Table of Contents

```
%Name: David George
%Student Number: 251004930
```

SetUp

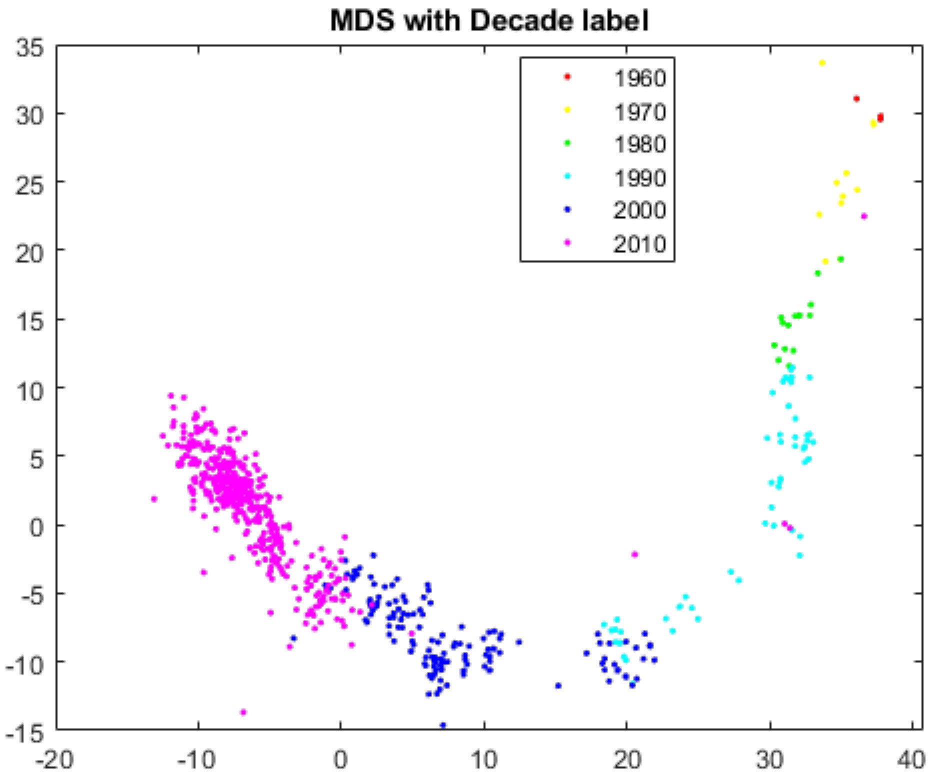
```
T = readtable("h3n2.csv");
T.name = string(T.name);
T.seq = string(T.seq);
T.year = double(T.year);
Decade = [];
for idx = 1:950

    if T.year(idx) <= 1969 & T.year(idx) >= 1960
        Decade(idx) = 1960;
    end

     if T.year(idx) <= 1979 & T.year(idx) >= 1970
        Decade(idx) = 1970;
     end

       if T.year(idx) <= 1989 & T.year(idx) >= 1980
        Decade(idx) = 1980;
     end

        if T.year(idx) <= 1999 & T.year(idx) >= 1990
        Decade(idx) = 1990;
        end

       if T.year(idx) <= 2009 & T.year(idx) >= 2000
        Decade(idx) = 2000;
       end

     if T.year(idx) <= 2019 & T.year(idx) >= 2010
        Decade(idx) = 2010;
     end



end
Decade = Decade.';
T.decade = Decade;
```

Part A

```
   %creating a matrix of the approproate dimmensions to put the
distances
   %in
   %D  = zeros(length(T.seq));



   %iteratre over each pair of sequences of points in the matrix

   %for i = 1:950

         %for j = 1:950

           %Setting the specfic posisiton in the matrix with
corresponding
           % pair-wise distance between points
          % D(i, j) = dist(T.seq(i), T.seq(j));
          % disp("here");
         %end
   %end
```

Part B)

```
MDS  = cmdscale(Matrix, 2);
figure

 %MDS with labels for manufacturer and year number
 gscatter(MDS(:,1), MDS(:,2),(rmmissing(T).decade));
 title("MDS with Decade label");
```



Part C)

Yes the MDS performed in b illustreates antigenic drift of influenza. Atingenic drift results, year after year, in new

```
        %strains of in?uenza virus that “look” di?erent from the
strains
        %of previous years (they are more “distant”. The MDS shows
clear
        %grouping based on decades, they are sepreated and the strains
are
        %changing over time, confirming  antingeic drift.

    function dist = dist(string1, string2)

    %Changing the sequences to char vectors immdeitley, to make
indexing
    %and iterating possible
    char1 = char(string1)
    char2 = char(string2)

    %Taking only the required elements from the sequence
    char1  = char1(100:500);
    char2 = char2(100:500);

    dist = 0;
    %This will iteratore over the length of sequences
    for idx = 1:length(char1)

        %Everytime the sequences are not equal to eachother, increment
 the
        %distance between the two sequences
        if char1(idx) ~= char2(idx)

            dist = dist +1;

        end


    end




    end
```

Published with MATLAB® R2019b