

%copyright Charles Davi 2022

%=====

===
%VECTORIZED GENOMICS COMMAND LINE

%=====

clear all
clc

file = "/Users/charlesdavi/Desktop/Datasets/Genetics/NIH/HumanMtDNA/MTDNA.txt";
dataset = csvread(file);

dataset = unique(dataset,'rows');

num_rows = size(dataset,1);
N = size(dataset,2)-1;

%=====

===
%RUNS NEAREST NEIGHBOR

%=====

tic;
[nearest_neighbors match_count] = Genetic_Nearest_Neighbor_Fast(dataset,N);
toc

%=====

===
%DISPLAYS ACCURACY AS A FUNCTION OF CONFIDENCE

%=====

predicted_class = dataset(nearest_neighbors,N+1);
actual_class = dataset(:,N+1);
error_vector = predicted_class != actual_class;

matches = unique(match_count);
num_levels = size(matches,2);

num_entries_vector = [];

for i = 1 : num_levels

 current_level = matches(i);

```

x = find(match_count >= current_level);
a = error_vector(x);
num_entries = size(a,1);
accuracy(i) = 1 - sum(a)/num_entries;

num_entries_vector(i) = num_entries;

endfor

figure, plot(accuracy)

%=====
===
%CALCULATES ACCURACY FOR EACH POPULATION
%=====
===

%Kazakh
x = find(dataset(:,N+1) == 1);

predicted_class = dataset(nearest_neighbors(x),N+1);
actual_class = dataset(x,N+1);
error_vector = predicted_class != actual_class;

num_entries = size(x,1)

accuracy = 1 - sum(error_vector)/num_entries

predicted_class

%Nepal
x = find(dataset(:,N+1) == 2);

predicted_class = dataset(nearest_neighbors(x),N+1);
actual_class = dataset(x,N+1);
error_vector = predicted_class != actual_class;

num_entries = size(x,1)

accuracy = 1 - sum(error_vector)/num_entries

predicted_class

%Iberian Roma
x = find(dataset(:,N+1) == 3);

```

```
predicted_class = dataset(nearest_neighbors(x),N+1);
actual_class = dataset(x,N+1);
error_vector = predicted_class != actual_class;
```

```
num_entries = size(x,1)
```

```
accuracy = 1 - sum(error_vector)/num_entries
```

```
predicted_class
```

```
%Japan
x = find(dataset(:,N+1) == 4);
```

```
predicted_class = dataset(nearest_neighbors(x),N+1);
actual_class = dataset(x,N+1);
error_vector = predicted_class != actual_class;
```

```
num_entries = size(x,1)
```

```
accuracy = 1 - sum(error_vector)/num_entries
```

```
predicted_class
```

```
%Italian
x = find(dataset(:,N+1) == 5);
```

```
predicted_class = dataset(nearest_neighbors(x),N+1);
actual_class = dataset(x,N+1);
error_vector = predicted_class != actual_class;
```

```
num_entries = size(x,1)
```

```
accuracy = 1 - sum(error_vector)/num_entries
```

```
predicted_class
```

```
%=====
===
%DISPLAYS THE CLASS DISTRIBUTION FOR EACH POPULATION USING
CLUSTERS
%=====
===
```

```
X = {'Kazakh','Nepalese','lb. Roma','Japanse','Italian'};
```

```
%Kazakh
x = find(dataset(:,N+1) == 1);
```

```

num_items = size(x,1);

K = ceil(min(match_count(x)))
cluster_matrix = zeros(num_rows,num_rows);

for i = 1 : num_rows

    match_matrix = Genetic_Cluster_Algorithm(i, dataset, N);
    cluster_match_count = sum(match_matrix,2);

    cluster_entries = cluster_match_count >= K; %min match count
    cluster_matrix(i,:) = cluster_entries';
    cluster_matrix(i,i) = 0;

endfor

class_distribution = zeros(1,5);

for i = 1 : num_items

    y = x(i); %the current row
    z = find(cluster_matrix(y,:) == 1);

    class_vector = dataset(z,N+1); %the classes in the cluster

    %searches for each class
    for j = 1 : 5

        a = find(class_vector == j);
        num_matches = size(a,1);

        class_distribution(j) = class_distribution(j) + num_matches;

    endfor

endfor

figure, bar(class_distribution)
set(gca,'xtick',[1:5],'xticklabel',X)

%Nepal
x = find(dataset(:,N+1) == 2);
num_items = size(x,1);

K = ceil(min(match_count(x)))
cluster_matrix = zeros(num_rows,num_rows);

