

Three-Dimensional Reconstruction Using the Encounter Probabilities

January 27, 2015

1 Using optimization methods

Given an encounter probability profile of the first bead in a Rouse chain of N bead, we use the expected encounter model in the Rouse chain to give estimate on the 3D organization of the chain. Given the encounter profile e , we minimize the goal function of the form

$$\min\{f(d)\} \quad (1)$$

$$Ad \leq b \quad (2)$$

$$A_{eq}d = b_{eq} \quad (3)$$

$$l_b \leq d \leq u_b \quad (4)$$

$$c(d) \leq 0 \quad (5)$$

$$c_{eq}(d) = 0 \quad (6)$$

where d is a vector of the distances of bead from bead 1, $f = \sum d_j$, $A \in M_{[N \times N]}$ is a matrix such that

$$A_{ii} = \begin{cases} -1 & \text{if } \exists j \quad ; \|e_j - e_i\| \leq \epsilon \\ 0 & \text{else} \end{cases} \quad (7)$$

$$A_{ij} = \begin{cases} 1/|J| & \text{if } \|e_j - e_i\| \leq \epsilon \\ 0 & \text{else} \end{cases} \quad (8)$$

for small ϵ of choice, and J is the group of all indices j such that $\|e_j - e_i\| \leq \epsilon$ for each i . The vector b is the zero vector, $b = 0$, l_b is the lower bound, set

to be a vector of all ones, u_b is the upper bound set to be a vector of all N . The non linear constraint

$$c(d) = \sum_{i=1}^N \left(\frac{d_i^{-1.5}}{\sum_{i=1}^n d_i^{-1.5}} - e_i \right)^2 \quad (9)$$

2 Using inverted probabilities assuming a Rouse chain

The procedure relies on the nearest neighbor connectivity. As with the previous method, we assume a Rouse polymer encounter probability formula, which drops with the linear distance on the chain. The main observation here is that if we estimate the nearest neighbors for each one of the beads, we can construct the whole connectivity map of the polymer. For this end, we take the encounter probability profile for each bead, and look for beads in distances for correspond to the nearest neighbor probability.

2.1 Smoothing the encounter signal

To locate prominent features of the signal, we have to reduce noise accompanying the experimental data. There are several ways to smooth the encounter probability data. We list few plausible methods. The smoothing is performed on the encounter frequency signal, before calculating the encounter probability.

2.1.1 Iterative Gaussian Smoothing

We start by iteratively applying Gaussian kernel smoothing with increasing variance for several iterations. The number of iteration is determined by the numerical accuracy of the machine used, which produces noise after several iterations, depending on machine precision, which prevents us from applying the filter indefinitely.

One possible numerical criterion for the presence of noise, is the change in the number of local maxima the signal has over all values of variance chosen in the smoothing procedure. We then choose the variance value that give rise to the minimal number of maximal points.

2.1.2 $\mu|\lambda$ smoothing

Taubin's $\mu|\lambda$ smoothing procedure [1] works by sequentially applying Gaussian smoothing with alternate positive and negative scales factors λ and μ respectively, for N iterations. In fact, any type of weighting other than the Gaussian can be applied, with the condition that the weights sum to one, and that no vertex is included in its own neighborhood.

At first we compute v'_i from v_i by adding to it an averaging of its neighborhood with a positive scale factor λ

$$v'_i = v_i + \lambda \sum_{j \in i^*} (v_j - v_i) w_j$$

Then, we compute v''_i from v'_i using a negative scale factor $\mu < -\lambda < 0$

$$v''_i = v'_i + \mu \sum_{j \in i^*} (v'_j - v'_i) w_j$$

with w_i the weighting mask, i^* are the indices of the vertex i neighborhood excluding itself.

In practice, we perform this by two steps convolution over N iterations with a $n_v \times n_v$ square convolution kernel W . The kernel is changed such that $W_{n_v/2, n_v/2} = 0$; $W_{ij} \leftarrow W_{ij} / \sum \sum W_{ij}$. For an input signal V , $V' = V + \lambda(V * W - V)$ and $V'' = V' + \mu(V' * W - V')$, over N iterations.

The variables μ, λ and N are determined by solving the inequalities

$$\begin{aligned} ((\lambda - \mu)^2 / (-4\lambda\mu))^N &< 1 + \delta_{pB} \\ ((1 - \lambda k_{sB})(1 - \mu k_{sB}))^N &< \delta_{sB} \end{aligned}$$

with the constraints:

$$N > 0, \quad 0 < \lambda < -\mu, \quad \lambda < \frac{1}{k_{sB}}, \quad \frac{1}{\lambda} + \frac{1}{\mu} = k_{pB}$$

The filter's parameters are then k_{sB}, k_{pB} . We have to change the shape of the neighborhood for the convolution operations, to take into account the shape of the encounter frequency vs. distance matrix.

2.2 The steps

1. Obtain the encounter histogram of the HiC data.

2. Arrange the encounters as 3D matrix E , with x -axes as bead index $b = 1, \dots, N$, y -axes as the distance $d = -N..N$ along the chain, and the z -axes the number of encounters $E(b, d)$. Arranging the HiC data from left to right, we distinguish between 'right' encounter matrix, $e_r(b, d)$ for $d > 0$ and 'left' encounter matrix $e_l(b, -d)$ for $d < 0$ such that $E = e_l(b, -d) \oplus e_r(b, d)$ which is $N \times 2N$ matrix.
3. Smooth $e_r(b, d)$ and $e_l(b, -d)$ using the procedure described in 2.1.2.
4. Normalize each row of the smoothed encounter histogram to get probabilities [Read about bias]
5. Analyze $e_l(b, -d)$ and $e_r(b, d)$ using the procedure described in [Missing subsection] to construct the connectivity graph.

References

- [1] Gabriel Taubin. Curve and surface smoothing without shrinkage. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 852–857. IEEE, 1995.