Structural dimension reduction (PMSA)

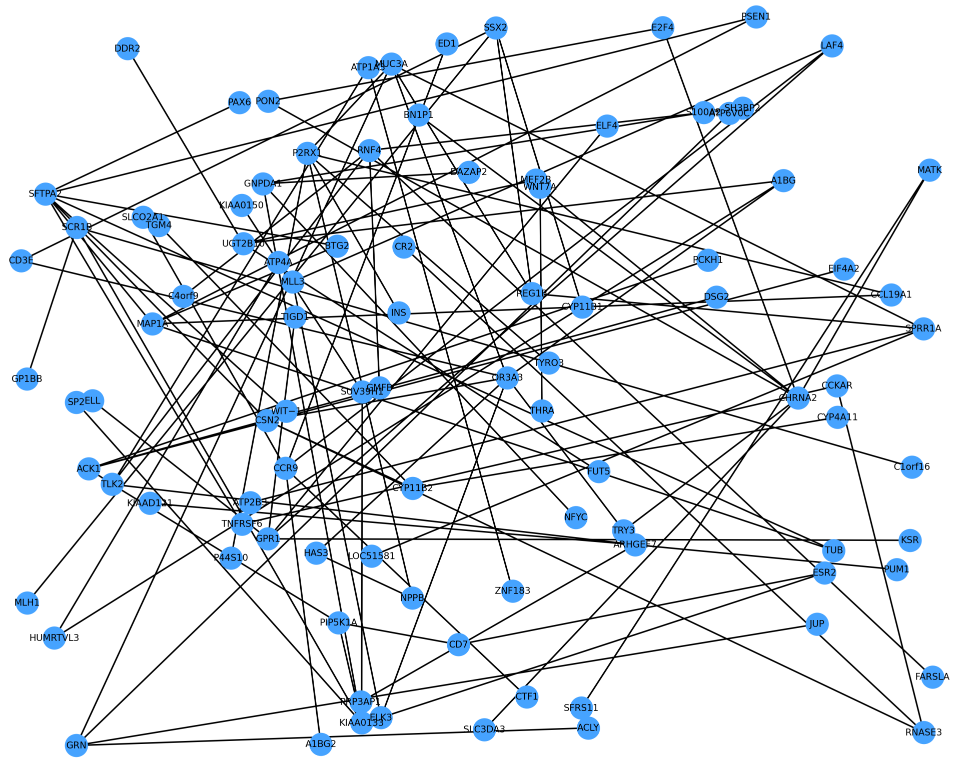
September 26, 2022

[1]:
```python
import random
import networkx as nx
import matplotlib.pyplot as plt
from networkx.algorithms import connectivity
from networkx.algorithms import all_shortest_paths
```

