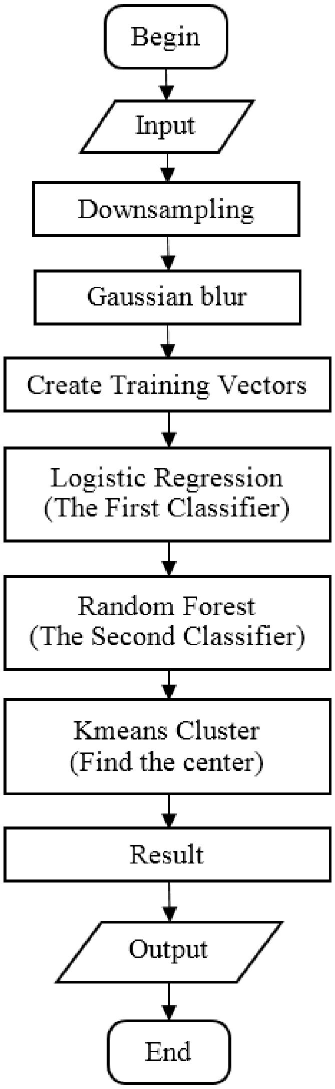
Computational Biology : Report for Project 1

Haoyu Zhao 2016012390
Lukai Li 2015012202
Yi Dai 2015012204

March 13, 2017

1 Introduction

Flow Chart:



2 Methods and Algorithms

2.1 Denoising

Before creating the training dataset, we take some steps to denoise the origin microgragh s. First, we downsample the pictures with a factor two by replacing every 2 2 grid with the average grey value of the pixels in it, so that the high-frequency noise can be decreased. Second, we run a Gaussian blur with $sigma = 4$ in the downsampled pictures.

2.2 Create Training Dataset

We choose first 30 micrographs in the training set as training pictures, and create training dataset from them in the way as following. According to the .star documents, the center of each particle in the training pictures is available, so pixel regions containing particles can be chosen as a right sample. Without manual feature selection, we use 10000-dimentional vectors to represent the 100 100 panel which has a particle in the center, plus little fluctuation. Those make up the positive training dataset. In addition, we randomly choose other panels of the same size per micrographh, and compute the distance with the centers of the particles to determine whether it is a positive sample or not. An image has 400 panels in total. Those make up the whole training dataset.

2.3 Logistic Regression (Classifier 1)

After getting the training vectors of two classes, we do a Logistic regression on it:
Input vectors have the form $X^i = (x^i_1, x^i_2, ..., x^i_{10000})$.
Next, we run the logistic regression to do the first classification.
There are 16000 training examples and we use the first 12000 to train and the last 4000 to test the performance. It comes out that the performance is between 0.85 to 0.89.
Among the 1800 particles and 2200 non-particles, the classifier predict about 2000 particles, which means that the classifier predicts many non-particles into particles.
We apply the sliding panel in the first 5 images in the test set an find that the recall of the first 5 images is roughly 0.98 or 0.99 or 1.0, which means that for almost all the particles, there will be a window near to the center that is predicted positive by the first classifier, but the precision is about 0.23 to 0.25, which means that the classifier select many non-particles as particles or select too far from the center of the particles.

2.4 Second Classifier

To select the particles from positive output of the first classifier, we generate the training data of the second classifier from the output of the first classifier. The training data of the second are all predicted to be the particles by the first classifier, and roughly half of them are partiles, others are noise. We use a random forest classifier to do the classification. We also ues a small set of the data to test the performance. After the second classifier, the recall is also high for the first 5 testing images but the precision becomes 0.47, 0.52, 0.37, 0.42 and

