%DEMO Self-organizing map visualization.
% Use SOM Toolbox 2.0, Version 2.0beta, May 30 2002
% http://www.cis.hut.fi/projects/somtoolbox/
clf reset;
clc
echo on
%    ==============================================================
%    DEMO - TRAIN and VISUALIZATION a SOM
%    ==============================================================
% The basic functions for SOM
% som_show - Visualize map.
% som_grid - Visualization with free coordinates.
% som_show_add - Add markers on som_show visualization.
% som_show_clear - Remove markers from som_show visualization.
% som_hits - Hit histograms for the map.
% som_make - Create, initialize and train a SOM
% kmeans_clusters - try and evaluate several k-means clusterings
% som_read_data - Reads a (SOM_PAK format) ASCII data file
% som_recolorbar - Refresh/reconfigure colorbars

pause % Strike any key to read the data and use SOM_MAKE...
clc
%    DEMO DATA
%    ===============
% The data set contructed for this demo consists
% of a subset of the CVE
% encode vectors data
sD = som_read_data('5000VectorData.txt');

pause % Strike any key to use SOM_MAKE...
clc
%    MAP
%    ===============
% Creates, initializes and trains a SOM with selected parameters.
% 'shape' map shape - 'sheet', 'cyl' or 'toroid'
% 'mapsize' do you want a 'small', 'normal' or 'big' map
% 'training' 'short', 'default' or 'long'
% 'tracking' how much to report, default = 1
sM=som_make(sD,'shape','toroid','mapsize','big','training','long','tracking',10);
%sM=som_make(sD,'shape','toroid','mapsize','small','training','short',
'tracking',10);

pause % Strike any key to visualize clusters with distance matrices...
clc
% 1. VISUALIZATION OF CLUSTERS: DISTANCE MATRICES

% Distance matrices are typically used to show the cluster structure of the SOM. They show distances between neighboring units, and are thus closely related to single linkage clustering techniques. The most widely used distance matrix technique is the U-matrix.
% Here, the U-matrix of the map is shown:
figure;
som_show(sM,'umat','all','norm','d');
pause % Strike any key to visualize data responses...
clc
% BEST-MATCHING UNITS (BMU)
% ================
% The BMU of a data vector is the unit on the map whose model vector best resembles the data vector. The BMUs can be calculated using function SOM_BMUS. This function gives the index of the unit.

bmu = som_bmus(sM, sD);
%fist 10 responses
bmu(1:10)

clc
% DATA ON MAP: HIT HISTOGRAMS
% ================
% One can also investigate whole data sets using the map. When the BMUs of multiple data samples are aggregated, a hit histogram results. Instead of BMUs, one can also aggregate for example fuzzy responses.

figure;
som_show(sM,'umat','all');
hits = som_hits(sM,sD);
som_show_add('hit',hits,'MarkerColor','w','Subplot',1);

% "Hits" output: number of records for which a node is "Best Match Unit" (BMU)
figure;
U = som_umat(sM);
Dm = U(1:2:size(U,1),1:2:size(U,2));
Dm = 1-Dm(:)/max(Dm(:)); Dm(find(hits==0)) = 0; % clustering info
som_cplane(sM,Dm);
hold on
som_grid(sM,'Label',cellstr(int2str(hits)),...  'Line','none','Marker','none','Labelcolor','k');
hold off
title('Sample Hits Plot')

pause % Strike any key to cluster the map...

clc
%    CLUSTERING OF THE MAP
%    =====================
%    Visual inspection already hinted that there are clusters in the data.
%    For further investigation, the map needs to be partitioned.
%    KMEANS_CLUSTERS function is used to find an initial partitioning. The plot shows the Davies-Boulding clustering index, which is minimized with best clustering.

% find at most 7 clusters
figure;
[c,p,err,ind] = kmeans_clusters(sM, 7);
plot(1:length(ind),ind,'x-')

% select the one with smallest index and visualize clusters
figure;
[dummy,i] = min(ind);
som_show(sM,'color',{p{i},sprintf('%d clusters',i)});

pause % Strike any key to cluster the map with specified value of k
k = input("Enter value for k: ")

figure;
[codes,basesM,errors]=som_kmeans('batch',sM,k);
som_show(sM,'color',{basesM',sprintf('%d OPT clusters using K-means',max(basesM))});