UNIVERSITÀ DEGLI STUDI DI PADOVA

Network Science

A.Y. 23/24

ICT for Internet & multimedia, Data science, Physics of data

Graph layout

Visualizing a graph

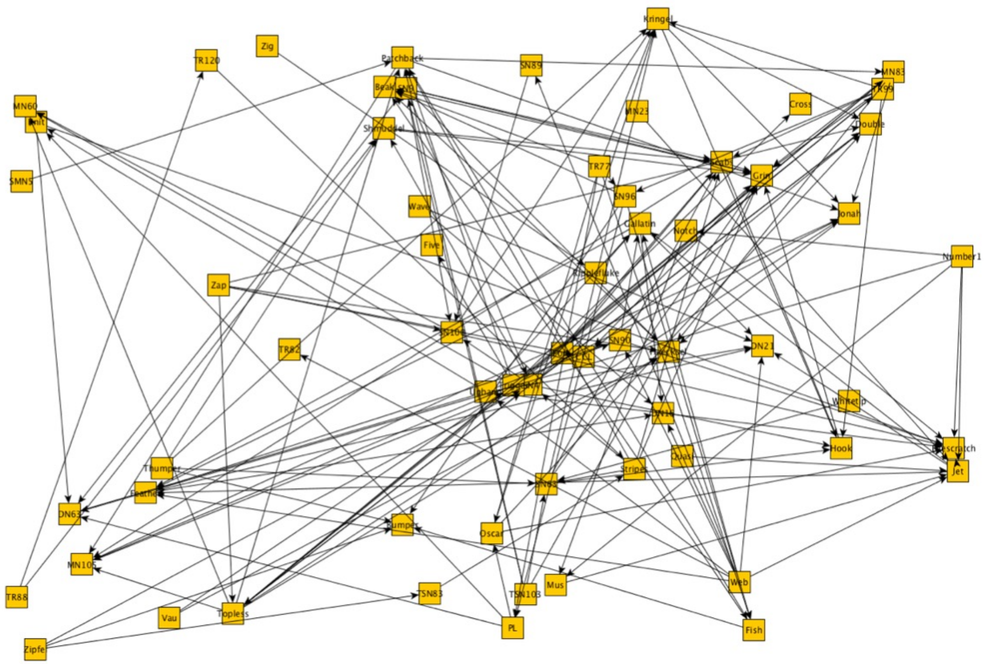
General layout problem

which aesthetic criteria would you optimize?

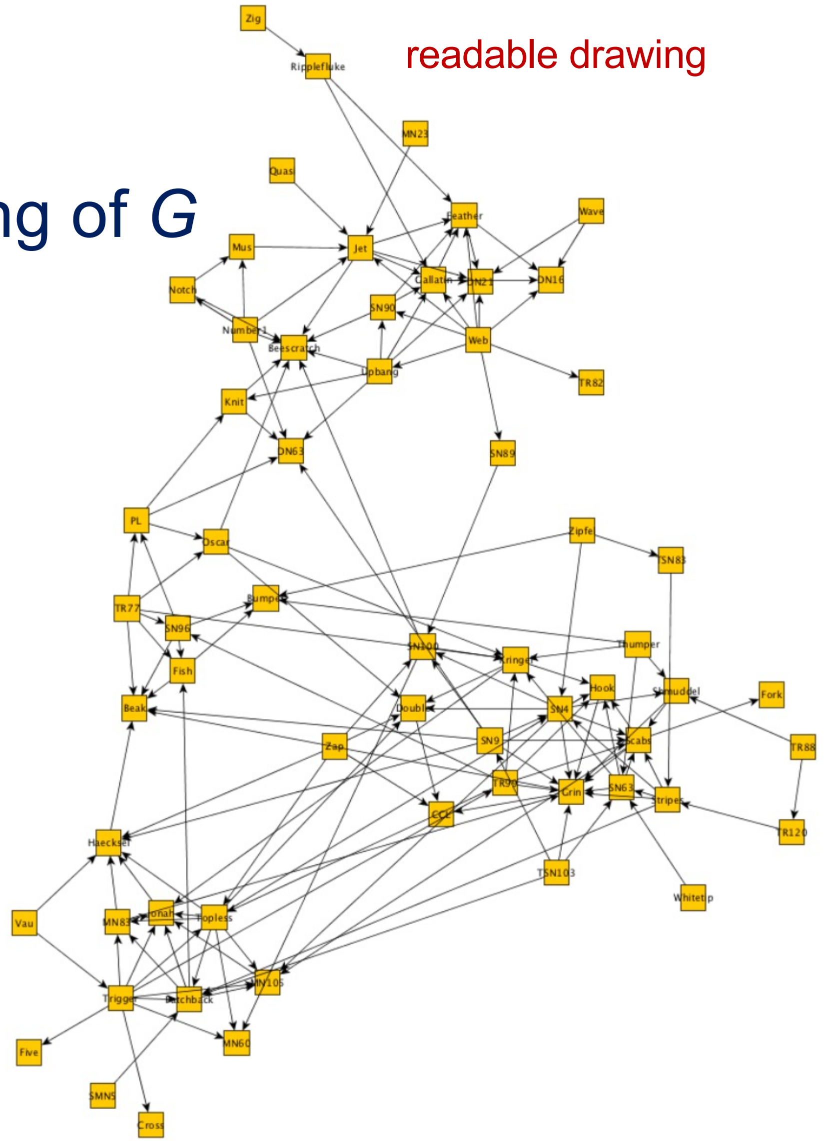
Problem:

Given a graph $G = (V,E)$

Find a clear and readable drawing of G



messy drawing



readable drawing

Aesthetic criteria

some relevant ones

- adjacent nodes are close
- non-adjacent far apart
- edges length proportional to their weight
- densely connected parts (clusters) to form communities
- as few crossings as possible
- nodes distributed evenly

… but optimization criteria partially contradict each other

Approaches to graph visualization

an overview

Before

- always based on some properties: tree, series-parallel graph, planar graph
- and on some additional information: ordering of the vertices, decompositions into SP-components
- NP-hard even in simple scenarios
  - edge lengths {1, 2} (Saxe, ’80)
  - planar drawing with unit edge lengths (Eades, Wormald, ’90)

Today

- more direct and intuitive method based on physical analogies : force directed algorithms
- the methods are very popular: intuitiveness, easy to program, generality, fairly satisfactory results,...

