Exploring Alternative Methods of PolyCRACKER for Subgenome Separation

Hongchuan Wang

March 2020

Introduction

Polyploidation is the the process of two or more genomes merging in a cell. It improves the adaptibility of organisms under stressful conditions[6]. These organisms include many crop plants, which human relies on for basic nutrients.Thus it is important to understand the process of polyploidation. And to be able to do that, we need to isolate subgenomes within a polyploid genome and find their origin.

There have been many attempts to solve the subgenome isolation problem. Many of these attempts use general genome characteristics as a "genome signature" to identify and seperate DNA sequences from different species. However, in some cases, the subgenomes has high level of similarity and using these features makes it challenging to correctly distinguish subgenomes from different origin.

PolyCRACKER,[4] is an unsupervised approach that utilizes repetitive DNA segments that act as the molecular barcodes. Each repetitive DNA segment is called k-mer, and the unique k-mers are defined by its length k, and the ordering of the DNA bases. Figure 1 show a high level view of their method.

1. First they generate fragmented contigs of equal size, then they use "semi-supervised" approach to assign labels, where they have labels for some contigs, then they use the relative position of the contigs in the genome origin to assign labels to unclassified contigs.
2. Identify k-mers that represent the contigs (The features of contigs), now the data set is obtained
3. Apply PCA to project the data into lower dimension space.
4. Calculate pairwise similarity measure, and use this similarity measure to construct graph.
5. Apply spectral clustering, contigs are grouped into categories.
6. Find differential k-mers from each cluster.
7. The identified differential k-mers can be used to recruit unclassified k-mers
8. Repeat step 6 and 7 until not needed.

The polyCRACKER method is able to achieve high accuracy of identifying labels of contigs. However, this could be due to the fact that they have used contigs that contain rich features which makes them highly identifiable. It is quite possible that with short read sequencing data, the performance of polyCRACKER would drop dramatically. In that scenerio, would it be possible that some alternative unsupervised learning methods can be used to improve the performance?

For this project, I aim to explore alternatives of the unsupervised learning approach that was implemented in the polyCRACKER and compare the performance of the different methods using short read sequencing data provided by Gillian Reynolds.

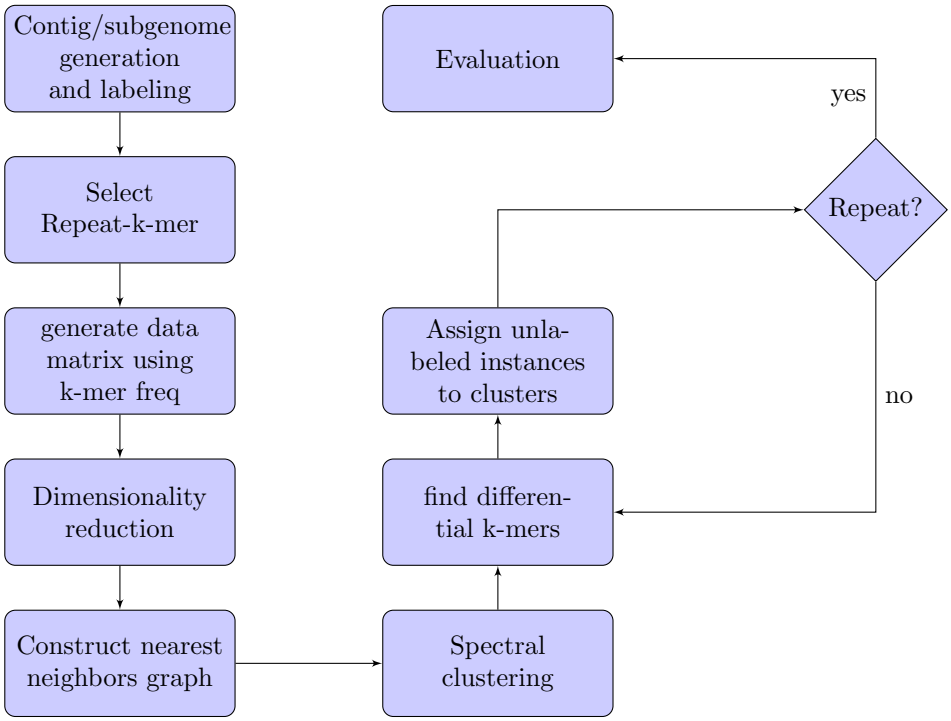

Figure 1: Experiment Procedure

Proposed approach

In the polyCRACKER experiment setup for their unsupervised learning model, no much explanations were given on some of the choices they made.

1. For dimensionality reduction, they used a Kernel PCA with a cosine kernel.
2. In the graph construction step, it is not clear what pairwise similarity measure they use.
3. They chose nearest neighbor graph.
4. And most important all, they didn't provide any reasoning on why they went with spectral clustering. It does not have to be graph based clustering if they chose not to do graph construction.