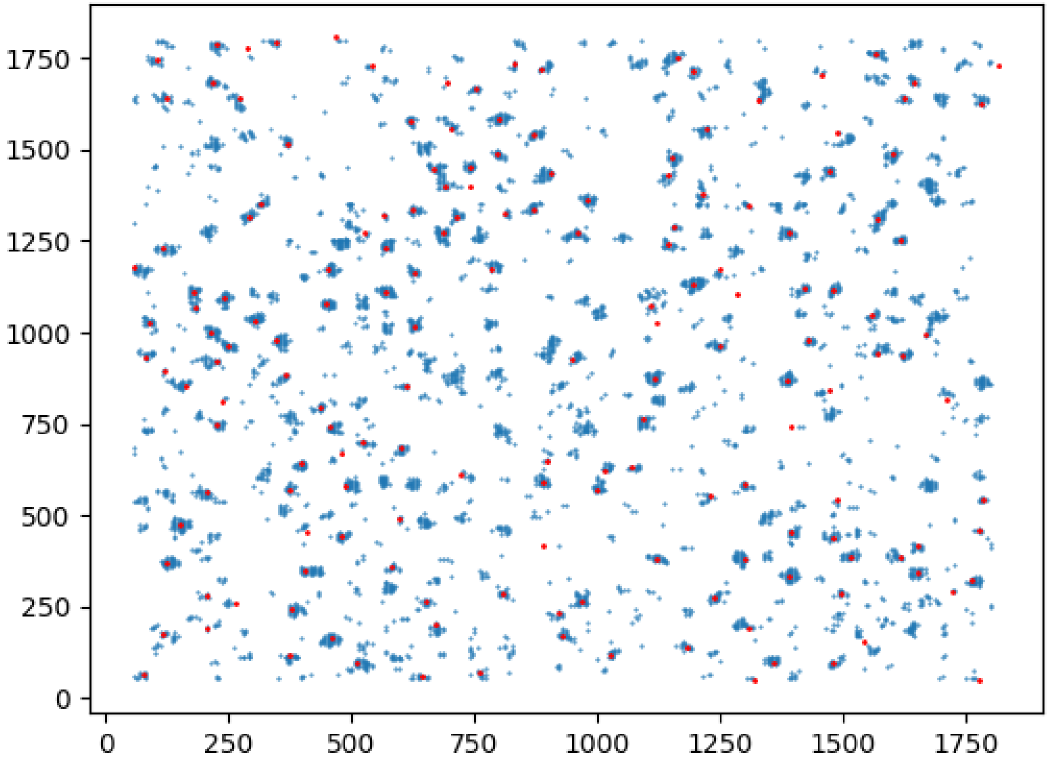
0.51.

2.5 Third Layer

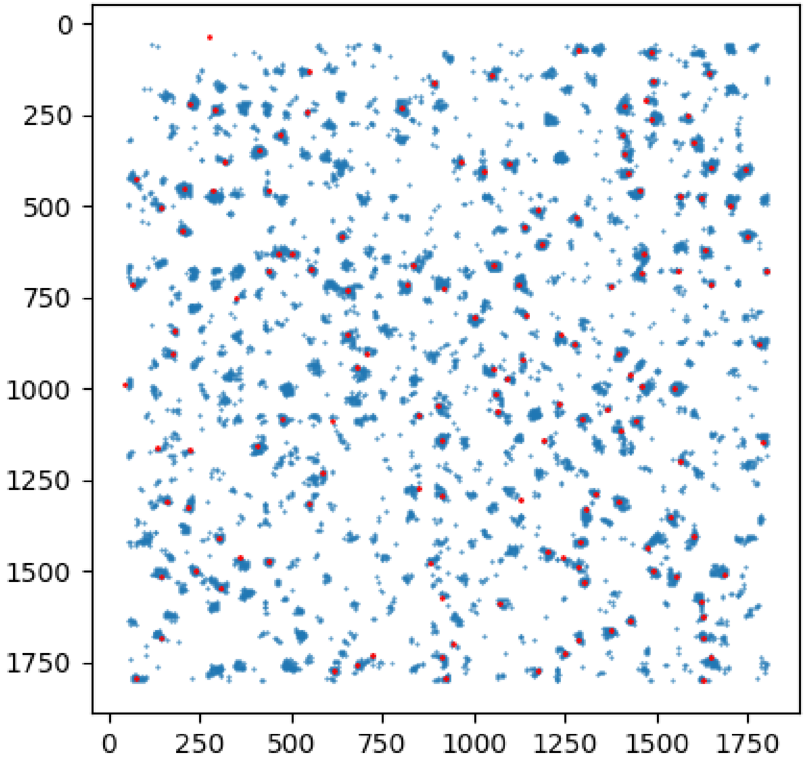
This layer aims to select the center of the cluster of the prediction. After 2 classifiers, many of the particles are selected by the sliding panels and many of the panels are actually containing the same particle. This layer aims to select the particle from many positive outputs. Because we do not have ideas first, we try to use the clustering algorithm to find the center of a cluster of positive prediction. After trying some of the values, we use 350 clusters.

