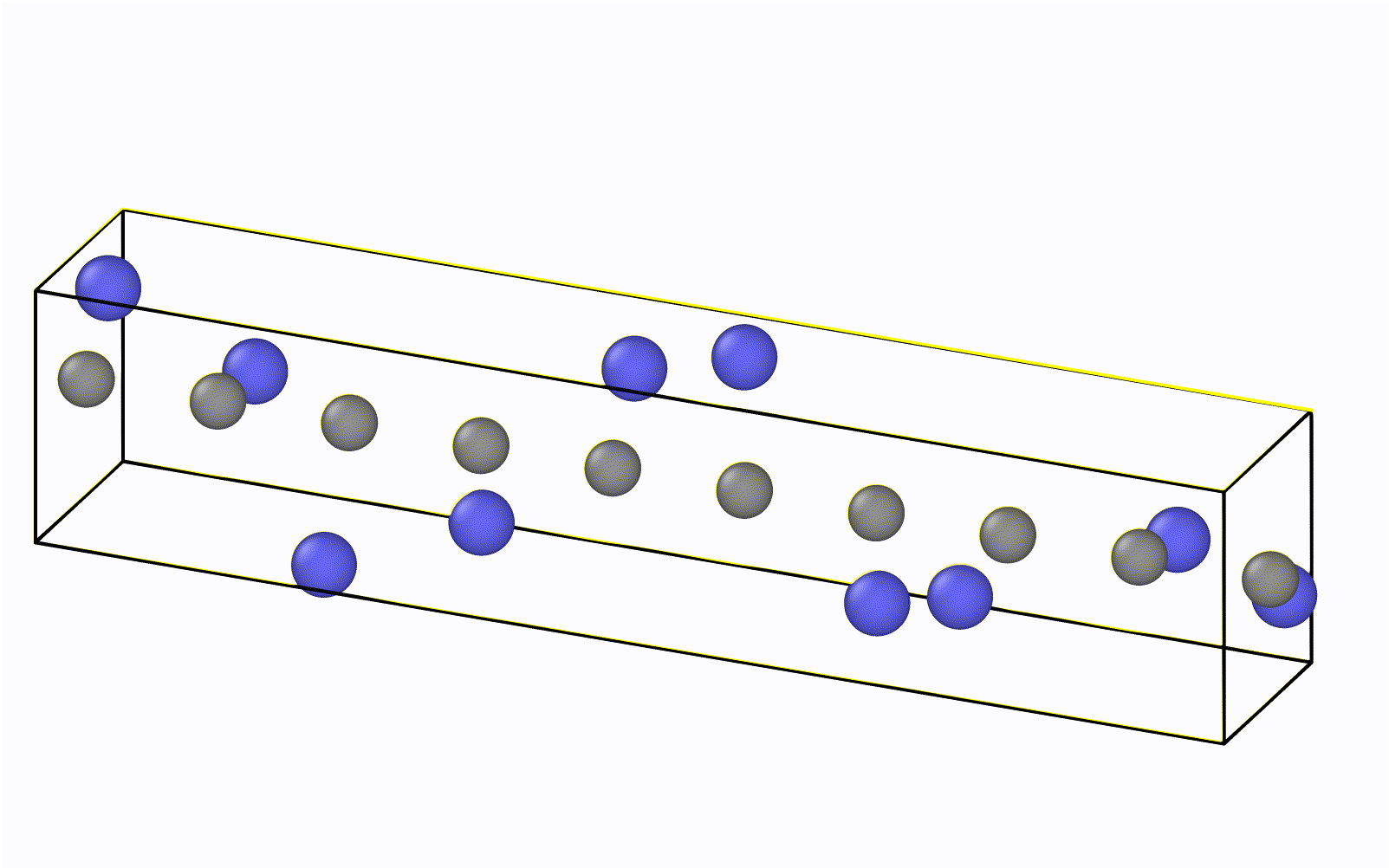
OpenCL™

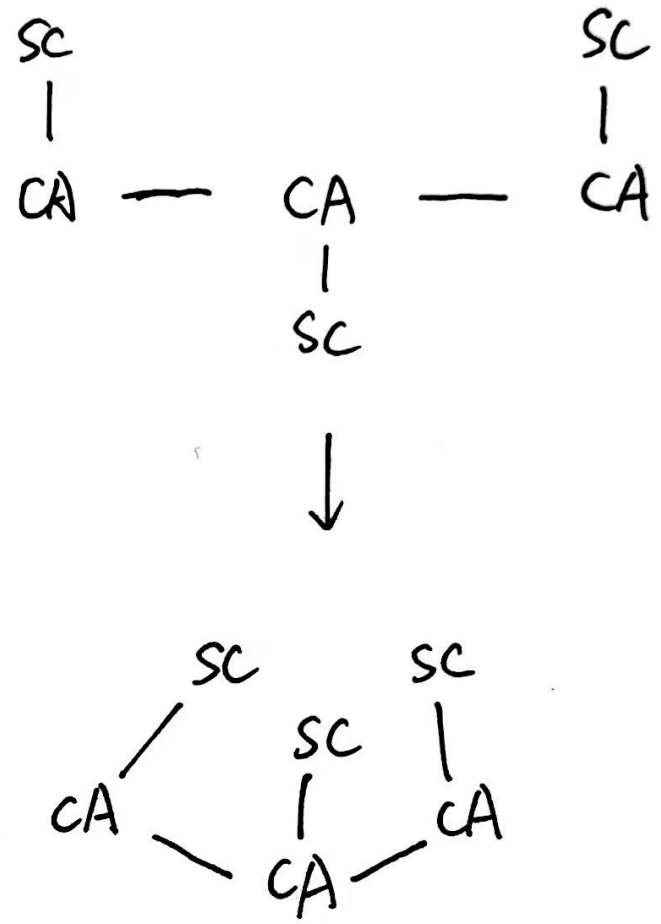
NVIDIA
CUDA

Demo Code

```
import openpd as pd
from openpd.unit import *

system = pd.SequenceLoader('data/simulation.json').createSystem()
ensemble = pd.ForceEncoder(system).createEnsemble()
integrator = pd.VelocityVerletIntegrator(1*femtosecond, 300*kelvin)
simulation = pd.Simulation(ensemble, integrator)

log_dumper = pd.LogDumper(
    'simulation.log', 50, get_potential_energy=True, get_temperature=True,
    get_steps=True, get_elapsed_time=True, get_remain_time=True
)
snapshot_dumper = pd.SnapshotDumper('simulation.pds', 50)
xyz_dumper = pd.XYZDumper('simulation.xyz', 50)
simulation.addDumpers(log_dumper, snapshot_dumper, xyz_dumper)

simulation.minimizeEnergy('gd', max_iteration=20)
simulation.step(50000)
```

Result





Discussion

Non-Bonded interaction only considered the **SC** bead;

The constraint on **CA** bead is completely **locally**:

- Neglect the interaction of distance between **CA** and non-neighbored **SC**;
- Neglect the interaction of distance between **CA** and non-neighbored **CA**;

Future proposal

- Implicit solvent
  - Machine learning solver for PE, PBE, LPBE
  - Solvent boundary potential for **arbitrary boundary**
- Coarse grained model
  - Refinement of PDFF
- Research **large functional** proteins
  - DNA polymerase

Acknowledge

- Thanks for Prof. **Yunfei Chen**'s guidance with patience
- Thanks the **Big Data Computing Center of Southeast University** for providing the facility support on the numerical calculations in this paper
- Thanks for the sponsorship of **National Natural Science Foundation of China**

Questions & Answers