Computational Biology: Progress

Due on Monday, May 12, 2014

Jianyang Zeng

Tianyi Hao, Weiyi Chen

Apply MLT to Model 3D Chromatin Structures

Overview

Since Prof. Jianyang Zeng has told Lingyu Wei and Zhengyu Wang performed well though didn't achieve a good result in this project. So in former two weeks we discussed with them about their former jobs in this project. We hope to develop our project based on their failure, or more exactly, based on their former progress. Last week we have implemented matlab code for their model and tried to analyze why the results of data experiment is not satisfiable.

Roughly speaking, we believe the reason is from model. They have assumed the dependence between interaction frequency and distance is inverse, or later they modified it with one more intersect parameter. This is still too easy compared to the sophisticated 3D chromatin structure. In our future jobs, we will add more parameters and use different models (not just linear models) to iterate these experiment. If the problem still exists, we may need to think of another reason. On the other hand, there is a thought in our discussion. We found that, not only in their project but also in all our homework, the ratio sequence alignment is fixed, however in real world, based on some reports we read in the reference, this is changeable, should also be set as a parameter as well. So this time, we will make the comparison of our generated structure and original structure is by zooming in or out through enumeration.

We hope this will help minimize the difference. In this progress report, we will explain what the problem is, how it shows up and how we generated these improving steps.

Review: Manifold learning techniques and LLE

Manifold Learning pursuits the goal to embed data that originally lies in a high dimensional space in a lower dimensional space, while preserving characteristic properties. This is possible because for any high dimensional data to be interesting, it must be intrinsically low dimensional. High-dimensional data, meaning data that requires more than two or three dimensions to represent, can be difficult to interpret. One approach to simplification is to assume that the data of interest lie on an embedded non-linear manifold within the higher-dimensional space. If the manifold is of low enough dimension, the data can be visualized in the low-dimensional space.

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.

Review: Pseudocode

- Parameter: K(The number of neighbors per point)
- Parameter: $X_1, ..., X_N \in R^D$
- Return: $Y_1, ..., Y_N \in R^d$
- Compute the nearest K neighbors of each data point X, as $\Gamma(i)$
- Compute the weighs $W_{i,j}$ that best reconstruct each data point X_i from its neighbors, minimizing the cost in $E(W) = \sum_i |X_i - \sum_{j \in \Gamma(i)} W_{ij} X_j|^2$ subjected to $\sum_{j \in \Gamma(i)} W_{ij} = 1$ for every i.
- Compute the vectors Y_i best reconstructed by the weights W_{ij}, minimizing the quadratic form $\Phi(W) = \sum_i |Y_i - \sum_{j \in \Gamma(i)} W_{ij} Y_j|^2$ by its bottom nonzero eigenvectors.

How we structure our code

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 $j \in \Gamma(i)$, otherwise $W_{ij} = 0$.
We implement the algorithm in the matlab language currently. This is because Lingyu Wei and Zhengyu Wang were using matlab before, we were hoping the repeat their former experiment using the same data and libraries. After our analysis for their problem, we may still come back to use our python language, as illustrated in research proposal.
The main algorithm is in *lle_chroma.m*. Function *lle_chroma* uses $freq$ (interaction frequency matrix, whose size is $N * N$) and K (the number of nearest neighbors to be considered) as input, and returns the embedding coordinates.

```
function Y = lle_chroma(freq,K)
N = length(freq);

% STEP1: FIND NEIGHBORS
[~,index] = sort(freq,'descend');
neighborhood = index(1:K,:);

% STEP2: SOLVE FOR RECONSTRUCTION WEIGHTS
W = zeros(N,N);
for j = 1:N
    W(neighborhood(:,j),j) = freq(neighborhood(:,j),j);
end
W = W ./ repmat(sum(W),N,1);

% STEP 3: COMPUTE EMBEDDING FROM EIGENVECTS OF COST MATRIX M=(I-W)'(I-W)
M = eye(N) - W;
M = M*M' + eye(N); % Non-singular requirement

% CALCULATION OF EMBEDDING
options.disp = 0; options.isreal = 1; options.issym = 1;
d = 3;
[Y,~] = eigs(M,d+1,0,options);
Y = Y(:,1:d)'*sqrt(N); % bottom evec is [1,1,1,1...] with eval 1
end
```