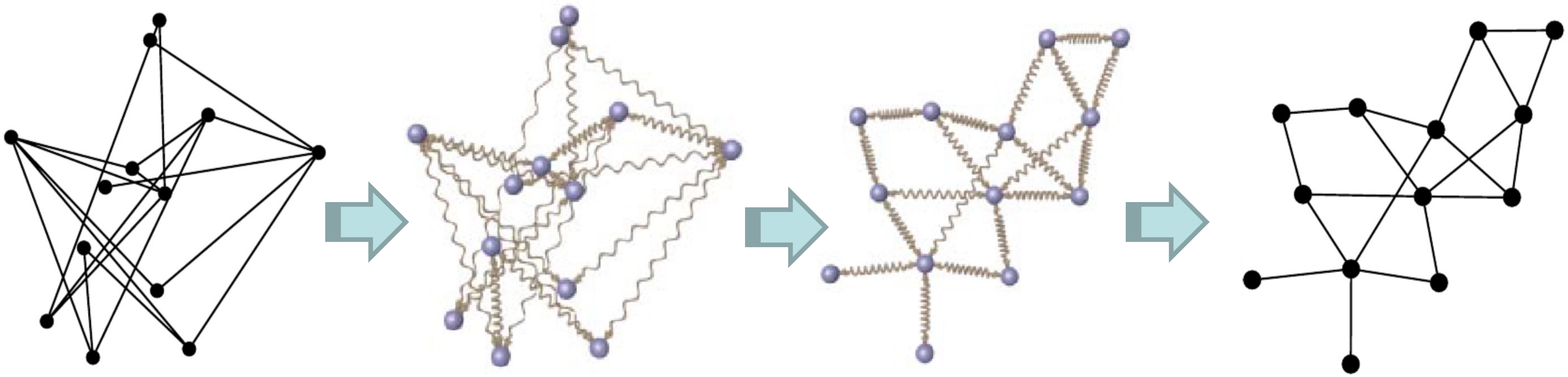
Force directed layout

a physical analogy for graphs

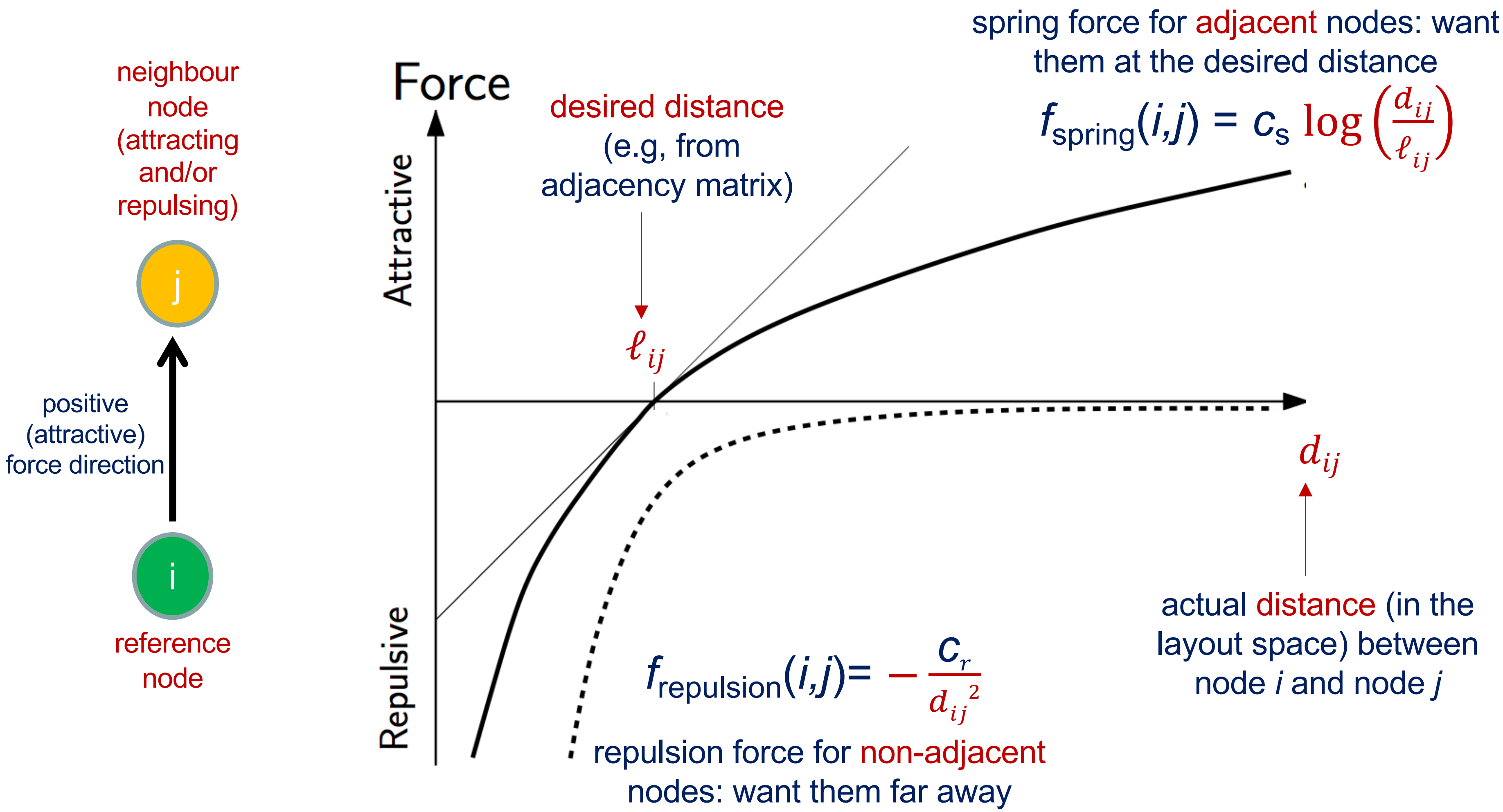
Springer embedder algorithm

Eades, "A heuristic for graph drawing" (1984)

"To embed a graph we replace the vertices by steel rings and replace each edge with a spring to form a mechanical system . . . The vertices are placed in some initial layout and let go so that the spring forces on the rings move the system to a minimal energy state."



Repulsive and attractive forces
in a spring embedder



The algorithm

iteratively reaching the steady state

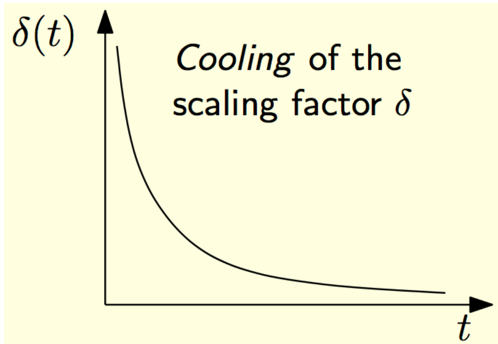
- Evaluate the force contribution on the *i*th node

direction from *i* to *j*

$$\boldsymbol{F}_i = \sum_{j \in N_i} f_{\text{spring}}(i,j)\,(\boldsymbol{p}_j - \boldsymbol{p}_i) + \sum_{j \notin N_i} f_{\text{repulsion}}(i,j) \cdot (\boldsymbol{p}_j - \boldsymbol{p}_i)$$

$$= c_s \sum_{j \in N_i} (\boldsymbol{p}_j - \boldsymbol{p}_i) \log\left(\frac{\|\boldsymbol{p}_j - \boldsymbol{p}_i\|}{\ell_{ij}}\right) - c_r \sum_{j \notin N_i} \frac{\boldsymbol{p}_j - \boldsymbol{p}_i}{\|\boldsymbol{p}_j - \boldsymbol{p}_i\|^2}$$

position (in the layout space) of node *i*



- Update the nodes position by applying forces

$$\boldsymbol{p}_i^{+} = \boldsymbol{p}_i + \delta\,\boldsymbol{F}_i$$

- Iterate until the forces are small enough: $\max_i \|\boldsymbol{F}_i\| < \varepsilon$

Discussion
on the spring embedder approach

Advantages

- very simple algorithm
- good results for small and medium-sized graphs
- good representation of symmetry/structure

Disadvantages

- system is not stable at the end
- converging to local minima
- not scalable - complexity is $O(N^2)$

Influence

- basis for many further ideas

Università degli Studi di Padova

Fruchterman and Reingold

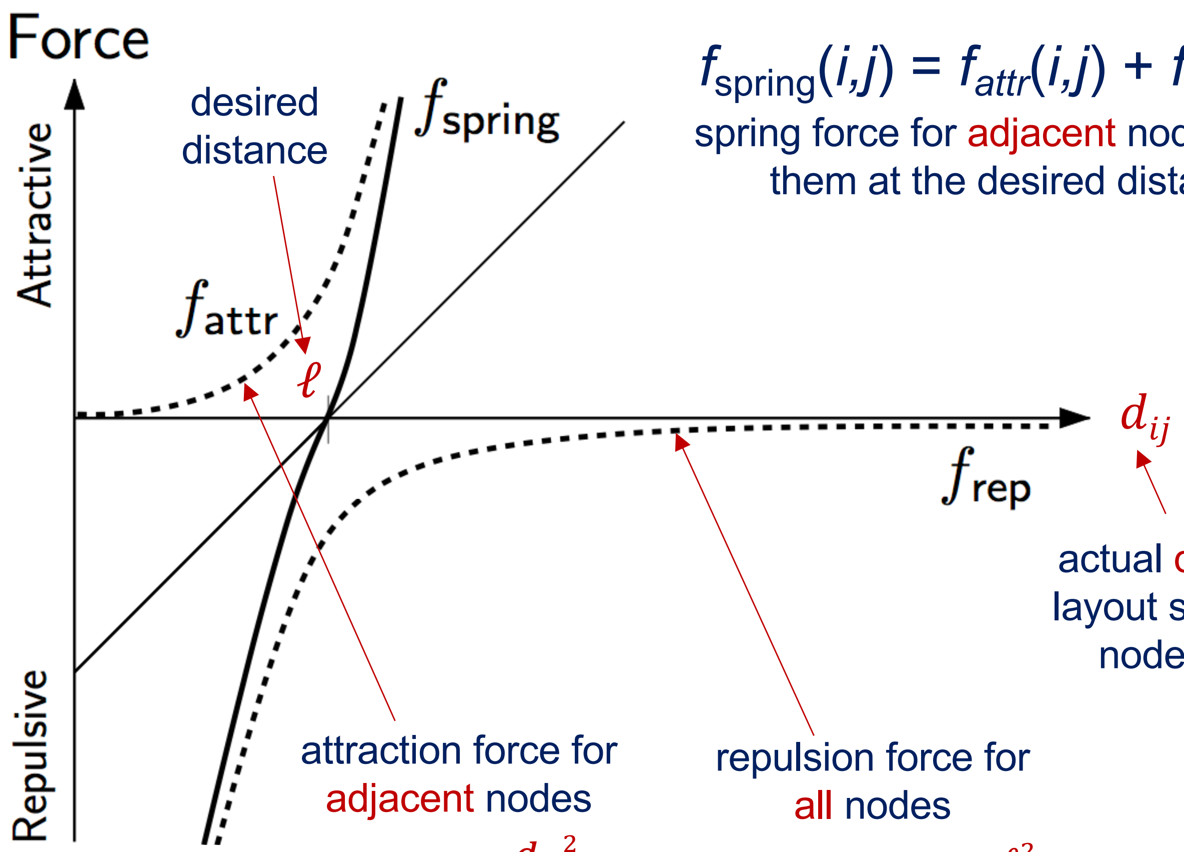
Fruchterman & Reingold, Graph drawing by force-directed placement (1991)