3 Final Result and Performance Evalution

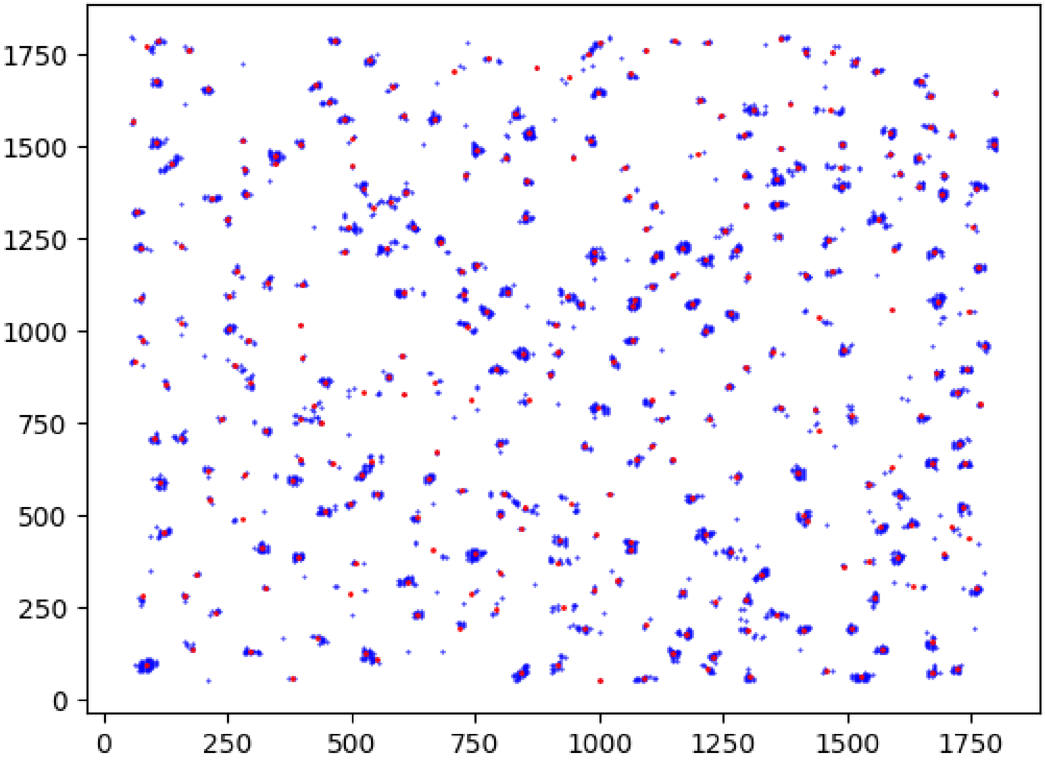
After the training of the second classifier, we do a evaluation, the precision is 0.40 and the recall is 0.82. After the clustering, the precision is 0.40 and the recall is 0.68.
There are some graphs that show the result after the second classifier.
The red point is the center of the particles in the image, the blues are the centers of the positive panels after the second classifier.
The following figure comes from the image 142.



