

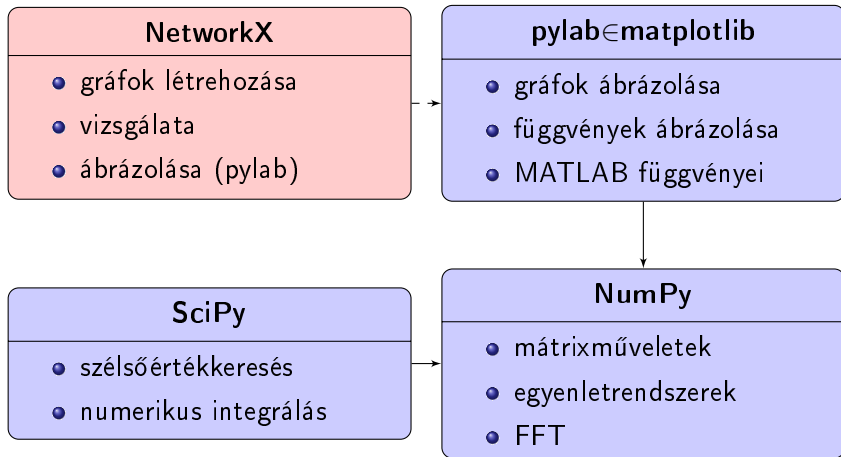
A NetworkX modul

A Python használata gráfok és komplex hálózatok vizsgálatában

Horváth Árpád <horvath.arpad@roik.bmf.hu>

Budapesti Műszaki Főiskola
Regionális Oktatási és Innovációs Központ (ROIK)
Székesfehérvár

2009. május 28.



Gráfok létrehozása

Lehet:

- súlyozott
- irányított,
DiGraph osztály
- többszörös él,
X(Di)Graph osztály

Export–import

- dot-file
- Ábra: pdf, png, eps...

Gráfok vizsgálata

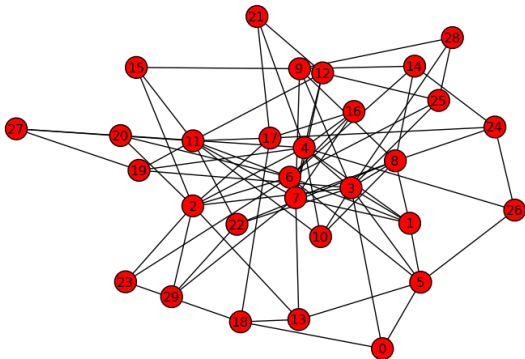
- átmérő
- távolságok

Rajzolás

- többszörös élé még nem megy (v0.99)
- v0.36-nál az ábrázolás néha szétesik
- ipython -pylab opcióval hasznos

```
import networkx as nx
nx.__version__ # 0.99
G = nx.complete_graph(5)
print G.nodes() # [0,1,2,3,4]
print G.number_of_edges() # 5*4/2 = 10
0 in G # True
(0,1) in G.edges() # True
G.add_path([2, "P", "y", "t", "o", "n", 4 ])
for csucs in G:
    print csucs, G.has_edge(0,csucs)
nx.draw(G)
pylab.show() # ipython -pylab esetén nem kell
```

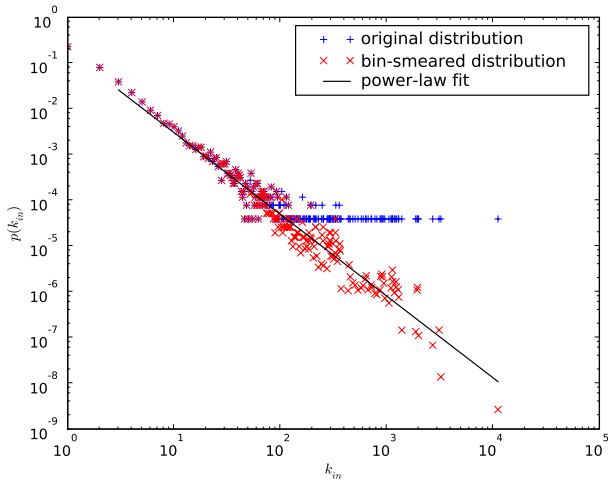
Barabási–Albert gráf



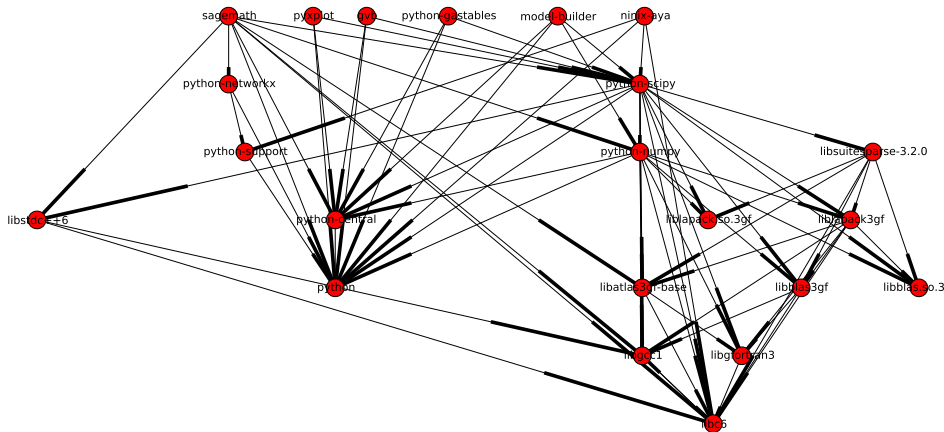
```
G = nx.barabasi_albert_graph(30, 3)
print G.degree()
# [3, 6, 9, 12, 11, 6, 10, 10, 8, 6, 4, 9, 5, 4, 4,
3, 4, 6, 4, 4, 4, 3, 4, 3, 4, 4, 3, 3, 3, 3]
nx.distance.diameter(G) # 3
nx.shortest_path(G, 27,26) # [27, 19, 4, 26]
```

packages (saját): Csomagfüggőségi gráf

```
import packages
G = packages.get_graph()
print G.number_of_nodes(), G.number_of_edges()
      # (27503, 126331)
deg = G.degree()
max(deg)  # 11858
deg.count(1) # 4337
deg.count(2) # 3564
deg.count(3) # 3401
deg.count(4) # 2952
dd = [ deg.count(i) for i in range(m+1) ]
pylab.plot(dd)
```



Csomagfüggőségi gráf



Köszönöm a figyelmet!

```
bzr branch http://mail.roik.bmf.hu/complex\_networks
```