

# network *visualization*

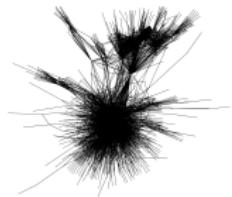
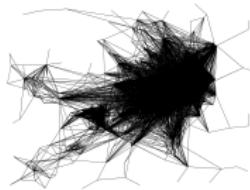
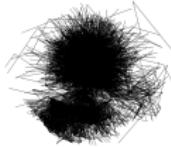
introduction to *network science in Python* (*NetPy*)

Lovro Šubelj  
University of Ljubljana  
18th Jan 2022

# visualization *overview*

network *visualization* with *wiring diagram*

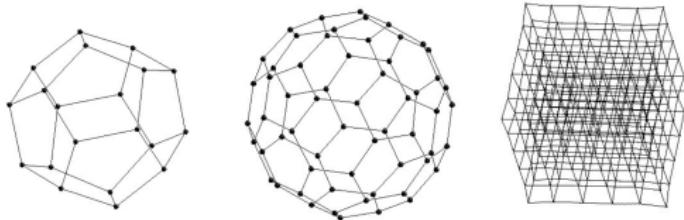
- 1st find *network layout* as node *coordinates* in Euclidean plane etc.
- 2nd *representation* of *network links*? strength, pattern, shape, color etc.
- 3rd *representation* of *network nodes*? size, shape, color, label etc.



similar link lengths, no crossings, displays symmetry, even node distribution etc.

## visualization *Fruchterman-Reingold*

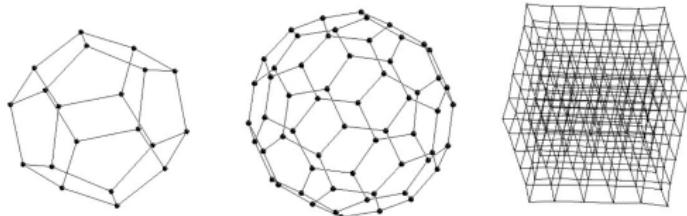
- Fruchterman-Reingold *force-directed layout* [FR91]
- move nodes thus to *minimize layout energy*
- *repulsive force* between *nodes i* and *j* is  $\propto -c^2/l_{ij}$
- *attractive force* between *neighbors i* and *j* is  $\propto l_{ij}^2/c$ 
  - $l_{ij}$  is *Euclidean distance* between *nodes i* and *j*
  - $c$  is *appropriate constant* set to  $\propto \sqrt{\text{area}/n}$



pleasing with similar link lengths, symmetry & even distribution

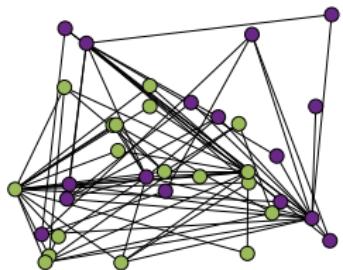
## visualization *Kamada-Kawai*

- Kamada-Kawai *graph theoretic layout* [KK89]
- move nodes thus to *minimize layout energy*  $l_{ij} \propto d_{ij}$
- *effective force* between *nodes i* and *j* is  $\propto 1/d_{ij}^2$ 
  - $l_{ij}$  is *Euclidean distance* between *nodes i* and *j*
  - $d_{ij}$  is *geodesic distance* between *nodes i* and *j*

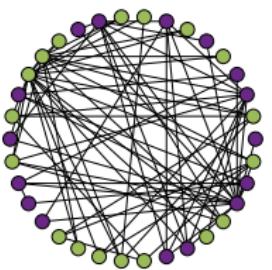


desired layout distance between nodes is their graph distance

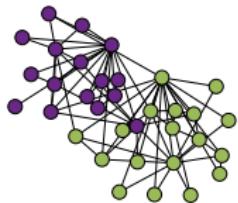
# visualization *karate*



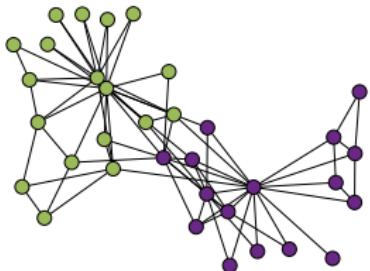
random layout



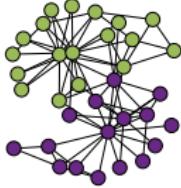
circular layout



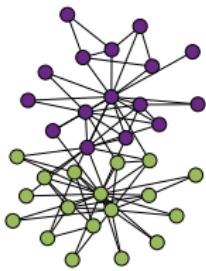
spring embedded layout [Ead84]



