Table of Contents

Initial Setup

```
rng(0, 'twister'); % For reproducibility
```

Preparing the input data

```
Nodes = readtable('data/Dataset1-Media-Example-NODES.csv');
Edges = readtable('data/Dataset1-Media-Example-EDGES.csv');
disp(Nodes(1:10, :)) % Display the first ten rows
disp(Edges(1:10, :)) % Display the first ten rows
```

```
    id              media              media_type    type_label     audience_size
    ____    ____________________    __________    ___________    _____________

    's01'   'NY Times'              1             'Newspaper'    20
    's02'   'Washington Post'       1             'Newspaper'    25
    's03'   'Wall Street Journal'   1             'Newspaper'    30
    's04'   'USA Today'             1             'Newspaper'    32
    's05'   'LA Times'              1             'Newspaper'    20
    's06'   'New York Post'         1             'Newspaper'    50
    's07'   'CNN'                   2             'TV'           56
    's08'   'MSNBC'                 2             'TV'           34
    's09'   'FOX News'              2             'TV'           60
    's10'   'ABC'                   2             'TV'           23

    from      to         type         weight
    ____    _____    ___________    ______

    's01'   's02'    'hyperlink'    22
    's01'   's03'    'hyperlink'    22
    's01'   's04'    'hyperlink'    21
    's01'   's15'    'mention'      20
    's02'   's01'    'hyperlink'    23
    's02'   's03'    'hyperlink'    21
    's02'   's09'    'hyperlink'     1
    's02'   's10'    'hyperlink'     5
    's03'   's01'    'hyperlink'    21
    's03'   's04'    'hyperlink'    22
```


Published with MATLAB® R2016b

```
nodes = Nodes.id;
from = Edges.from;
to = Edges.to;

[~, ib_from] = ismember(from, nodes);
[~, ib_to] = ismember(to, nodes);
W = sparse(ib_from, ib_to, true, length(nodes), length(nodes));
```

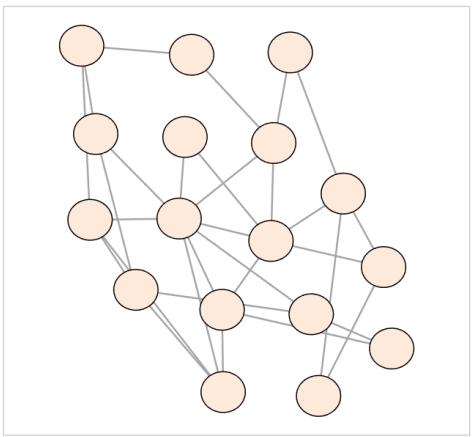
Getting started

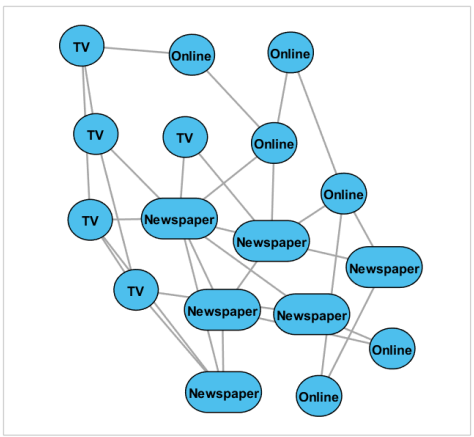
Let's start by visualizing our network with the default options (figure left). Next, we can change the coloring and some node labels to make our plot prettier (figure right).

```
net = networkvisualizer(W);

figure(1);
clf();                                      % Clear the current figure
set(gcf, 'Position', [0 80 1280 480]);      % Setup the figure resolution
subplot(1, 2, 1);
plot(net)

% Set the color of all nodes to light-blue
setNodeColors(net, [0.30 0.75 0.93]);
%
setNodeLabels(net, Nodes.type_label);
setNodeFontSize(net, 9);
setNodeSizes(net, 'auto');
subplot(1, 2, 2);
plot(net);
```





Published with MATLAB® R2016b

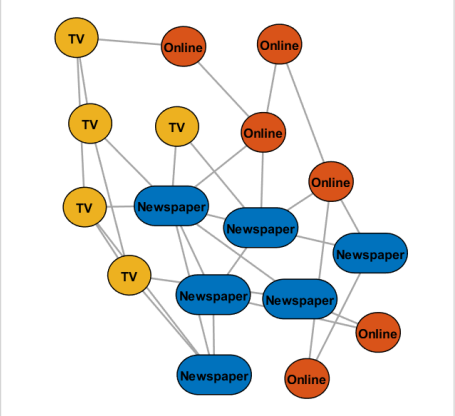
Using node classes

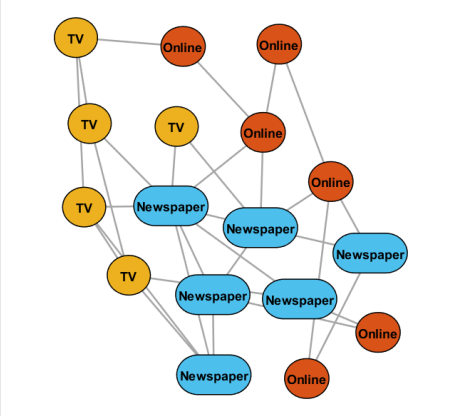
Our nodes represent media channels, thus, each node has a media type, which is either 'TV', 'Online', or 'Newspaper'. We can represent this type of structure by adding a *node class* to our network object, which we name 'Media Type'. By default, when we add a node class for the first time, the nodes are colored according to the provided categories (unless we specify the node colors manually).