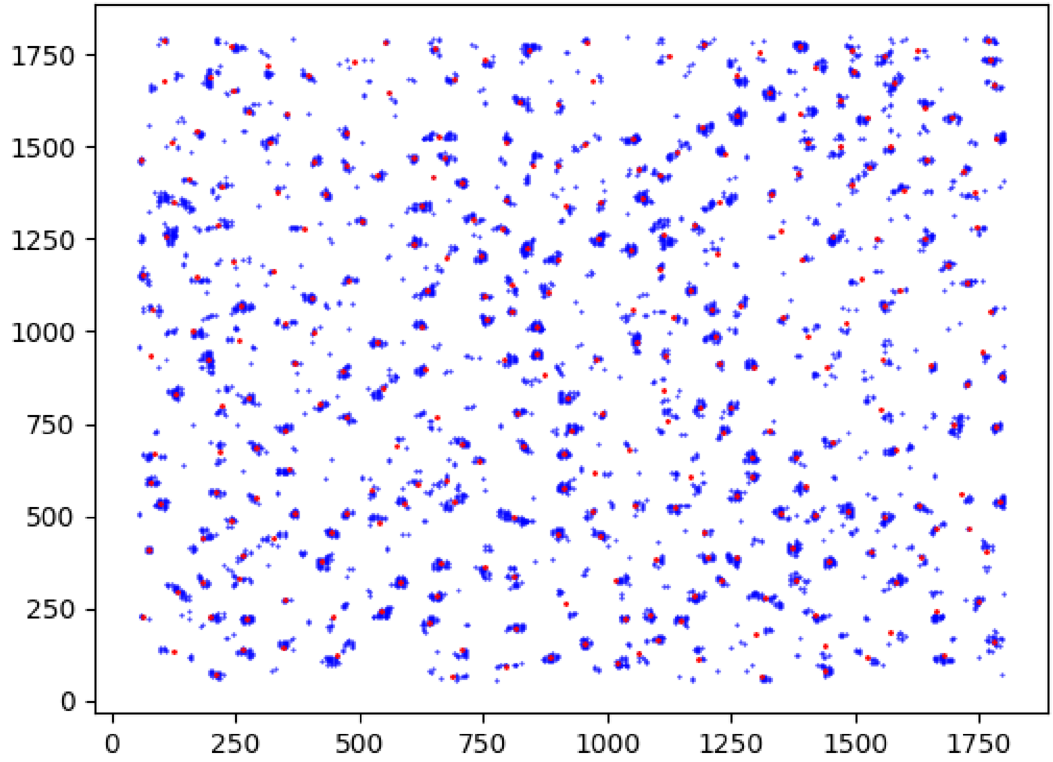
The following figure comes from the image 153.



Below are some of the results after the classification, the red is the center chose by the KMeans algorithm.
The image 150:



The image 157:

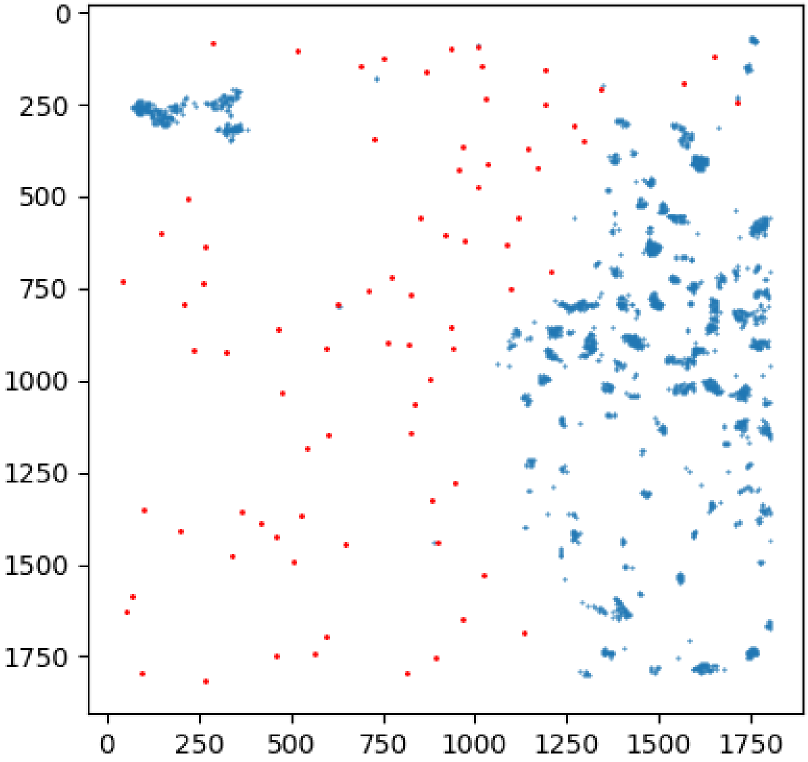


The above accuracy is tested only once, the .star file is in the folder 'centers_firsttry'. Later we do a second try and we found that our accuracy decrease, because the image 161 predict much worse than the first time, recall and precision are all about 0.10, and the final recall is 0.64 and the final precision is 0.39. The .star file of the second try stores in the folder 'centers_secondtry'. I do not know why there is so much prediction difference between the first try and second try on the image 161, maybe it is because the algorithm depends on randomization a lot and is not very stable.