For 1, my first choice to select the Gaussian kernel for dimensionality reduction. It is quite possible that the data provided by Gill will not containing lots of features thus in this case the PCA step can be removed from the experiment. However, in case PCA is needed, Gaussian kernel PCA will be implemented. Standard PCA will also be implemented for comparison.If time permits, polynomial kernel will also be implemented.

For 2, I will calculate the pairwise Euclidean distance and convert it using Gaussian function: G(x) = $e^{-(x-\mu)^2/2\sigma^2}$

For 3, there are 3 options listed in [7]:

1. *ϵ-neighborhood graph* - points with pairwise distance less than ϵ is connected
2. *k-nearest neighbor graph* - as the name suggests, connect k-nearest neighbors. Although the graph created is directed because neighborhood relationship is not symmetric. To make the graph undirected, either simply ignore the direction of the edge, or remove edges that are only going on direction.
3. *fully connected graph* - fully connected graph, connect all pair of points that has positive similarity. One example of similarity function is the Gaussian similarity function.

I will implement the fully conntected graph, considering the fact that it does not lose any information during graph construction. It should help in the case where in the data set there is not enough features.

For 4, I plan to implement K-means, Agglomerative Hierarchical clustering, DBSCAN as they are the typical examples of the representative-based clustering, hierarchical clustering and density based clustering respectivaly.

contig ID	3-mer 1 freq	3-mer 2 freq	...
contig 1	50	40	...
contig 2	...	...	...
contig n	...	...	...

Table 1: k-mer frequency table

Here is the description of the approach I will be taking:

1. I will use dataset kindly provided by Gillian Reynolds. Data set will be preprocessed in the form of Table 1.
2. Unsupervised learning potion of the polyCRAKER method will be implemented from scratch, data preprocessing portion is not implemented. And the amplication phase (where differential-kmer are extracted and used to assign labels) will not be implemented either. The idea of the amplication phase being: after clustering with initial data, a set of differential k-mers can be extracted that can be used to differentiate contigs that belong to different clusters. If contigs with unknown label is given, the cluster assignment can be done by comparing the k-mers of the unclassified contigs, and the differential k-mers. The point of having the amplication phase, (I assume) is with the amplication phase, after the initial clustering is done, no more clustering is needed for contig cluster assignment, which reduces the computation cost. However, it may not be necessary for this study, because the amplication phase is based on the performance of the spectral clustering. Poor clustering performance would lead to inaccurate cluster assignment in the amplication phase. For this comparative study, I will only focus on the performance of different clustering algorithms. The amplification phase can be used after any of the clustering algorithms that gives high performance. This polyCRACKER approach will be used as the baseline model.
3. Modification of polyCRAKER is made to allow customization of the options discussed in the beginning of this section.
4. K-means, agglomerative hierarchical clustering and DBSCAN will be implemented.
   For K-means, I will implement the K-means++ algorithm [1] where the selection of initial centrods are optimized.
   For agglomerative, list of linkage options will be used: single, complete, group average, mean distance, and Ward.

   For DBSCAN[3], ϵ will be chosen based on the range of features. Number of minimum neighbors will be chosen based on number of instances in the data set. I plan to do grid search for both parameters.
5. If time permits, Kernel K-means, divisive clustering, and DENCLUE[2] will be implemented.
6. For all distance measure, I will use Euclidean distance.

Everything will be implemented from scratch as it gives the benefit of fully customization. The clustering algorithms will be tested and visually validated first using the Fundamental Clustering Problems Suite (FCPS)[5].

Validation

Given that the dataset provided will be labeled, I will use external measures: Purity and Maximum matching and F-measure[8].For each dataset provided, all the measures wil be calculated for each algorithm and each clustering hyperparameter setting.

References

[1] Sergei Vassilvitskii David Arthur. k-means++: the advantages of careful seeding. SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, pages 102–1035, 2007.

[2] Alexander Hinneburg and Hans-Henning Gabriel. Denclue 2.0: Fast clustering based on kernel density estimation. In Michael R. Berthold, John Shawe-Taylor, and Nada Lavrač, editors, Advances in Intelligent Data Analysis VII, pages 70–80, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[3] Jörg Sander Xiaowei Xu Martin Ester, Hans-Peter Kriegel. A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), pages 22–231, 1996.

[4] John P. Vogel Sean P. Gordon, Joshua J. Levy. Polycracker, a robust method for the unsupervised partitioning of polyploid subgenomes by signatures of repetitive dna evolution. BMC Genomics, 20(580), 2019.

[5] Alfred Ultsch. Fundamental clustering problems suite (fcps), 01 2005.

[6] Yves Van de Peer, Eshchar Mizrachi, and Kathleen Marchal. The evolutionary significance of polyploidy. Nature Reviews Genetics, 18(7):411, 2017.

[7] Ulrike von Luxburg. A tutorial on spectral clustering. Statistics and Computing, 17:395–416, 2007.

[8] Mohammed J. Zaki and Jr. Wagner Meira. Data Mining and Analysis: Fundamental Concepts and Algorithms. Cambridge University Press, May 2014.