Estimating Recombination Rate in Satellite DNA using Machine Learning Models

Anya Greenberg, Amanda Larracuente
University of Rochester, Department of Biology

Project Statement

Using information about the distribution of copy number for a repetitive DNA sequence in a simulated population, recombination rate was inferred using machine learning techniques. As this research was exploratory in nature, analysis was done using unsupervised learning (clustering) and supervised learning (regression, classification) methods.

Data Collection

The Stephan Model

For analysis, datasets were simulated using Stephan's model of satellite DNA evolution [2]. In this model, each generation was simulated via a 4 step process:

1. Sampling and selection of gametes (random mating)
2. Recombination (exchange of copies)
3. Survival check (fitness function: $w_i = 1 - s(i-1)$, where i = copy number)
4. Generation of offspring (multinomial sampling)

The simulation took parameters:

- Population size (2N)
- Initial copy number
- Selection coefficient (per copy)
- Exchange rate (per cluster per generation per pair of chromosomes)
- Generation time

Program Design

For the creation of datasets, a program (written by Songeun Lee, a former lab member) was used. The program took parameters similar to the simulation and output normal distribution statistics (mean, standard deviation), gamma distribution statistics (alpha, beta), as well as median, mode, and p-value for the Kolmogorov-Smirnov test of probability distribution comparison.

Datasets

Analysis was done on 3 different datasets using various methods of specifying parameters. For all methods these parameters remained constant: population size=5000, selection coefficient=1.0×10^{-7}, generation time=1000. The variations in the methods were as follows:

1. constant initial copy number=500, exchange rate randomized from gamma(2,10)
2. random initial copy number, exchange rate randomized from gamma(2,10)
3. random initial copy number, exchange rate held constant for each generation

Conclusions

- Regression was unsuccessful in predicting recombination rate
- Classification showed promise in predicting clusters
- k-Means and Birch clustering methods were effective at separating data based on beta value from the gamma distribution
- Clusters were not informative about exchange rate

Future Direction

- Look into how classification works with more precision and what it tells us about recombination rate
- Look into how more complicated evolution models affect results
- Look into other metrics that can be used in machine learning models
  - ex. repeat sequence instead of just copy number

References

Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

Stephan, W. (1986). Recombination and the evolution of satellite DNA. *Genetical Research, 47*(3), 167-174.

Methods: Supervised Learning

Basic algorithm for supervised learning

1. Acquire dataset
2. Split dataset into 4 subsets
   - training - attributes, labels
   - testing - attributes, labels
3. Use the training datasets to train a model
4. Use the testing attributes to predict labels
5. Evaluate the performance of the model by comparing predicted labels to testing labels

Selected Models

Different prediction models work better with different types of data depending on the data's shape and distribution. As such, I chose 3 different models in order to encompass the different possibilities. The selected models were k-Nearest Neighbors, Multinomial Naive Bayes, Decision Trees, and Random Forest. The parameters used for each model were mean, alpha, and beta. For regression, the models were predicting exchange rate. For classification, the models were predicting the cluster label assigned during unsupervised learning.

k-Nearest Neighbors is an instance-based learning model and is one of the simplest and easiest to implement. It's also a good model to start with if conducting exploratory analysis.

Multinomial Naive Bayes is a model based on Bayes theorem and has similar assumptions and execution to Approximate Bayesian Computation, which was previously attempted to predict recombination rate. So, it would be interesting to see how this model compared to others.

Decision Trees are a very explanatory models that builds a tree containing attributes as nodes and decision criteria at each branch. Since these models produce an easy to understand explanation for results, a positive evaluation of this model would help in understanding the information available in the distribution of copy number.

Random Forest models utilize multiple, different decision trees and select a label based on the majority result of those trees. This model is also very explanatory and easy to understand. However, it tends to take more time to complete.

Regression vs. Classification

Supervised learning has two subcategories of predictive models: regression and classification. Regression is used for predicting continuous labels, whereas classification is used for predicting discrete, binary, ordinal, or categorical labels. For this project, the exchange rate was the label while mean, alpha, and beta were selected as attributes. Because exchange rate is inherently a continuous variable, regression models were made. However, because they showed promising results, classification was also attempted. All of the selected models had both regression and classification versions except for Multinomial Naive Bayes.

Methods: Unsupervised Learning

Basic algorithm for unsupervised learning

1. Acquire dataset
2. Feed dataset into a clustering algorithm
   - most algorithms group data based on a similarity or distance metric
3. Retrieve inferred clusters and labels
4. Evaluate cluster labels using…
   - metrics that compare homogeneity/ heterogeneity of clusters
   - metrics that compare cluster labels to some "true labels" which are known
   - cluster labels as the labels in a supervised learning model

Selected Models

Different clustering methods work better with different types of data depending on the data's shape and distribution. As such, I chose 4 different methods in order to encompass the different possibilities. The selected models were k-Means, DBSCAN, and Birch. The parameters used for each method were mean, alpha, and beta.

k-Means is a centroid-based clustering method. Given a set number of clusters to form, this algorithm divides the data and calculates a centroid for each cluster. A centroid is the mean of all points in that cluster. It then reassigns each data point to the cluster whose centroid is closest, distance-wise. Lastly, it updates the centroid of each cluster and repeats until no more updates were made. k-Means works well with large samples and equal sized clusters.