http://www.mathe2.uni-bayreuth.de/axel/papers/reingold:graph_drawing_by_force_directed_placement.pdf

neighbour node (attracting and/or repulsing)

positive (attractive) force direction

reference node



$f_{\text{spring}}(i,j) = f_{attr}(i,j) + f_{\text{rep}}(i,j)$

spring force for adjacent nodes: want them at the desired distance

actual distance (in the layout space) between node *i* and node *j*

attraction force for adjacent nodes

$$f_{\text{attr}}(i,j) = \frac{{d_{ij}}^2}{\ell}$$

more attractive

repulsion force for all nodes

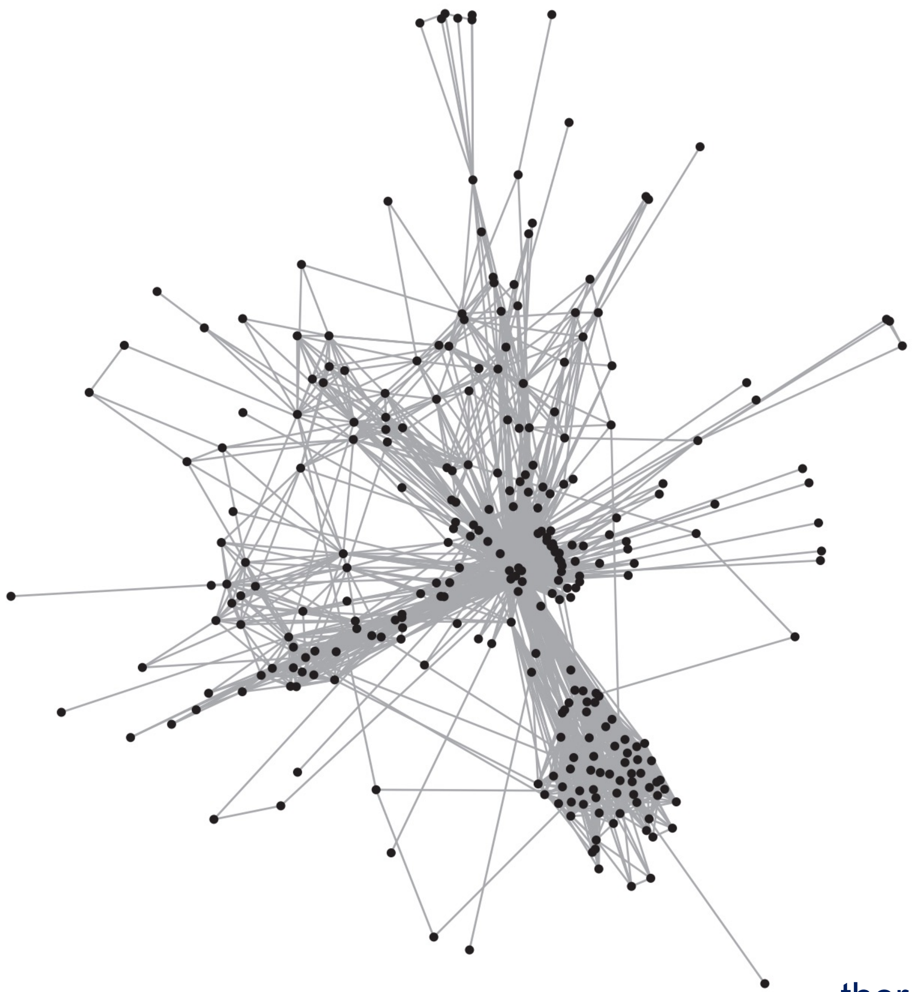
$$f_{\text{rep}}(i,j) = -\frac{\ell^2}{d_{ij}}$$