Fruchterman-Reingold layout [FR91]

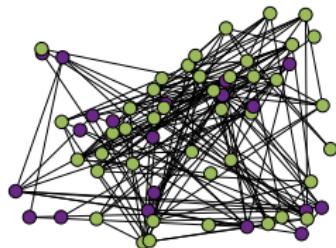


Kamada-Kawai layout [KK89]

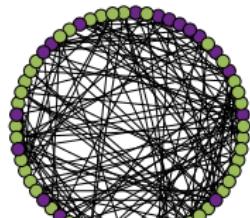


LGL layout [ADWM04]

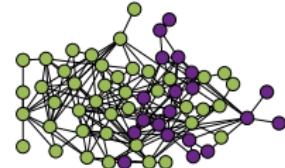
# visualization *dolphins*



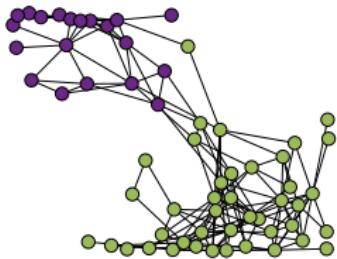
*random layout*



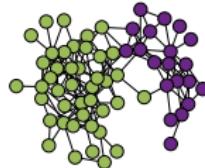
*circular layout*



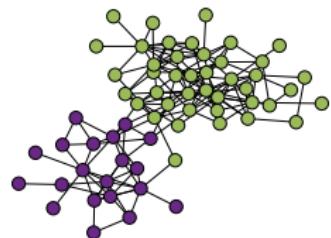
*spring embedded layout* [Ead84]



*Fruchterman-Reingold layout* [FR91]

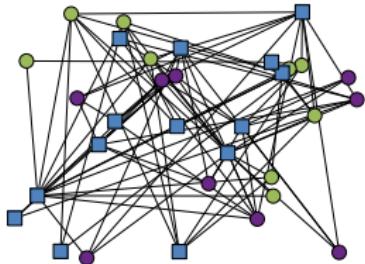


*Kamada-Kawai layout* [KK89]

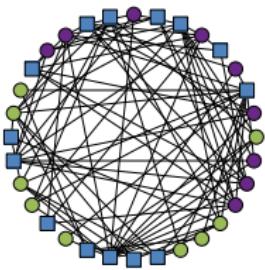


*LGL layout* [ADWM04]

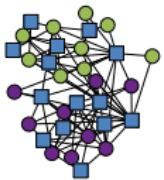
# visualization *women*



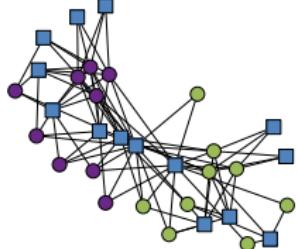
random layout



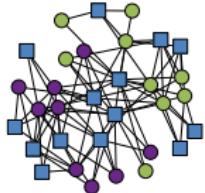
circular layout



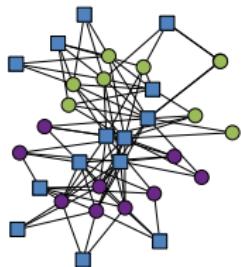
spring embedded layout [Ead84]



Fruchterman-Reingold layout [FR91]



Kamada-Kawai layout [KK89]



LGL layout [ADWM04]

# visualization *references*

-  Alex T. Adai, Shailesh V. Date, Shannon Wieland, and Edward M. Marcotte.  
LGL: Creating a map of protein function with an algorithm for visualizing very large biological networks.  
*J. Mol. Biol.*, 340(1):179–190, 2004.
-  Peter Eades.  
A heuristic for graph drawing.  
*Congressus Numerantium*, 42:149–160, 1984.
-  Thomas M. J. Fruchterman and Edward M. Reingold.  
Graph drawing by force-directed placement.  
*Softw: Pract. Exper.*, 21(11):1129–1164, 1991.
-  Helen Gibson, Joe Faith, and Paul Vickers.  
A survey of two-dimensional graph layout techniques for information visualisation.  
*Infor. Visual.*, 12(3-4):324–357, 2013.
-  Tomihisa Kamada and Satoru Kawai.  
An algorithm for drawing general undirected graphs.  
*Inform. Process. Lett.*, 31(1):7–15, 1989.
-  Stephen G. Kobourov.  
Force-directed drawing algorithms.  
In Roberto Tamassia, editor, *Handbook of Graph Drawing and Visualization*, pages 383–408. CRC Press, 2013.
-  Kwan-Liu Ma and Chris W. Muelder.  
Large-scale graph visualization and analytics.  
*Computer*, 46(7):39–46, 2013.