Lle_eval.m gives the implementation of evaluation function (*lle_eval*), as well as input (readPDB) and output (printPDB) manipulation.

```
function Y = lle_eval(origin, outputfile, freq, p, K) % for example, inputfile =
    'PMA_HoxA_Interactions.txt' and outputfile = 'out.pdb' and K = 7
    %[freq,p] = random_structure();
    Y = lle_chroma(freq,K);
    printPDB(origin, p);
```

```
    printPDB(outputfile, Y);
end

function freq = readPDB(filename)
    input = importdata(filename);
    data = input.data;
    first = min(min(data(:,1:2)))-1;
    last = max(max(data(:,1:2)));
    N = last-first;
    freq = zeros(N);
    for i=1:length(data)
        freq(data(i,1)-first,data(i,2)-first) = data(i,3);
    end
    freq = freq + freq';
end

function printPDB(filename, Y)
    fid=fopen(filename,'w');
    fmt = 'ATOM  % 4d C   LIG A       % 8.3f% 8.3f% 8.3f 1.00 75.00  \n';
    fprintf(fid,fmt, [(1:length(Y)); Y]);
    fprintf(fid,'CONECT % 4d% 4d\n', [(1:length(Y)-1);(2:length(Y))]);
        fprintf(fid,'END');
end
```

Dataset

In our proposal, we planned to conduct experiment on the data in [6,7] as our first test data for the compare of performance. However, after discussing with Zhengyu Wang, we were told he used to ask Prof. Jianyang Zeng where we can download the data. However the answer is nowhere. We were suggested by Lingyu Wei and Zhengyu Wang to generate data randomly.
In other words, the structure is generated through a random walk and whose interaction frequency is obtained through the reciprocal to the distance, plus a random Gaussian perturbation. The code is implemented in *random_structure.m*.

```
function [c,p] = random_structure()
N = 75;
p = zeros(3, N);
for i = 2 : N
    p(:,i) = p(:,i-1) + normrnd(0, 2, [3,1]);
end
plot3(p(1,:),p(2,:),p(3,:),'b*-');
c = zeros(N,N);
M = 10;
sig = 10;
for i = 1 : N
    for j = 1 : N
        if (i~=j)
            c(i,j) = M / norm(p(:,i)-p(:,j), 'fro');%+randn();
        end
    end
end
```

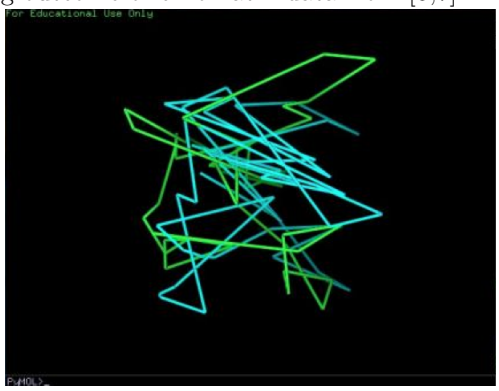
Apply MLT to Model 3D Chromatin Structures [Dataset] continued on next page...

In the next section, we will give the 3D embedding outcome for real data appeared in [6, 7], as illustrated in our research proposal. Then we will give the outcome of a simple empirical study on simulated data generated by the above function.

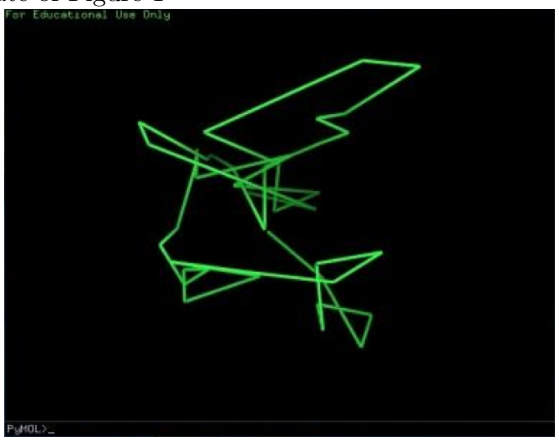
Experiment

Real data experiment:

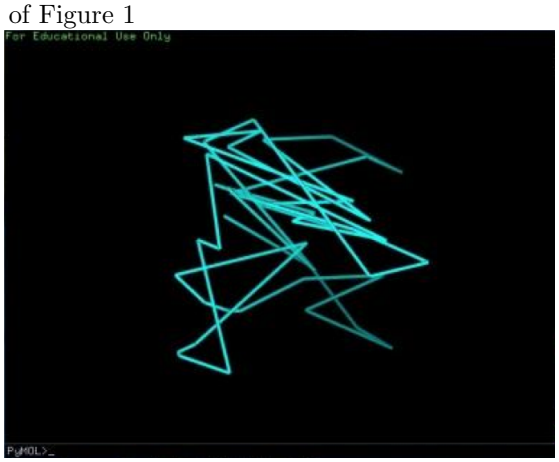
- Figure 1. The aligned embedding outcome of chromatin data from [6,7]



- Figure 2. Undifferentiated State of Figure 1

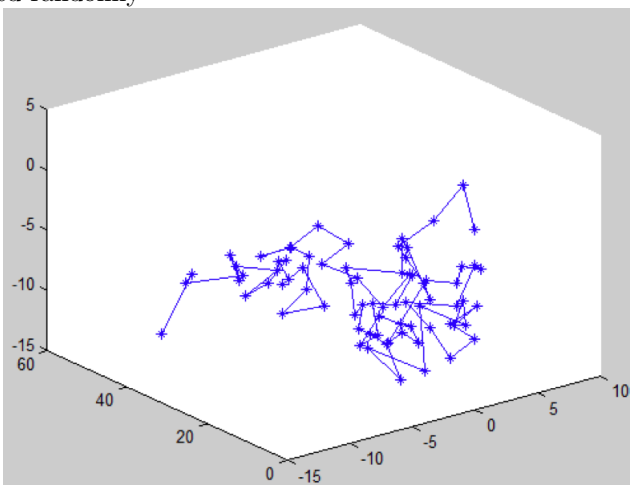


- Figure 3. Differentiated State of Figure 1

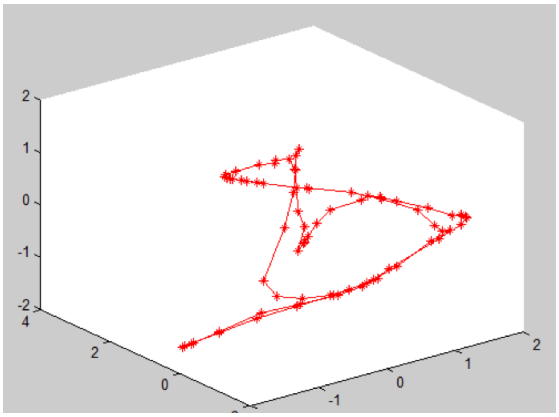


Random data experiment:

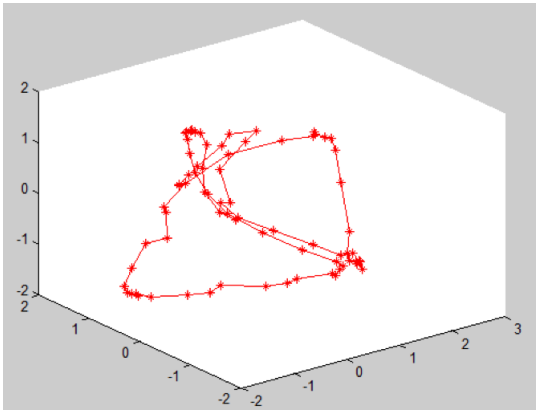
- Original structure generated randomly



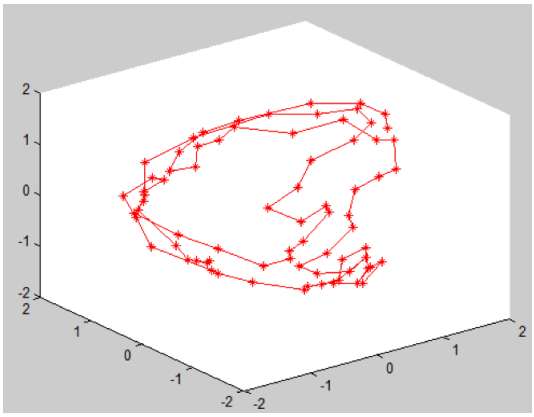
- Embedding with parameter $k = 3$



- Embedding with parameter $k = 7$



- Embedding with parameter $k = 15$



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. Chromosome
4. CS NYU
5. 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.