more repulsive

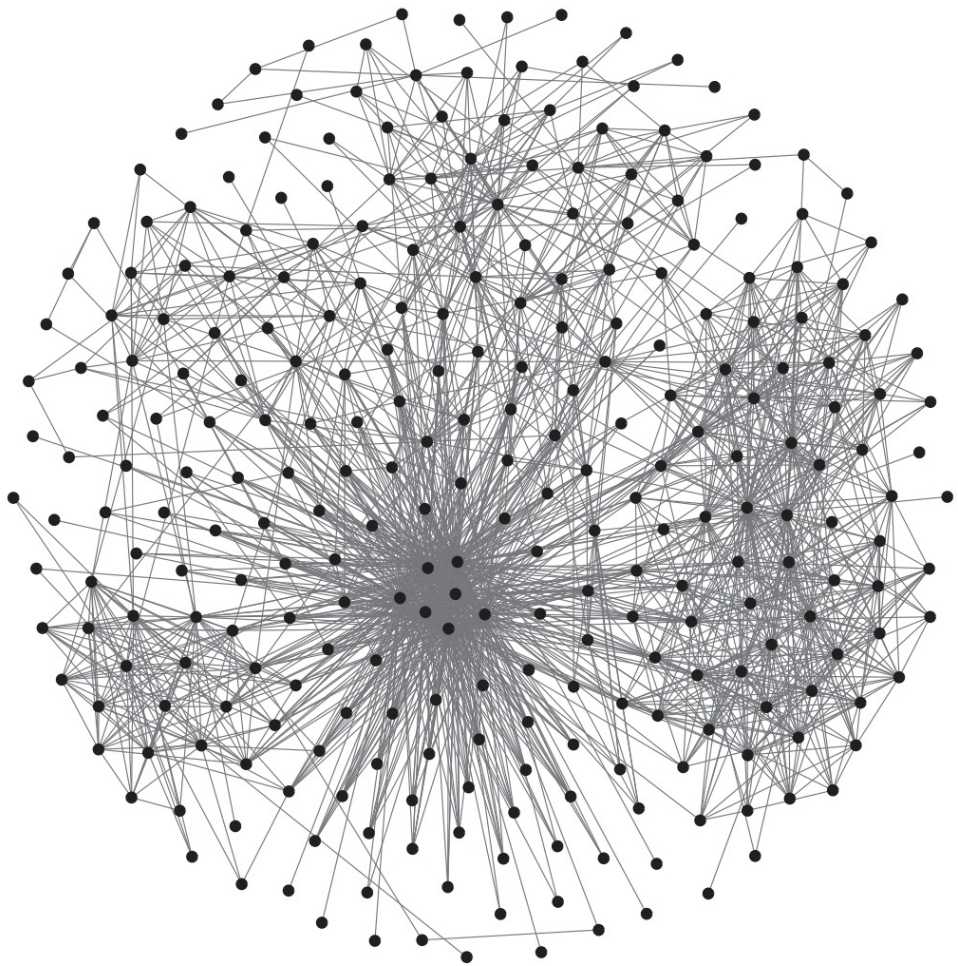
A visual example

protein interaction network

spring embedder



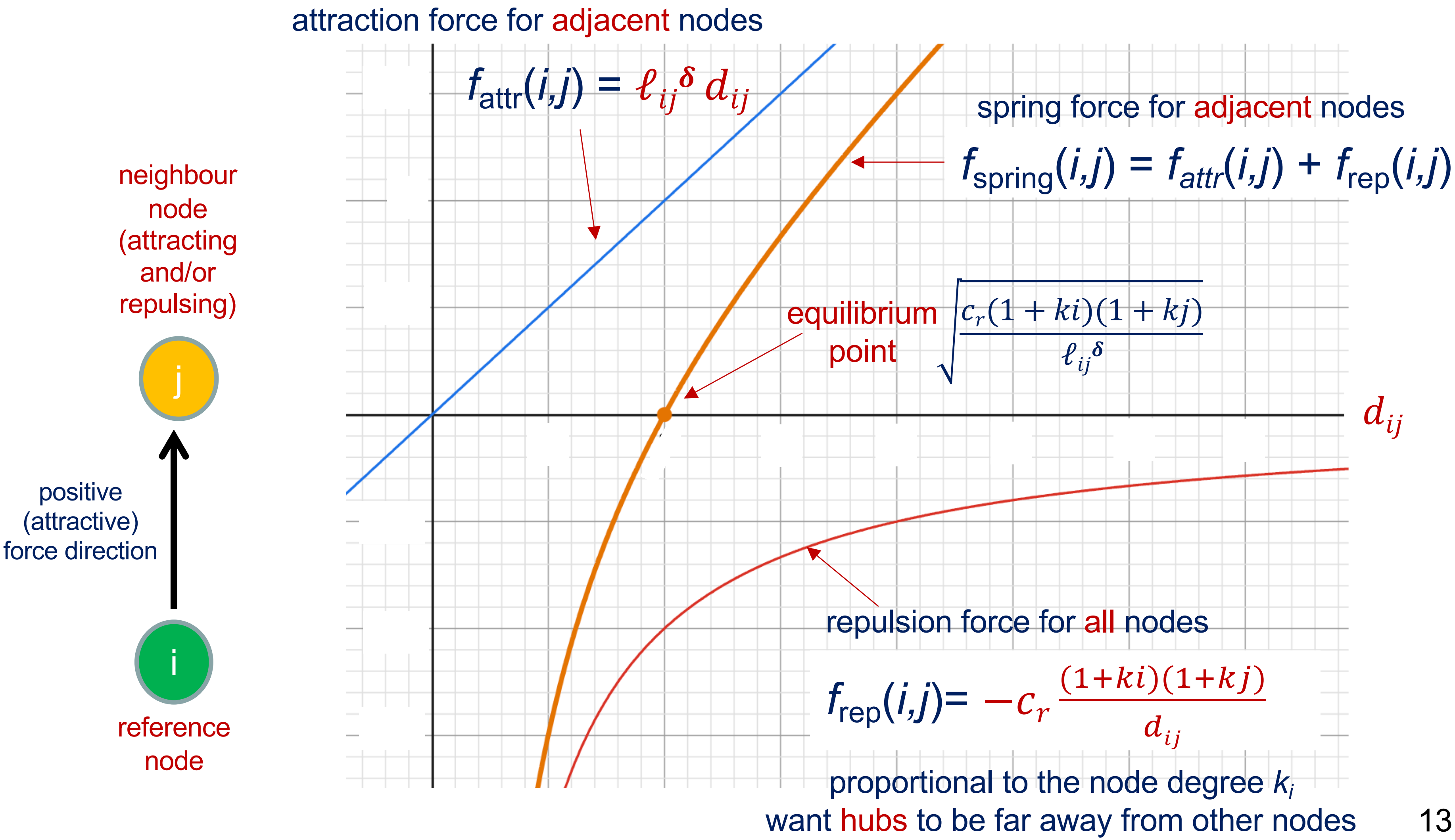
Fruchterman-Reingold



there exists a relation between Fruchterman-Reingold layout and the communities found by modularity

Force Atlas 2

Jacomy, Venturini, Heymann, Bastian, ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software, (2014)
https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0098679



Alternative modes

for attractive and repulsive forces

attraction force for adjacent nodes

$$f_{attr}(i,j) = \begin{cases} \ell_{ij}{}^{\delta}\, d_{ij} \\ \dfrac{\ell_{ij}{}^{\delta}\, d_{ij}}{1+ki} & \text{dissuade hubs} \\ \ell_{ij}{}^{\delta}\, \log(1+d_{ij}) & \text{linlog mode} \end{cases}$$

less attraction for authorities (will have more space)

weaker dependence on distance

repulsion force for all nodes

$$f_{rep}(i,j) = -c_r \frac{(1+ki)(1+kj)}{d_{ij}}$$

resized distance taking into account node sizes

Prevent overlapping mode $d'_{ij} = [d_{ij} - size_i - size_j]^+$

$d'_{ij} = 0$ ⇒

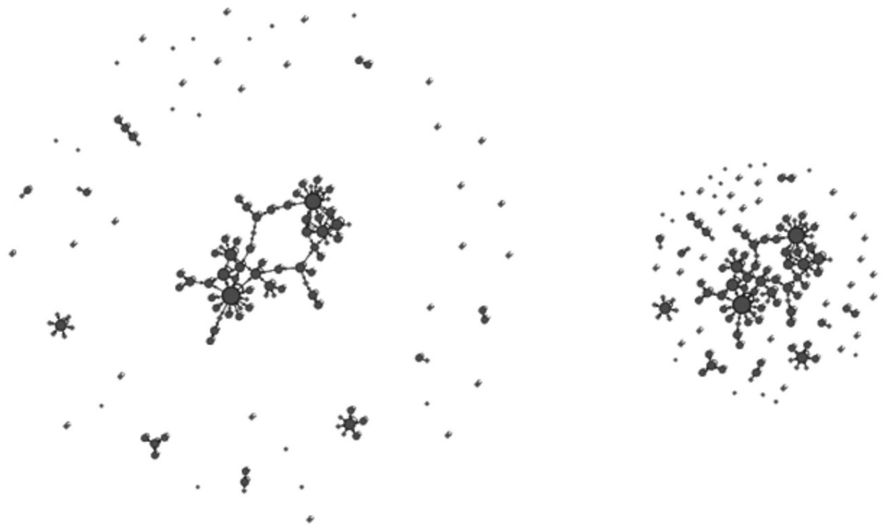
$$f_{attr}(i,j) = 0$$

$$f_{rep}(i,j) = -c'_r(1+ki)(1+kj)$$

Gravity

prevents disconnected components from drifting away

attracts nodes to the centre of the spatialisation. Its main purpose is to compensate repulsion for nodes that are far away from the centre



gravity force for
all nodes

$$f_{\text{gravity}}(i) = \begin{cases} c_g\,(1 + ki) \\ c_g\,(1 + ki)\,k_i \end{cases} \quad \text{strong gravity}$$

applied towards the baricenter $\boldsymbol{p}_{\text{bary}} = \frac{1}{N}\sum_i \boldsymbol{p}_i$

UNIVERSITÀ DEGLI STUDI DI PADOVA

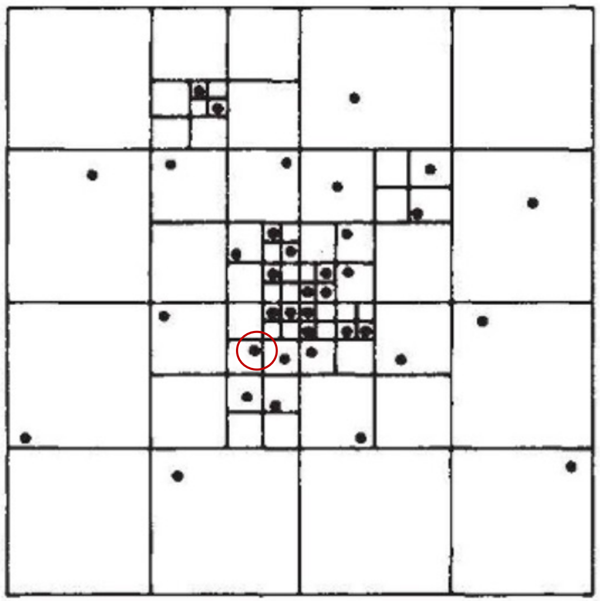
Approximate repulsion

Josh, Hut. "A hierarchical O (N log N) force-calculation algorithm." (1986)

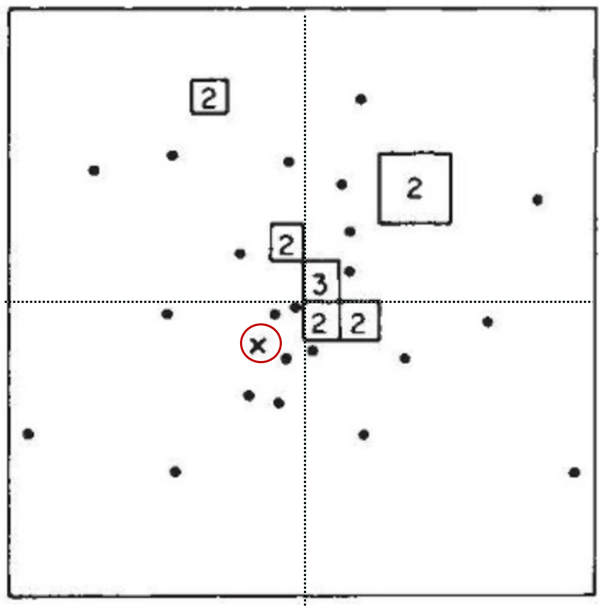
https://www.nature.com/articles/324446a0

an heuristic to circumvent the $O(N^2)$ complexity of calculating repulsion forces (2D example)

binary partition the space, minimum partition to isolate nodes



nodes far away from x are condensed in a single entity located at the baricenter



baricenter

$$\boldsymbol{p}_{\mathrm{b}} = \frac{1}{C}\sum_{j \in C} \boldsymbol{p}_j$$

approximate force

$$\boldsymbol{f}_{iC} = \sum_{j \in C} f_{ij} \cdot (\boldsymbol{p}_j - \boldsymbol{p}_i)$$