```

```

%generates the clusters
for i = 1 : num_rows

    match_matrix = Genetic_Cluster_Algorithm(i, dataset, N);
    cluster_match_count = sum(match_matrix,2);

    cluster_entries = cluster_match_count >= K; %min match count
    cluster_matrix(i,:) = cluster_entries';
    cluster_matrix(i,i) = 0;

endfor

class_distribution = zeros(1,5);

%calculates the distribution of classes over all clusters
for i = 1 : num_items

    y = x(i); %the current row
    z = find(cluster_matrix(y,:) == 1);

    class_vector = dataset(z,N+1); %the classes in the cluster

    %searches for each class
    for j = 1 : 5

        a = find(class_vector == j);
        num_matches = size(a,1);

        class_distribution(j) = class_distribution(j) + num_matches;

    endfor

endfor

figure, bar(class_distribution)
set(gca,'xtick',[1:5],'xticklabel',X)

%Iberian Roma
x = find(dataset(:,N+1) == 3);
num_items = size(x,1);

K = ceil(min(match_count(x)))
cluster_matrix = zeros(num_rows,num_rows);

%generates the clusters
for i = 1 : num_rows

```

```

match_matrix = Genetic_Cluster_Algorithm(i, dataset, N);
cluster_match_count = sum(match_matrix,2);

cluster_entries = cluster_match_count >= K; %min match count
cluster_matrix(i,:) = cluster_entries';
cluster_matrix(i,i) = 0;

endfor

class_distribution = zeros(1,5);

for i = 1 : num_items

    y = x(i); %the current row
    z = find(cluster_matrix(y,:) == 1);

    class_vector = dataset(z,N+1); %the classes in the cluster

    %searches for each class
    for j = 1 : 5

        a = find(class_vector == j);
        num_matches = size(a,1);

        class_distribution(j) = class_distribution(j) + num_matches;

    endfor

endfor

figure, bar(class_distribution)
set(gca,'xtick',[1:5], 'xticklabel',X)

%Japan
x = find(dataset(:,N+1) == 4);
num_items = size(x,1);

K = ceil(min(match_count(x)))
cluster_matrix = zeros(num_rows,num_rows);

%generates the clusters
for i = 1 : num_rows

    match_matrix = Genetic_Cluster_Algorithm(i, dataset, N);
    cluster_match_count = sum(match_matrix,2);

    cluster_entries = cluster_match_count >= K; %min match count

```

```

cluster_matrix(i,:) = cluster_entries';
cluster_matrix(i,i) = 0;

endfor

class_distribution = zeros(1,5);

%calculates the distribution of classes over all clusters
for i = 1 : num_items

    y = x(i); %the current row
    z = find(cluster_matrix(y,:) == 1);

    class_vector = dataset(z,N+1); %the classes in the cluster

    %searches for each class
    for j = 1 : 5

        a = find(class_vector == j);
        num_matches = size(a,1);

        class_distribution(j) = class_distribution(j) + num_matches;

    endfor

endfor

figure, bar(class_distribution)
set(gca,'xtick',[1:5],'xticklabel',X)

%Italy
x = find(dataset(:,N+1) == 5);
num_items = size(x,1);

K = ceil(min(match_count(x)))
cluster_matrix = zeros(num_rows,num_rows);

%generates the clusters
for i = 1 : num_rows

    match_matrix = Genetic_Cluster_Algorithm(i, dataset, N);
    cluster_match_count = sum(match_matrix,2);

    cluster_entries = cluster_match_count >= K; %min match count
    cluster_matrix(i,:) = cluster_entries';
    cluster_matrix(i,i) = 0;

```

```

endfor

class_distribution = zeros(1,5);

%calculates the distribution of classes over all clusters
for i = 1 : num_items

    y = x(i); %the current row
    z = find(cluster_matrix(y,:) == 1);

    class_vector = dataset(z,N+1); %the classes in the cluster

    %searches for each class
    for j = 1 : 5

        a = find(class_vector == j);
        num_matches = size(a,1);

        class_distribution(j) = class_distribution(j) + num_matches;

    endfor

endfor

figure, bar(class_distribution)
set(gca,'xtick',[1:5],'xticklabel',X)

%Japan ex anamolies
x = find(dataset(:,N+1) == 4);

%deletes anamolies
x(1:2) = [];
x(8) = [];

num_items = size(x,1);

K = ceil(min(match_count(x)))
cluster_matrix = zeros(num_rows,num_rows);

%generates the clusters
for i = 1 : num_rows

    match_matrix = Genetic_Cluster_Algorithm(i, dataset, N);
    cluster_match_count = sum(match_matrix,2);

    cluster_entries = cluster_match_count >= K; %min match count
    cluster_matrix(i,:) = cluster_entries';

```



```

    cluster_matrix(i,i) = 0;

endfor

class_distribution = zeros(1,5);

%calculates the distribution of classes over all clusters
for i = 1 : num_items

    y = x(i); %the current row
    z = find(cluster_matrix(y,:) == 1);

    class_vector = dataset(z,N+1); %the classes in the cluster

    %searches for each class
    for j = 1 : 5

        a = find(class_vector == j);
        num_matches = size(a,1);

        class_distribution(j) = class_distribution(j) + num_matches;

    endfor

endfor

figure, bar(class_distribution)
set(gca,'xtick',[1:5], 'xticklabel',X)

```