[2]:
```python
# This gene network is obtained from "An empirical Bayes approach to inferring␣
↪large-scale gene association networks", which contains 96 genes and 104␣
↪edges.
G_nodes=["PCKH1","ACK1","DSG2","EIF4A2","TLK2","ZNF183","ATP1A3","RNF4","FARSLA",
         "GMFB","SH3BP2","C4orf9","PRP3AP1","KIAAD121","PUM1","ARHGEF7",
         "KIAA0150","PIP5K1A","SUV39H1","KIAA0133","SCR1B","C1orf16","GP1BB",
         "GNPDA1","NFYC","SP2","DAZAP2","S100A2","THRA","WNT7A", "CHRNA2",
         "PON2","SLC3DA3","E2F4","BN1P1","SFRS11","MATK","TRY3","OR3A3","ELK3",
         "ESR2","CD7","MLL3","LAF4","MEF2B", "DDR2","FUT5","UGT2B10","ATP4A",
         "HUMRTVL3","A1BG","CCR9","CTF1","SLCO2A1","A1BG2","ED1","KSR","TIGD1",
         "GPR1","ELL","ELF4","JUP","ATP6V0C","ACLY","GRN","ATP2B3","SPRR1A",
         "REG1B","LOC51581","CYP11B1","MLH1","SSX2","MUC3A","CD3E","HAS3",
         "CSN2","NPPB","P2RX1","P44S10","TYRO3","CCL19A1","INS","MAP1A","BTG2",
         "PSEN1","PAX6","TUB","WIT-1","SFTPA2","RNASE3","CCKAR","CR2",
         "CYP11B2","TNFRSF6","TGM4","CYP4A11"]
edges=[("PCKH1","ACK1"),("DSG2","ACK1"),("EIF4A2","ACK1"),("TLK2","ACK1"),
       ("TLK2","ATP1A3"),("ZNF183","ATP1A3"),("RNF4","GMFB"), ("RNF4","FARSLA"),
       ("RNF4","SH3BP2"),("RNF4","C4orf9"),("RNF4","TLK2"),("PIP5K1A","TLK2"),
       ("ARHGEF7","TLK2"),("ARHGEF7","PRP3AP1"), ("ARHGEF7","KIAAD121"),
       ("ARHGEF7","PUM1"),("PIP5K1A","KIAA0133"),("KIAA0133","SUV39H1"),
       ("SP2","KIAA0133"),("SUV39H1","KIAA0150"),("SCR1B","KIAA0133"),
       ("SCR1B","C1orf16"),("SCR1B","GP1BB"),("GNPDA1","KIAA0133"),
       ("GNPDA1","S100A2"),("GNPDA1","NFYC"),("GNPDA1","DAZAP2"),
       ("CHRNA2","WNT7A"),("CHRNA2","BN1P1"),("CHRNA2","E2F4"),
       ("CHRNA2","SLC3DA3"),("CHRNA2","PON2"),("E2F4","PON2"),("CHRNA2","MATK"),
       ("CHRNA2","TRY3"),("SFRS11","MATK"),("OR3A3","TRY3"),("OR3A3","ELK3"),
       ("ESR2","ELK3"),  ("MLL3","LAF4"),("MLL3","ELK3"),("ESR2","CD7"),
       ("PIP5K1A","CD7"),("UGT2B10","DDR2"),("UGT2B10","MEF2B"),
       ("UGT2B10","A1BG"),("OR3A3","A1BG"),("HUMRTVL3","A1BG"),
       ("LAF4","GPR1"),("CCR9","LAF4"),("CCR9","SLCO2A1"),("CCR9","CTF1"),
       ("CCR9","ED1"), ("CCR9","A1BG2"),("ELL","GPR1"),("TIGD1","GPR1"),
       ("KSR","GPR1"),("ELF4","GPR1"),("GRN","ACLY"),("GRN","JUP"),
       ("GRN","ATP6V0C"),("GRN","MUC3A"), ("REG1B","MUC3A"),
       ("SPRR1A","MUC3A"),("SPRR1A","ATP2B3"),("SPRR1A","LOC51581"),
       ("SPRR1A","REG1B"),("SSX2","REG1B"),("SSX2","CYP11B1"),("SSX2","MLH1"),
       ("SSX2","CD3E"),("OR3A3","CD3E"),("P2RX1","P44S10"),("P2RX1","TYRO3"),
       ("P2RX1","INS"),("P2RX1","CCL19A1"),("P2RX1","NPPB"),("HAS3","NPPB"),
       ("HAS3","OR3A3"),("CSN2","OR3A3"),("CSN2","RNASE3"), ("CSN2","CYP11B2"),
       ("CSN2","SFTPA2"),("MAP1A","SFTPA2"),("TNFRSF6","SFTPA2"),
       ("PAX6","SFTPA2"),("TUB","SFTPA2"),("PSEN1","SFTPA2"),("BTG2","SFTPA2"),
       ("MAP1A","CCL19A1"),("MAP1A","BTG2"),("MAP1A","PSEN1"),("MAP1A","TUB"),
       ("RNASE3","CR2"),("RNASE3","CCKAR"),("TNFRSF6","CCKAR"),
       ("CYP11B2","TGM4"),("THRA","WNT7A"),("CYP11B2","CYP4A11"),
       ("UGT2B10","FUT5"),("HUMRTVL3","ATP4A"),("WIT-1","SFTPA2"),
       ("OR3A3","MUC3A"), ("P2RX1","MUC3A")]
```

[3]:
```
G = nx.Graph()
G.add_nodes_from(G_nodes)
G.add_edges_from(edges)
```

[4]:
```
plt.figure(figsize=(10,8),dpi=400)
nx.draw(G,pos = nx.random_layout(G),node_color = '#46A3FF',edge_color =␣
 ↪'black',with_labels = True,font_size = 6,node_size =200)
```