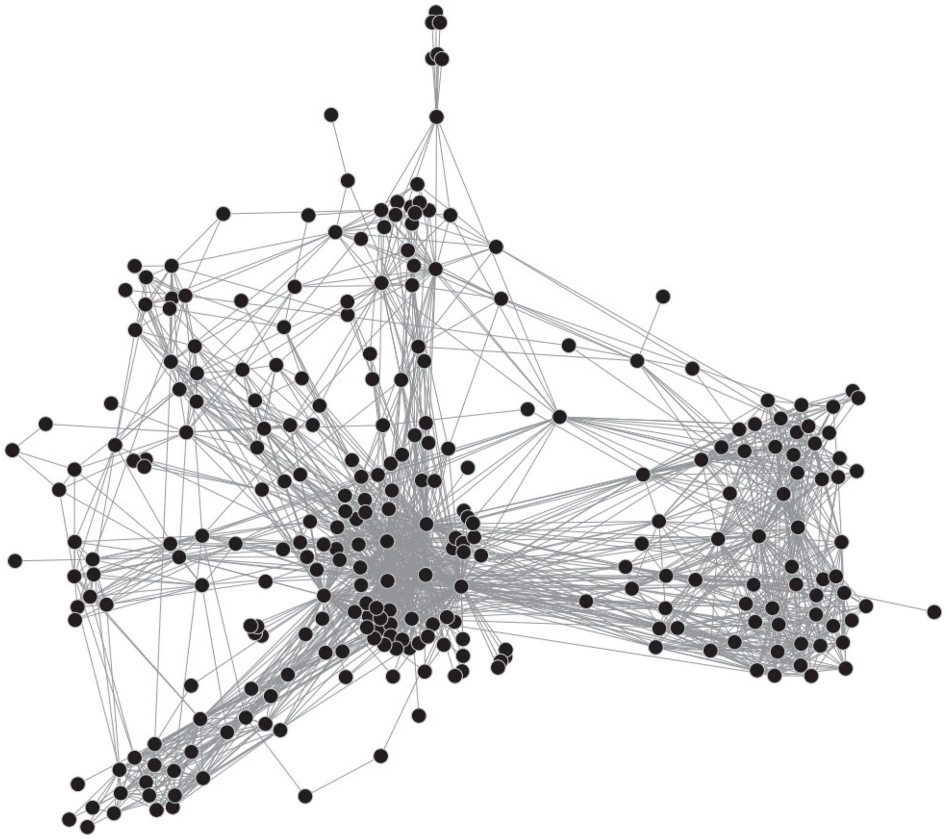
$$\cong \sum_{j \in C} f_{ib} \cdot (\boldsymbol{p}_j - \boldsymbol{p}_i)$$