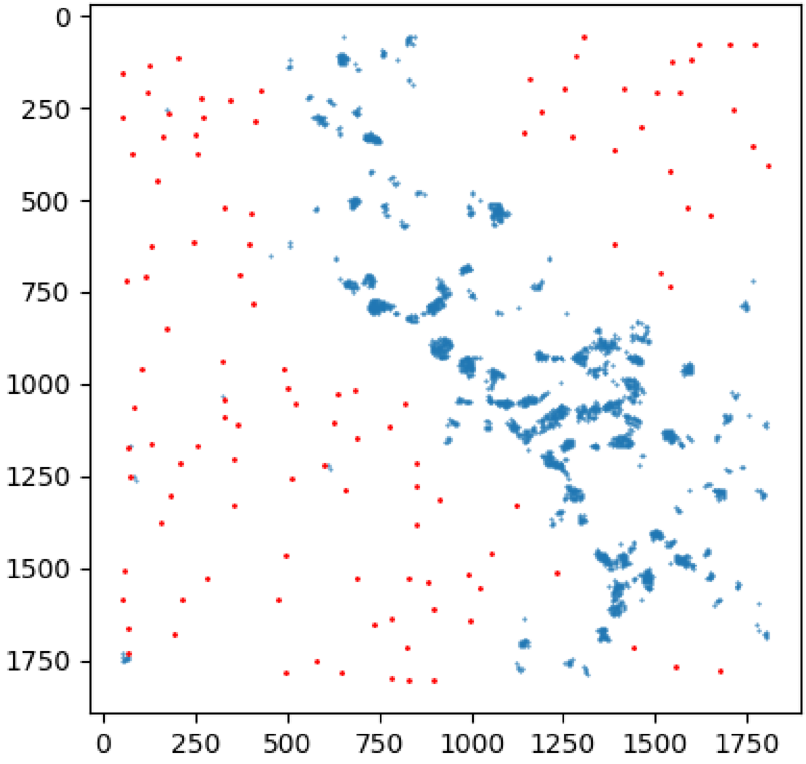
4 Discussion

We do not surprise when we find the low recall after the clustering, but we are surprised when we find the precision after the second classificaiton. We see the precision and recall in each graph and we find some outliers.

The following is 151 image.



The following is 159 image.



These figures have so obvious contamination and has recall and prediction near to 0. There are other figures that contains contaminants, but has recall near to 0.6 or 0.4.

From the result we find that the contaminants matter a lot, which seriously deteriorate the precision. Then we review the algorism and draw a conclusion that we shouldnt have normalized the vectors before Logistic regression. We should have keep the difference of average values of gray, so that the contaminants can be better separated from the particles.

There are another shortcoming in our algorism. We did not do the denoising well. We may learn the method of Bayesian wavelet downsampling and do it instead of average downsampling the next time.

As we didnt do any feature selection manually, we may change the way to select feature according to the knowledge of the target particle.

And at last, Haoyu Zhao found that he made a mistake in the logistic regression code. He forgot to add the constant element in the regression model.

Above are some thinkings and shortcomings in this project. We will talk more about our thinking in Monday's in-class presentation.

5 Contribution of Each Member

- Yi Dai: Reading and Concluding the papers referenced in the document project 1, transfer the information about classifier to the group; Modifying and writing some parts the report with LaTeX, do the ppt of the presentation.
- Lukai Li: Employing the downsampling and Gaussian blur to denoise the micrographs; reding the papers to provide ideas. Writing parts of the report, do the ppt of the presentation.
- Haoyu Zhao: Writing all the Python code including generating data, implementation and testing, choosing algorithms to do the classifying and clustering and writing the related parts of the report, coming up with the downsampling idea together with Lukai Li.

6 Reference

[1] Robert Langlois and Joachim Frank. A clarication of the terms used in comparing semi-automated particle selection algorithms in cryo-em. Journal of Structural Biology, 175(3):348 352, 2011.

[2] Robert Langlois, Jesper Pallesen, Jordan T. Ash, Danny Nam Ho, John L. Rubinstein, and Joachim Frank. Automated particle picking for low-contrast macromolecules in cryo-electron microscopy. Journal of Structural Biology, 186(1):1 7, 2014.

[3] C.O.S. Sorzano, E. Recarte, M. Alcorlo, J.R. Bilbao-Castro, C. San-Martn, R. Marabini, and J.M. Carazo. Automatic particle selection from electron micrographs using machine learning techniques. Journal of Structural Biology, 167(3):252 260, 2009.