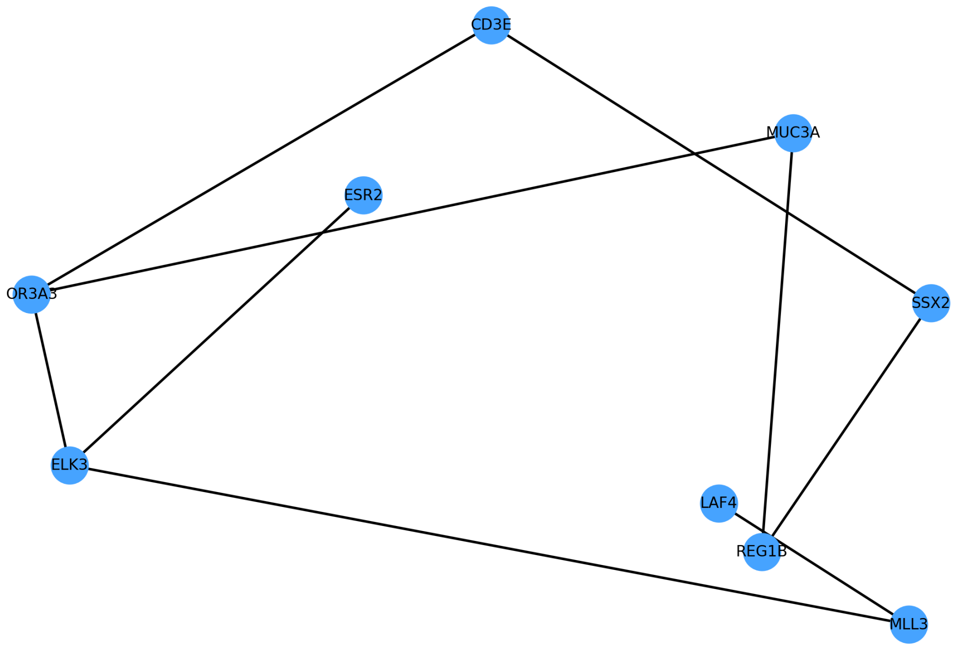
```
[5]: def all_CloseSeparator(g_C,a,b):
         N_A = set(g_C.adj[a])
         N_V_C = set()
         V_remove_N_A = set(g_C.nodes) - N_A
         M_c = list(nx.connected_components(nx.subgraph(g_C, V_remove_N_A)))
         for i in range(0,len(M_c)):
             if b in M_c[i]:
                 for j in M_c[i]:
                     N_V_C = N_V_C | set(g_C.adj[j])
                 N_V_C = N_V_C - M_c[i]
                 break
         return N_V_C
     def PMSA(r, g):
         s = 1
         H = set(r)
         while s == 1:
             s = 0
             M = set(g.nodes) - H
             M_c=list(nx.connected_components(nx.subgraph(g, M)))
```

```
        for i in range(0,len(M_c)):
            Cap_1 = set()
            set_ = set()
            Cap_1 = Cap_1 | M_c[i]
            for j in list(M_c[i]):
                set_ = (set_ | set(g.adj[j]))
            set_ = ((set_ - M_c[i]) & H)
            if len(set_)>1:
                for a in set_.copy():
                    set_.remove(a)
                    for b in set_:
                        if b not in g.adj[a]:
                            Cap_1 = Cap_1 | {a,b}
                            G_Cap_1=nx.subgraph(g, Cap_1)
                            Min_close_a = all_CloseSeparator(G_Cap_1,a,b)
                            Min_close_b = all_CloseSeparator(G_Cap_1,b,a)
                            H = (H | (Min_close_a|Min_close_b))
                            s = 1
                            break
                    else:
                        continue
                    break
    return H
```

[6]:
```
R=["ESR2","LAF4","SSX2"]#Set of variables of interest.
H = PMSA(R, G)
H
```

[6]:
```
{'CD3E', 'ELK3', 'ESR2', 'LAF4', 'MLL3', 'MUC3A', 'OR3A3', 'REG1B', 'SSX2'}
```

[7]:
```
#This means that the original model with 96-dimensional space is collapsed into␣
 ↪a submodel in a 9-dimensional space, which reduces the computational␣
 ↪complexity in further research.
G_H = nx.subgraph(G, H)
plt.figure(dpi=400)
nx.draw(G_H,pos = nx.random_layout(G),node_color = '#46A3FF',edge_color =␣
 ↪'black',with_labels = True,font_size = 6,node_size =200)
```


CD3E
MUC3A
ESR2
OR3A3
SSX2
ELK3
LAF4
REG1B
MLL3

```
[ ]:
```