SUPPORTING INFORMATION

for

OpenMSCG: A Software Tool for Bottom-up Coarse-graining

Yuxing Peng,[1] Alexander J. Pak,[2] Aleksander E. P. Durumeric,[3] Patrick G. Sahrmann,[4] Sriramvignesh Mani,[4] Jaehyeok Jin,[4] Timothy D. Loose,[4] Jeriann Beiter,[4] and Gregory A. Voth*[4, a)]

[1] NVIDIA Corporation, 2788 San Tomas Expressway, Santa Clara, CA 95051, USA

[2]Department of Chemical and Biological Engineering, Colorado School of Mines, Golden, Colorado 80401, USA

[3]Department of Mathematics and Computer Science, Freie Universität Berlin, Berlin, Germany

[4]Department of Chemistry, Chicago Center for Theoretical Chemistry, James Franck Institute, and Institute for Biophysical Dynamics, The University of Chicago, Chicago, Illinois 60637, USA

[a)] Author to whom correspondence should be addressed: gavoth@uchicago.edu

Dynamic Programming Algorithm for Essential Dynamics Coarse-Graining (EDCG)

The goal of EDCG is to minimize the residual from mapping $\boldsymbol{n}$ C_α atoms of a protein into $\boldsymbol{N}$ CG sites, which is defined as

$$\chi = \frac{1}{3N}\sum_{I=1}^{N} \mathbf{C}_I$$

where

$$\mathbf{C}_I = \sum_{i\in I}\sum_{j\geq i\in I} \langle \left|\Delta \boldsymbol{r}_i^{ED} - \Delta \boldsymbol{r}_j^{ED}\right|^2 \rangle$$

is the loss of total covariance from mapping a group of C_α atoms to the CG site $\boldsymbol{I}$. In a sequential/linear model, C_α atoms associated with each CG site are assumed contiguous in the protein primary sequence, and a loss function for mapping a group of consecutive C_α atoms $\{a, a+1, a+2 \ldots b\}$ into a CG site can be rewritten as

$$\mathbf{C}(a,b) = \sum_{i=a}^{b-1}\sum_{j=i+1}^{b} \langle \left|\Delta \boldsymbol{r}_i^{ED} - \Delta \boldsymbol{r}_j^{ED}\right|^2 \rangle$$

Therefore, to calculate the loss function for all possible values of $1 \le a < b \le n$, the time complexity is $O(N^4)$. To gain a higher computational efficiency, the loss function can be calculated from the recurrence equation:

$$\mathbf{C}(a,b) = \begin{cases} \mathbf{C}(a,b-1) + \mathbf{C}(a+1,b) - \mathbf{C}(a+1,b-1) + \langle |\Delta \boldsymbol{r}_a^{ED} - \Delta \boldsymbol{r}_b^{ED}|^2 \rangle, & a < b \\ 0, & a = b \end{cases}$$

The pseudo code with a time complexity of $O(N^2)$ can be designed as

```
loop L from 1 to n # number of Cα atoms to be mapped
    loop a from 1 to n-L # starting Cα atom
        b = a + L – 1 # ending Cα atom
        Calculate C(a,b) # calculate the loss
```

After obtaining all values of the loss functions, we can then define a sub-residual, $\chi(k,m)$, which is the minimum of total loss by mapping the first $\boldsymbol{k}$ C_α atoms into $\boldsymbol{m}$ CG sites, where $1 \le k \le n$, and $1 \le m \le N$ and $m \le k$. A recurrence equation can then be defined as

$$\chi(k,m) = \begin{cases} \min\limits_{\mathbf{1 \le i \le k-m+1}} \{\chi(k-i,m-1) + C(k-i+1,k)\}, & m > 1 \\ C(1,k), & m = 1 \end{cases}$$

The pseudo code can be designed as

```
Chi(k,1) = C(1,k)
loop m from 1 to N
    loop k from 1 to n
        Chi(k,m) = min{Chi(k-i,m-1)+C(k-i+1,k)}
```

The time complexity is $O(N^3)$ for calculating all sub-loss functions and the final residual $\chi(n,N)$ is the global minimum. This algorithm, known as dynamic programming, can be used to obtain the globally optimized solution.