

```

clear all
close all
clc
pause off

% d36e70
Param.ColorMyRed = [82.75,43.14,43.92]/100;
% 6a93b0
Param.ColorMyBlue = [41.57,57.65,69.02]/100;
% 7fb0b2
Param.ColorMyGreen = [49.8,69.02,69.8]/100;

K = 20;
Size = 20;
A = ones(K);
B = rand(K,K);
Indices1 = find(B < 2/20);
Indices2 = find(B >= 2/20);

figure
spy(A,'k',Size)
set(gcf,'Color',Param.ColorMyRed)
set(gcf,'InvertHardCopy','off');
set(gcf,'position',[1000,1000,410,410])
axis square
axis off
saveas(gcf,'MatrixFull.jpg');

B(Indices1) = 0;
B(Indices2) = 1;
figure
spy(B,'k',Size)
set(gcf,'Color',Param.ColorMyBlue)
set(gcf,'InvertHardCopy','off');
set(gcf,'position',[1000,1000,410,410])
axis square
axis off
saveas(gcf,'MatrixDense.jpg');

B = 1 - B;
figure
spy(B,'k',Size)
set(gcf,'Color',Param.ColorMyGreen)
set(gcf,'InvertHardCopy','off');
set(gcf,'position',[1000,1000,410,410])
axis square
axis off
saveas(gcf,'MatrixSparse.jpg');

Density = 0.2;
A = GenerateBlockMatrix(10*[1,2,3,4],Density);
figure
spy(A,'k',8)
set(gcf,'Color',Param.ColorMyRed)
set(gcf,'InvertHardCopy','off');

```

```

set(gcf, 'position', [1000, 1000, 410, 410])
axis square
axis off
saveas(gcf, 'GraphArtificial.jpg');
pause

```

```

A = ReadNetwork('aves-wildbird-network.edges');
figure
spy(A, 'k', 8)
set(gcf, 'Color', Param.ColorMyBlue)
set(gcf, 'InvertHardCopy', 'off');
set(gcf, 'position', [1000, 1000, 410, 410])
axis square
axis off
saveas(gcf, 'GraphRealWorld_Unstructured.jpg');
pause

```

```

[A, communities] ...
= ReadNetwork('aves-wildbird-network.edges', 'aves-wildbird-network.txt');
figure
spy(A, 'k', 8)
set(gcf, 'Color', Param.ColorMyRed)
set(gcf, 'InvertHardCopy', 'off');
set(gcf, 'position', [1000, 1000, 410, 410])
axis square
axis off
saveas(gcf, 'GraphRealWorld_Structured.jpg');
pause

```

```

function A = GenerateBlockMatrix(Sizes, Density)

```

```

M = sum(Sizes);
Aux = arrayfun(@ones, Sizes, 'UniformOutput', false);
A = blkdiag(Aux{:});
AZero = find(A==0);
B = rand(M, M);
Indices1 = find(B < Density);
Indices2 = find(B >= Density);
B(Indices1) = 0;
B(Indices2) = 1;
B(AZero) = 1-B(AZero);
A = B;

```

```

end

```

```

function [A, Communities] = ReadNetwork(FileGraph, FileCommunities)

```

```

fclose('all');
File = fopen(FileGraph);
A = zeros(2);
Nodes = 2;
while true
    line = fgetl(File);
    if isnumeric(line)
        break
    end
end

```

```

end
Edges = str2num(line);
if isempty(Edges)
    continue
end
if min(Edges(1),Edges(2)) < 0.5
    continue
end
Nodes = max([Edges(1:2),Nodes]);
if size(A,1) < Nodes
    A = blkdiag(A,zeros(max(Nodes-size(A,1),1000)));
end
A(Edges(1),Edges(2)) = true;
end
fclose('all');
A = A(1:Nodes,1:Nodes);
A = double(A + A' > 0);
if nargin > 1
    File = fopen(FileCommunities);
    if ~isnumeric(FileCommunities) & File > 2
        Aux = {};
        while true
            line = fgetl(File);
            if isnumeric(line)
                break
            end
            Aux{end+1} = str2num(line);
        end
        m = min(cellfun(@min,Aux));
        if m < 1
            Aux = cellfun(@(x)x+1,Aux,'UniformOutput',false);
        end
        aux = [Aux{:}];
        B = zeros(size(aux));
        B(aux) = 1:Nodes;
        A = A(aux,aux);
        Communities = cell(size(Aux));
        for i = 1:length(Communities)
            Communities{i} = B(Aux{i});
        end
    else
        Communities = {};
    end
    if ~isnumeric(FileCommunities)
        fclose('all');
    end
else
    Communities = {};
end
end
end

```