$$= f_{ib} C (\boldsymbol{p}_{\mathrm{b}} - \boldsymbol{p}_i)$$

A visual example

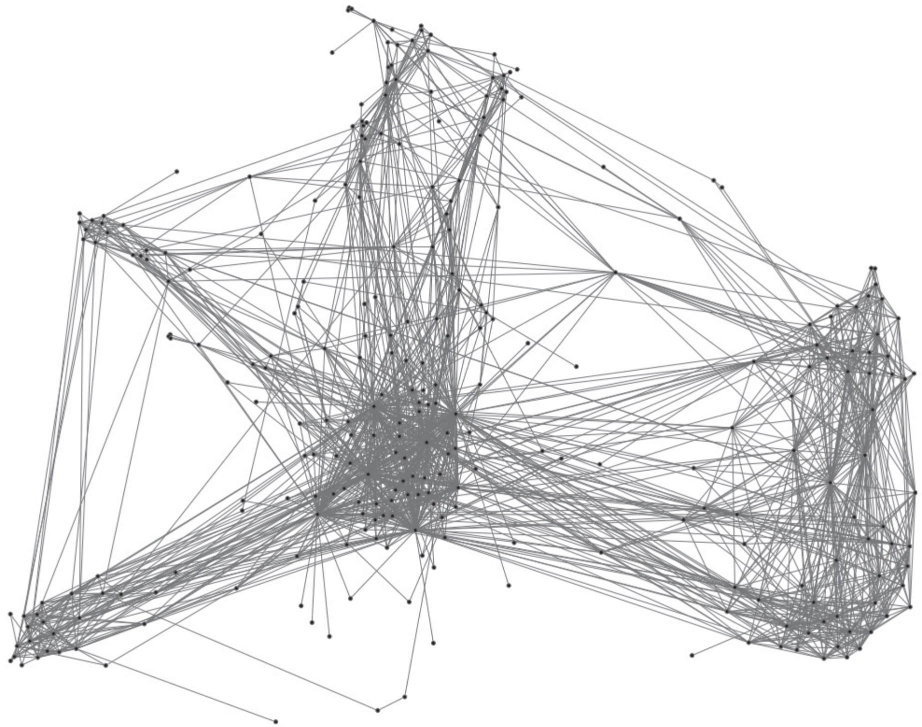
protein interaction network

Force Atlas 2



more clearly separate communities,
compact layout

Force Atlas 2 – linlog mode

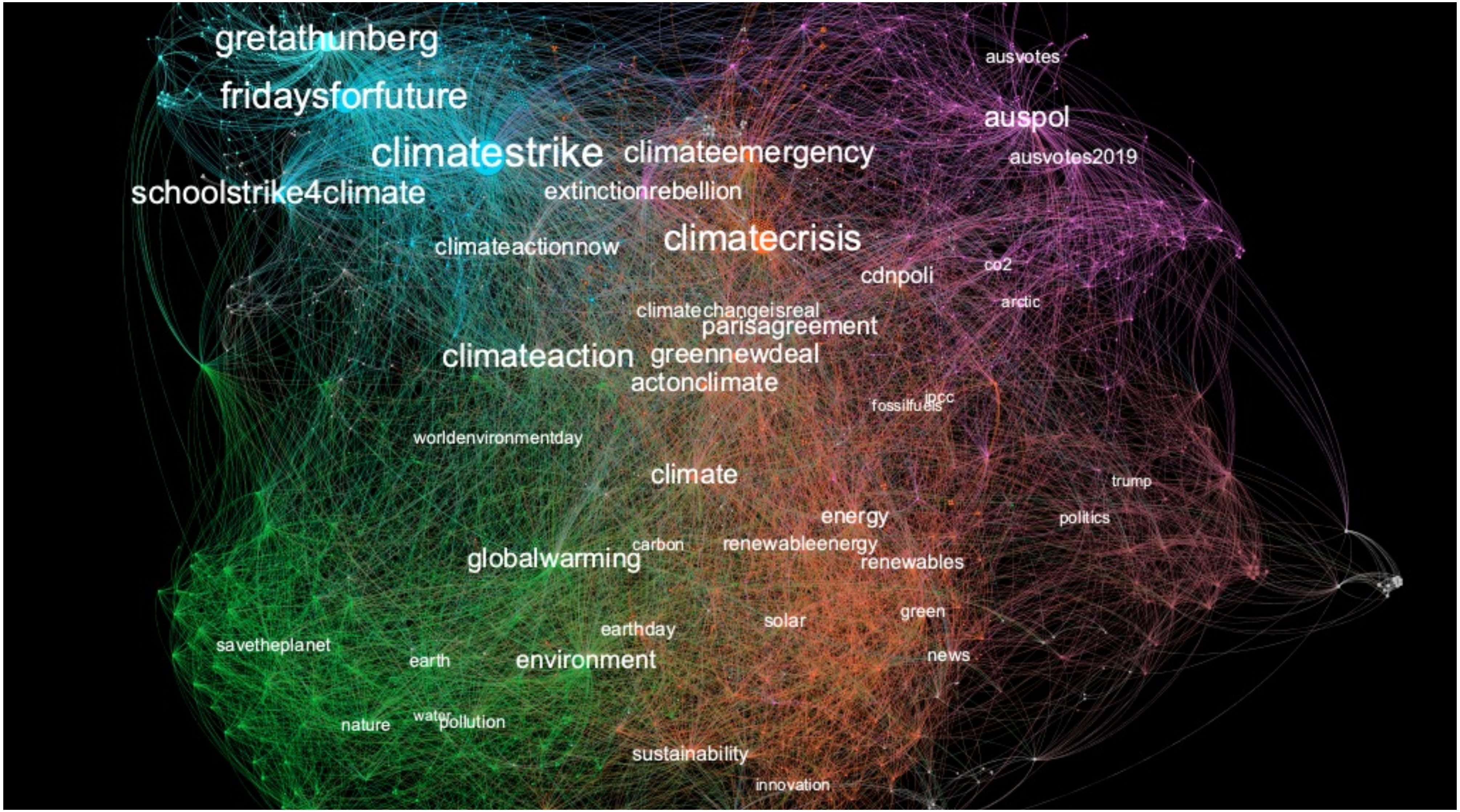


much greater spacing with linlog mode

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

A visual example

semantic network on #climateaction

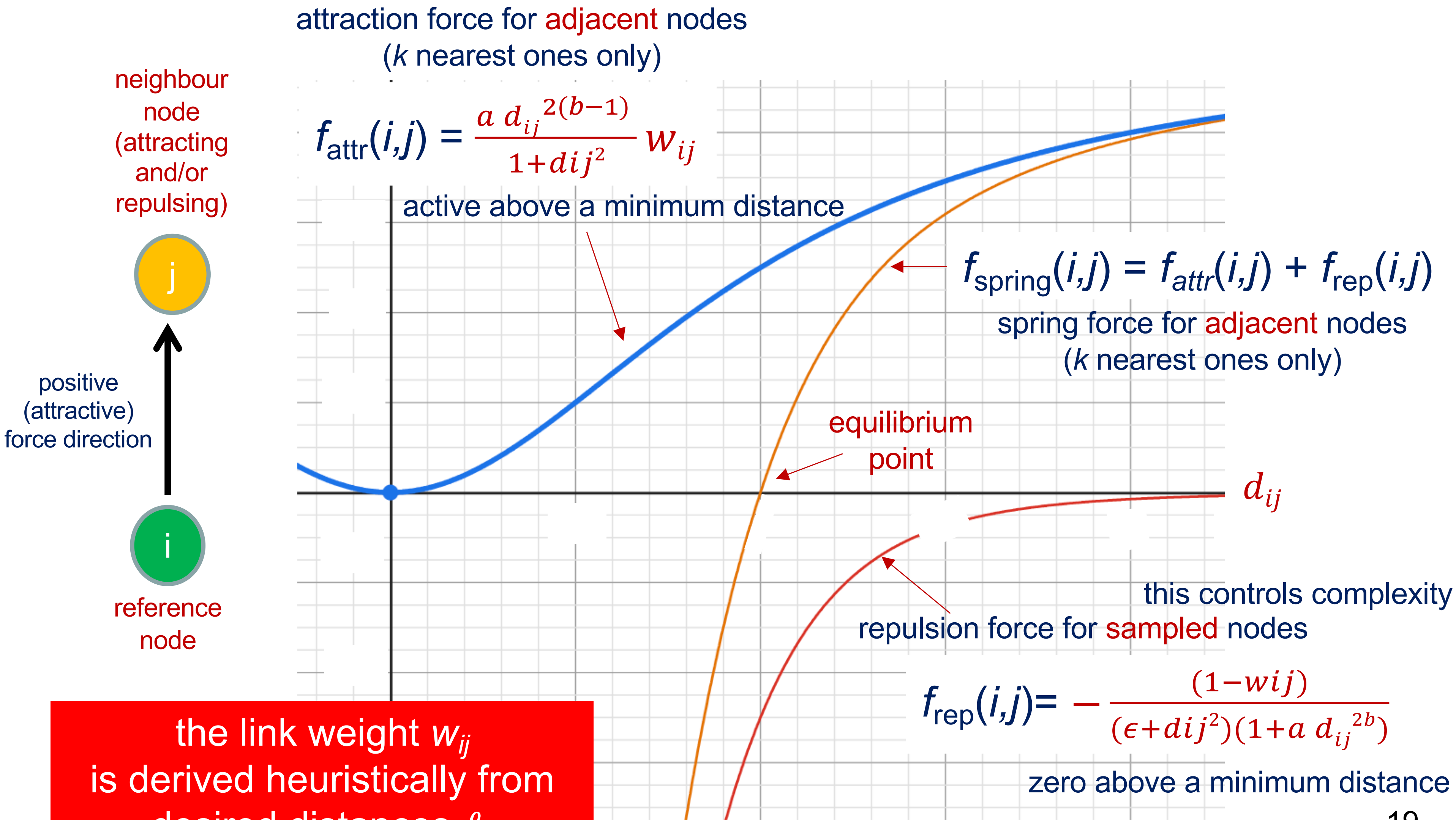


but to get such a nice output you also need manual intervention !!!

UMAP

Leland, Healy, Melville. "Umap: Uniform manifold approximation and projection for dimension reduction." (2018)

https://arxiv.org/abs/1802.03426



the link weight w_{ij} is derived heuristically from desired distances ℓ_{ij}

Implementation details

umap-learn.readthedocs.io/en/latest

assume a desired link distance ℓ_{ij} is available;
set $\ell_{ij} = \infty$ for nodes that are not connected

can also be extracted from an adjacency matrix

link weight model

$\ell_{i\,\min} = \min_j \ell_{ij}$

$$w_{ij} = \exp\left(-\frac{\ell_{ij} - \ell_{i\,\min}}{\sigma_i}\right)$$

identify σ_i by solving

$$\sum_{j \epsilon K i} \exp\left(-\frac{\ell_{ij} - \ell_{i\,\min}}{\sigma_i}\right) = \log_2(k)$$

k nearest neighbours to node *i*

hyperparameters model

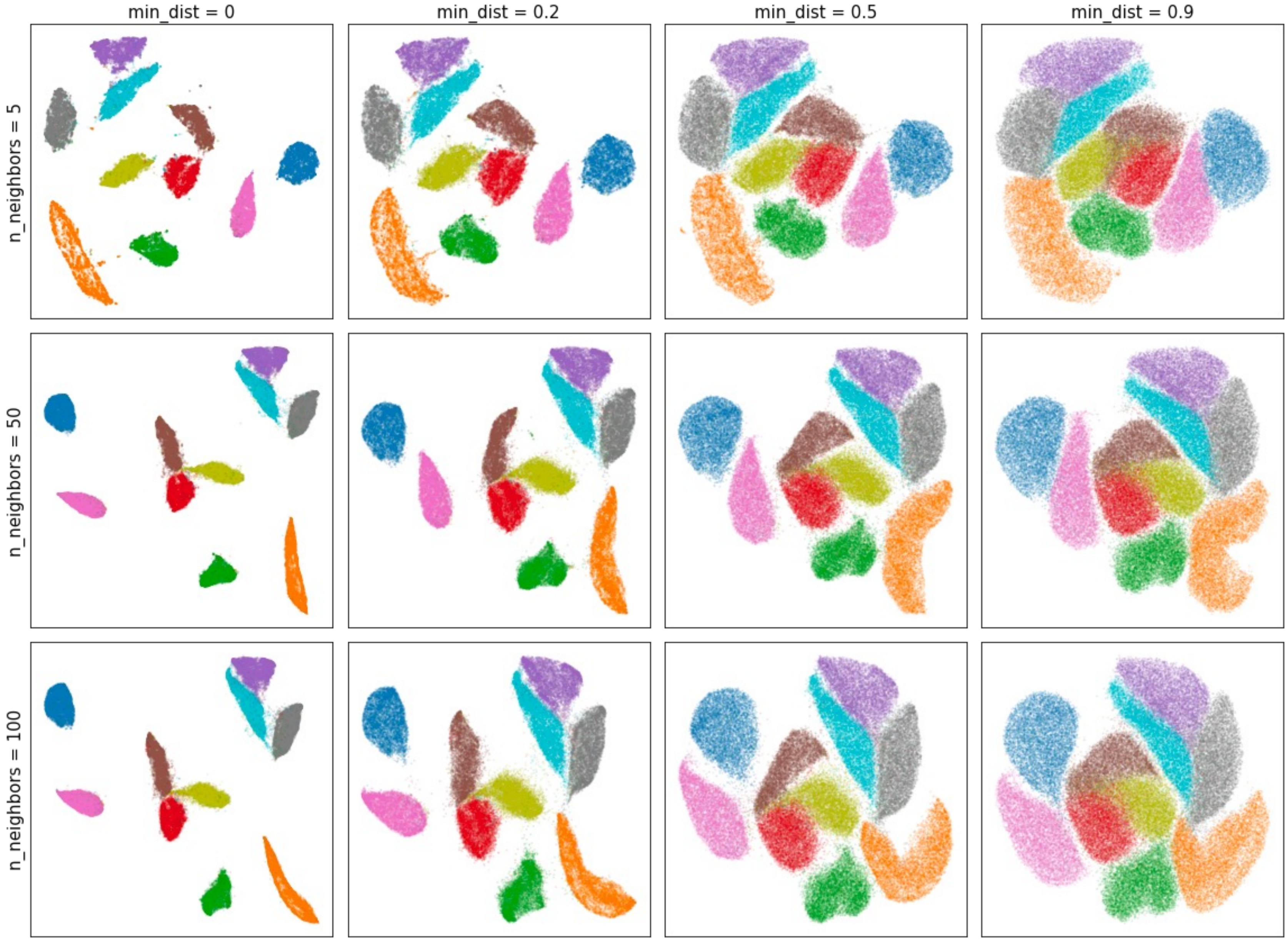
$$(a, b) = \operatorname{argmin}_{a,b} \sum_{i,j} \left(\frac{1}{1 + a\, dij^{2b}} - e^{-[d_{ij} - d_{\min}]^{+}}\right)^2$$