DBSCAN is a density based clustering method. It views clusters as areas of high density surrounded by areas of low density. Because of this, it works well with different sized and different shaped clusters, unlike k-Means.

Birch is an agglomerative hierarchical clustering method. Agglomerative hierarchical methods are considered bottom-up. They start by considering each data point a separate cluster. Through iterations, they successively group data points that are nearest to each other - forming a structure similar to a phylogeny tree. Birch, in particular, is know for being good at handling noise in the data and completing fairly quickly on large datasets.

Results

Supervised Learning: Regression

The most common metric for evaluation the performance of a regression model is the coefficient of determination (R^2). The coefficient of determination represents the proportion of total variation explained by the model. All models presented with a negative R^2, indicating that all models did not follow the trend of the data.

kNN	Decision Tree	Random Forest
-0.29	-0.20	-0.12

Supervised Learning: Classification

For evaluating the performance of classification models, there are several metrics to consider:

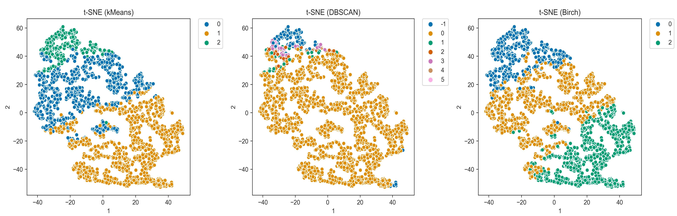
- accuracy measures how many predictions were correct
- precision measures the model's ability to not label a negative sample as a positive
- recall measures the model's ability to find all the positive samples
- F1 is another measure of the model's accuracy accounting for precision and recall
- AUC measures the true positive rate against the false positive rate

For all these metrics, a value closer to 1 means the model performs better in that aspect. The table below shows the metrics calculated when using k-Means clusters as the labels. All models were similar and presented with very high scores for all metrics, indicating the models were great at predicting exchange rate when fitted with a training dataset

Prediction	Accuracy	Precision (P)	Recall (R)	F1	AUC
kNN	0.99	0.99	0.99	0.99	1
Naive Bayes	0.88	0.82	0.88	0.84	0.92
Decision Tree	0.99	0.99	0.99	0.99	0.98
Random Forest	0.99	0.98	0.99	0.98	0.99
		TP/(TP+FP)	TP/(TP+FN)	P*R/(P+R)	

Unsupervised Learning

Algorithm	Silhouette	Calinski-Harabasz	Davies-Bouldin
kMeans	0.57	4427	0.54
DBSCAN	0.39	343	1.57
Birch	0.54	3711	0.55
	higher=better		lower=better



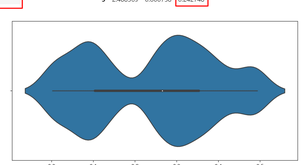
Cluster Composition

Since two of the clustering methods performed fairly well, it would be beneficial to understand how each of the clusters formed by each method differ - to figure out what separates one cluster from another. The tables to the right show the average alpha, beta, and exchange rate per cluster for each method. From the tables, it looks like the major difference between clusters for all methods lies in the beta value, a parameter for the distribution. However, it does not seem that the clusters are informative about the exchange rate. The mean exchange rate for all clusters for all methods lies between 0.2 and 0.3. Shown in the violin plot of the distribution of exchange rate, the mean exchange rate of the whole dataset lies within this range as well. This indicates that the clusters do not provide information about the exchange rate.

km	alpha	beta	exchg_r
0	2.513132	0.003690	0.235601
1	2.527040	0.000986	0.268574
2	2.914524	0.001873	0.223703
3	2.449350	0.000610	0.289307

db	alpha	beta	exchg_r
-1	2.470480	0.001668	0.288213
0	2.684805	0.002614	0.233858
1	2.475293	0.000911	0.275524
2	2.474803	0.000863	0.255058
3	2.446344	0.000791	0.279407
4	2.491049	0.000833	0.270017
5	2.488369	0.000758	0.242748

b	alpha	beta	exchg_r
0	2.511986	0.000913	0.272592
1	2.851191	0.001965	0.221386
2	2.490782	0.004045	0.241944



Unsupervised Learning

The t-SNE plots to the left show a visualization of the cluster separation and composition for each clustering method. Besides visualizing the clusters, there are also several metrics to analytically evaluate the clustering methods.

- Silhouette coefficient measures how homogeneous the clusters are (higher is better)
- Calinski-Harabasz score is the ratio of the intra-cluster variance to the inter-cluster variance (higher is better)
- Davies-Bouldin score measures the average similarity of clusters (lower is better)

From the plots and the metrics, it is shown that both k-Means and Birch had more defined clusters than DBSCAN with k-Means being slightly better.