Notice that, the default color of the 'Newspaper' nodes is a bit hard to see. Now, we can easily replace its color using the node class 'Media Type' that we added.

```
addNodeClass(net, Nodes.type_label, 'Media Type');
setNodeColors(net, 'default');
subplot(1, 2, 1);
plot(net);

% Set the color of 'Newspaper' nodes to light-blue
setNodeColors(net, [0.30 0.75 0.93], 'Newspaper', 'Media Type');
subplot(1, 2, 2);
plot(net);
```



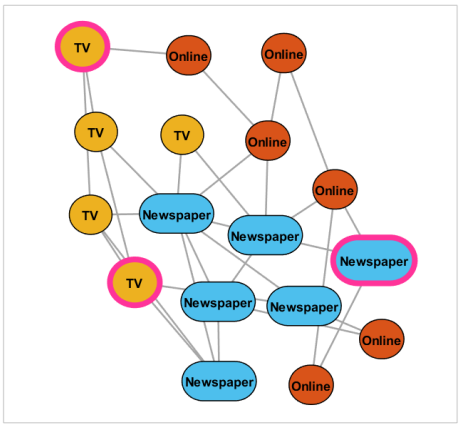


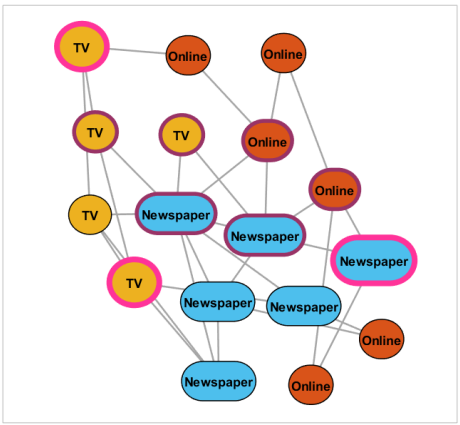
Next, to further annotate the nodes, let's create two new node classes named 'Large Audience' (if the audience size is larger than 40) and 'Medium Audience' (if audience size is between 30 and 40). These classes are essentially binary variables with true or false values. Now, let's show these audience size categories in our network by setting the line widths and line colors of our nodes (pink for large and brown for medium audience size).

```
addNodeClass(net, Nodes.audience_size >= 40, 'Large Audience');
setNodeLineWidth(net, 4, true, 'Large Audience');
setNodeLineColor(net, [1 0.2 0.6], true, 'Large Audience');
setNodeSizes(net, 12, true, 'Large Audience');
setNodeSizes(net, 'auto');
subplot(1, 2, 1);
plot(net);
```

Published with MATLAB® R2016b

```
addNodeClass(net, Nodes.audience_size < 40 & Nodes.audience_size >= 30, 'Medium Audience');
setNodeLineWidth(net, 2.75, true, 'Medium Audience');
setNodeLineColor(net, [0.6 0.2 0.4], true, 'Medium Audience');
subplot(1, 2, 2);
plot(net);
```





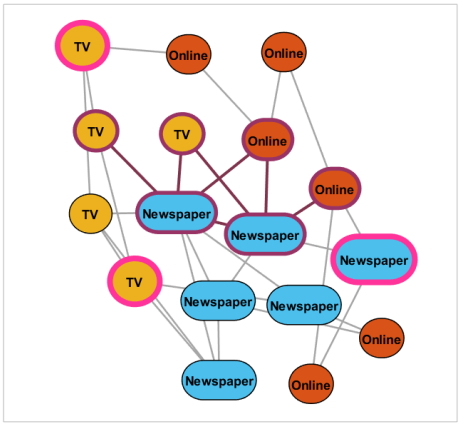
Using edge classes

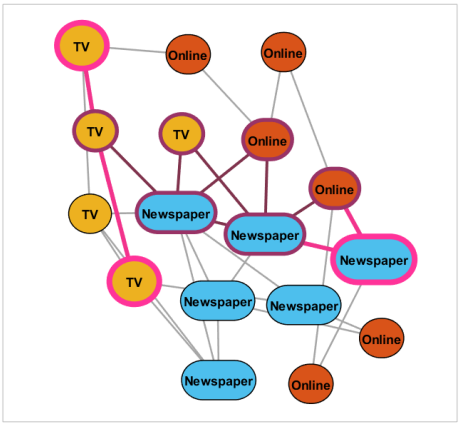
```
createEdgeClass(net, 'edgeclass1', true, true, 'Medium Audience');
setEdgeColors(net, [0.5 0.2 0.3], true, 'edgeclass1');
setEdgeLineWidth(net, 2,  true, 'edgeclass1');

subplot(1, 2, 1);
plot(net);

createEdgeClass(net, 'edgeclass2', true, true, 'Medium Audience', 'Large Audience');
setEdgeColors(net, [0.95 0.2 0.55], true, 'edgeclass2');
setEdgeLineWidth(net, 3,  true, 'edgeclass2');

subplot(1, 2, 2);
plot(net);
```





Published with MATLAB® R2016b

Final network

```
figure(2);
clf();
set(gcf, 'Position', [200 40 950 640]);
setNodeLabels(net, Nodes.media);
setNodeFontSize(net, 11);
plot(net);
```