non-linear fitting to a smooth function

minimum distance parameter

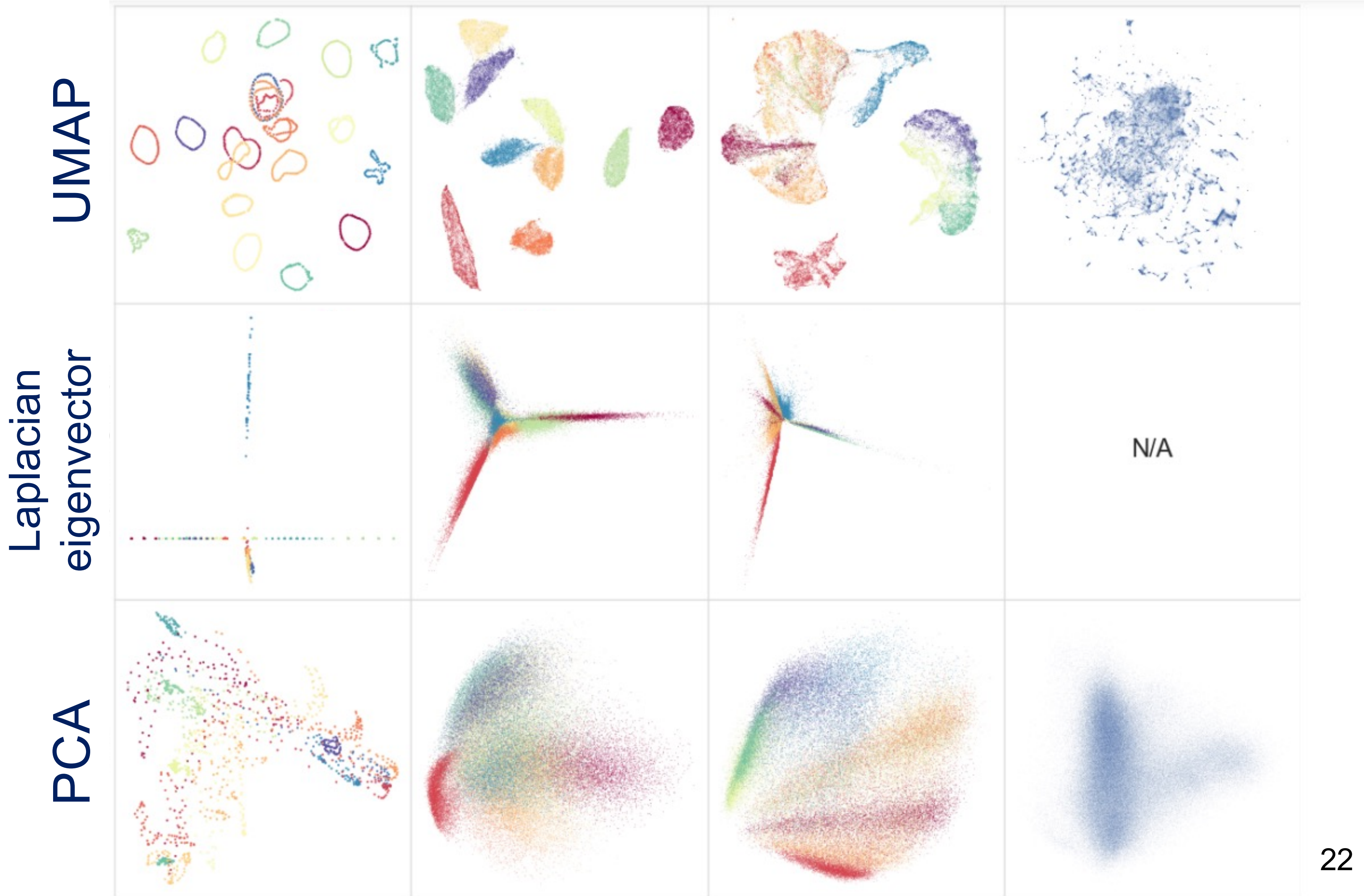
Parameters selection

on the role of the minimum distance



Comparison with spectral approaches

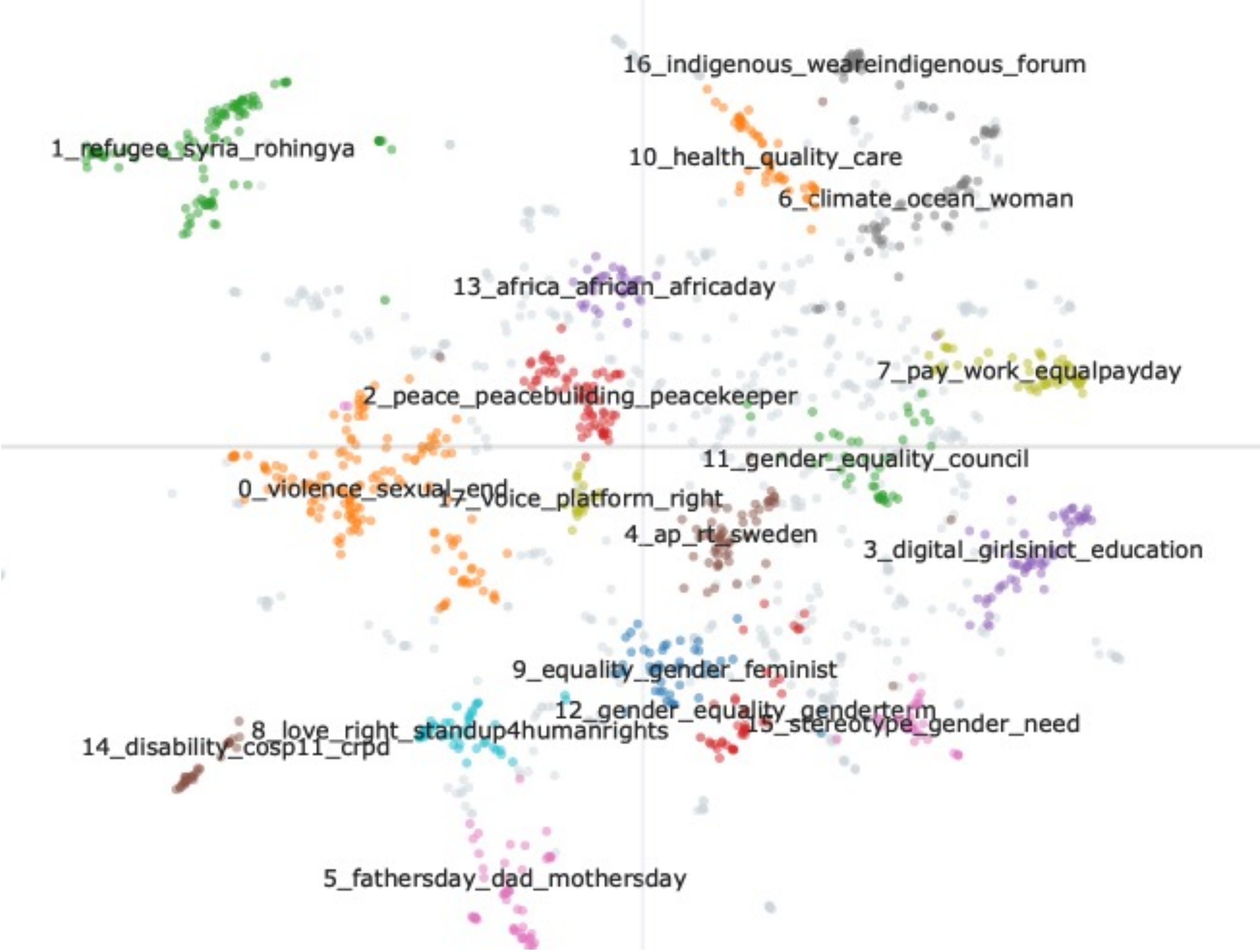
on the superior performance of force-directed algorithms



UNIVERSITÀ DEGLI STUDI DI PADOVA

A visual example

document network using BERTopic



Takeaways

on force directed layout

- easily understandable and implementable
- depending on the graphs (small and sparse)
- amazingly good layouts
- easily adaptable and configurable
- robust
- scalable (if wisely implemented)

But...

