

# **Apply Locally Linear Embedding Technique to Model Three-Dimensional Chromatin Structures**

(Thesis Proposal for Computational Biology)

*Lingyu Wei, Zhengyu Wang*

*2013-5-15*

## **Introduction**

Three-Dimensional Chromatin Structure Modeling via frequency data is an important task in computational biology research, and there are several methods proposed to handle it [6][7]. However, those methods are indirect ways by converting interaction frequency into local distance of using the interaction frequency data, which will be very noisy (i.e. having a large variance) when two fragments are far away from each other. Locally Linear Embedding (LLE) [2,1,4] is among the most important Nonlinear Dimensionality Reduction techniques [5], which have extensive applications in manifold learning. It only considers several nearest neighbors (i.e. local information) for calculating a low dimensional embedding. Therefore, we plan to use LLE technique to embed the chromatin structures into the 3D Euclidean space, by directly using the interaction frequency data of pairs of DNA fragments as the distance measure. Our project expects a better experiment result over the previous methods [6][7].

## **Locally Linear Embedding(LLE)**

We follow the work by Saul and Roweis [1][2]. And the following is the original Locally Linear Embedding algorithm proposed in [1]. The main idea is that if the sample data are dense enough on a low dimensional manifold, any of the points can be approximately expressed as an affine combination of its neighbors. We reserve the affine combinations and fit the data points in a smaller dimension.

<b>LLE Algorithm:</b>
<b>Parameter:</b> $K$ (The number of neighbors per point) <b>Input:</b> $X_1, \dots, X_N \in \mathbb{R}^D$ <b>Output:</b> $Y_1, \dots, Y_N \in \mathbb{R}^d$ <ol style="list-style-type: none"> <li>1. Compute the nearest <math>K</math> neighbors of each data point <math>X_i</math>, as <math>\Gamma(i)</math>.</li> <li>2. Compute the weights <math>W_{ij}</math> that best reconstruct each data point <math>X_i</math> from its neighbors, minimizing the cost in <math>E(W) = \sum_i  X_i - \sum_{j \in \Gamma(i)} W_{ij} X_j ^2</math> subjected to <math>\sum_{j \in \Gamma(i)} W_{ij} = 1</math> for every <math>i</math>.</li> <li>3. Compute the vectors <math>Y_i</math> best reconstructed by the weights <math>W_{ij}</math>, minimizing the quadratic form <math>\Phi(W) = \sum_i  Y_i - \sum_{j \in \Gamma(i)} W_{ij} Y_j ^2</math> by its bottom nonzero eigenvectors.</li> </ol>

## Pseudo-Code of Our Algorithm

We modify the LLE Algorithm a bit to fit in interaction frequency data. We can make  $d = 3$  in order to get three-dimensional points as the output. We can select the neighbors by the interaction frequency data because we can assume that the bigger frequency two fragments interact, the nearer they are. The most important step is the modification of step 2 in LLE, i.e., calculating  $W_{ij}$  out of the interaction frequency data. We want to firstly try a very natural way of setting  $W_{ij}$ : proportional to the interaction frequency  $f_{ij}$  ( $f_{ij}$  is the interaction frequency between fragment  $i$  and fragment  $j$ ), i.e.

$$W_{ij} = \frac{f_{ij}}{\sum_{j \in \Gamma(i)} f_{ij}}$$

Where  $j \in \Gamma(i)$ , and otherwise  $W_{ij} = 0$ .

There are also some other possible ways to set  $W_{ij}$ . For example,  $W_{ij}$  is proportional to  $f_{ij}^p$  where  $p > 0$ . And furthermore, we can make use of the frequency  $f_{j,j'}$  where  $j, j' \in \Gamma(i)$  and  $j \neq j'$  to calculate  $W_{ij}$  for point  $i$ . In that case, we utilize more data nearby and possibly achieve better result.

Intuitively, setting  $W_{ij}$  proportional to  $f_{ij}^p$  is like we are assuming the distance of two fragments is inversely proportional to  $f_{ij}^p$  since when we want to minimize the sum of products, the smaller the distance is, the larger the weight would be.

## Data Set

We plan to conduct experiment on the data in [6][7] as our first test data for the compare of performance. After that, we want to further explore it on other interaction frequency data base such as [3]. The data attached on our Homework 4 may also be used for the comparison of our previous homework performance.

## Experiment

We have already got the source code of LLE [4], and so we can soon conduct the experiment. One freedom of the algorithm is the choice of  $K$ . If it is very small, then the neighbors may not provide a very good approximation for a data point. If it is very large, then the algorithm may suffer more from the error in the action frequency data. Therefore, in the experiment, it would be important to explore the performance of the proposed algorithm on different values of  $K$ . Following experiments may cover different choices of algorithm calculating  $W_{ij}$ .

## Reference:

- [1] L. Saul and S. Roweis, *An Introduction to Locally Linear Embedding*.
- [2] S. Roweis and L. Saul, *Nonlinear Dimensionality Reduction by Locally Linear Embedding*. Science **290**, pp.2323-2326 (2000)
- [3] <http://chromosome.sdsc.edu/mouse/hi-c/index.html>
- [4] <http://www.cs.nyu.edu/~roweis/lle/>
- [5] [http://en.wikipedia.org/wiki/Nonlinear\\_dimensionality\\_reduction](http://en.wikipedia.org/wiki/Nonlinear_dimensionality_reduction)
- [6] Fraser J, Rousseau M, Shenker S, Ferraiuolo MA, Hayashizaki Y, Blanchette M, Dostie J., *Chromatin Conformation Signatures of Cellular Differentiation*. Genome Biol. 2009;10(4):R37.
- [7] Rousseau M, Fraser J, Ferraiuolo MA, Dostie J, Blanchette M. *Three-Dimensional Modeling of Chromatin Structure from Interaction Frequency Data Using Markov Chain Monte Carlo Sampling*. BMC Bioinformatics. 2011.