- quality mostly depends on the data (e.g., how to identify the desired link distance ℓ_{ij} … might be challenging from an adjacency matrix)
- fine-turning can be done by experts
- might need manual intervention

Software tools

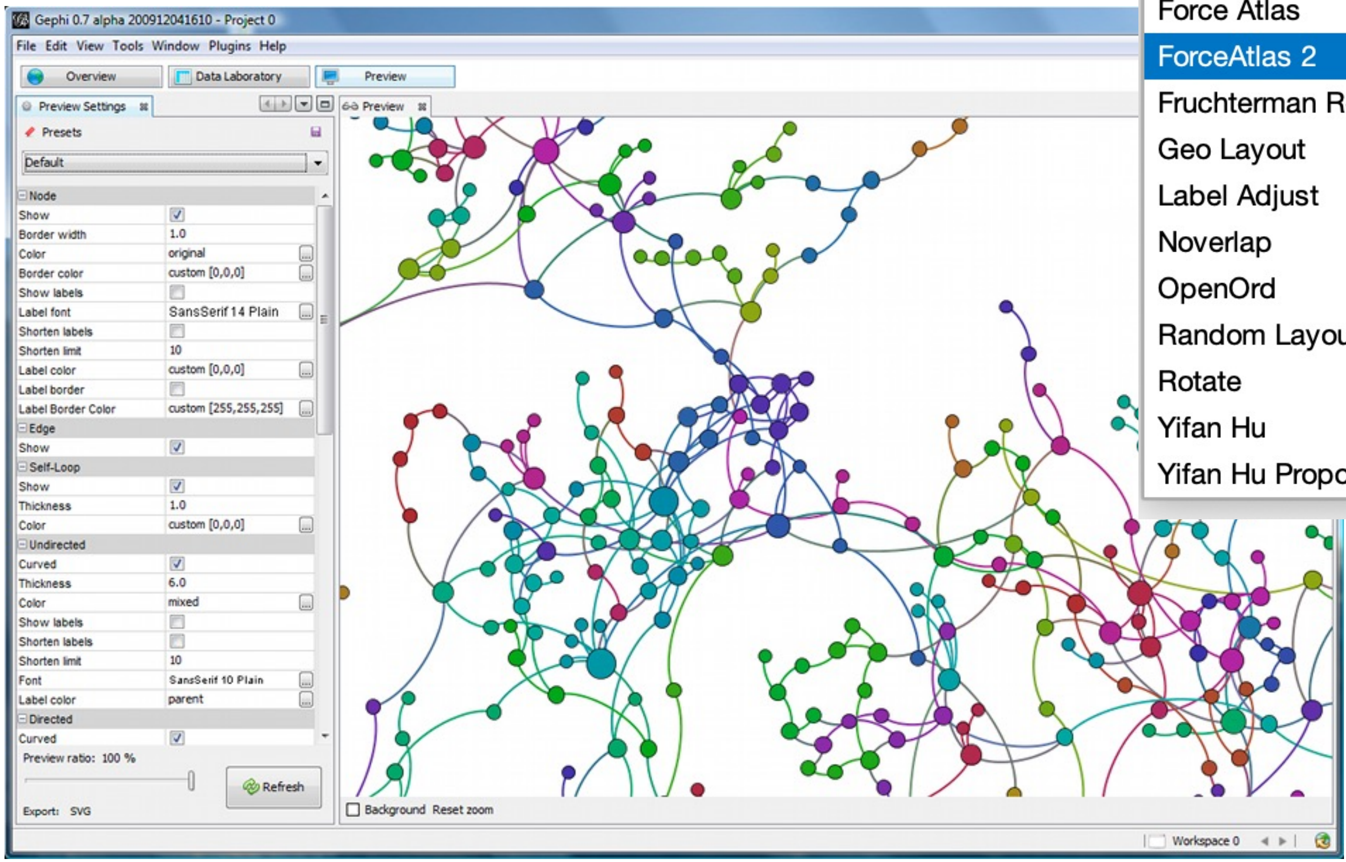
for force-directed layouts

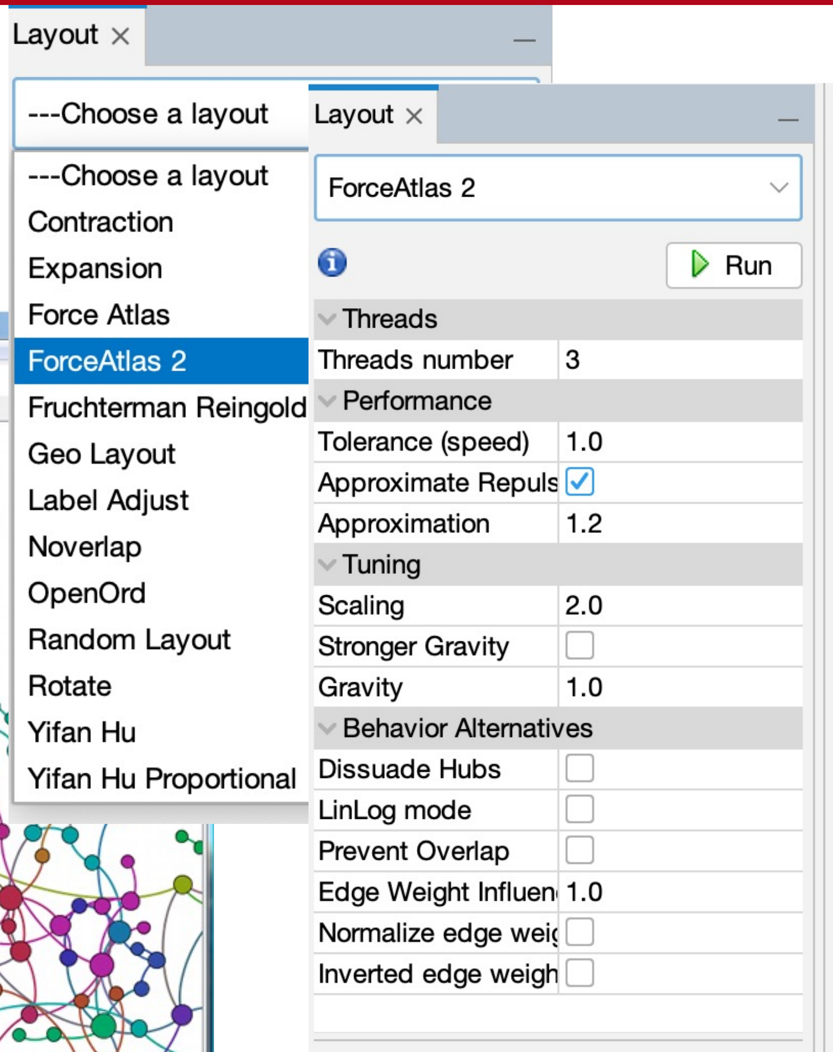
UNIVERSITÀ DEGLI STUDI DI PADOVA

Gephi

https://gephi.org/

Will be using it, install it in your PCs asap !!!





- NetworkX networkx.org/documentation/stable/index.html

  kamada_kawai_layout
  spring_layout → Fruchterman Reingold
  spectral_layout
  pydot_layout, graphviz_layout

- iGraph python.igraph.org/en/stable/

  layout_drl
  layout_fruchterman_reingold
  layout_graphopt
  layout_kamada_kawai
  layout_lgl, layout_mds
  layout_umap → experimental ☺

- UMAP https://umap-learn.readthedocs.io/en/latest/

  UMAP

UNIVERSITÀ DEGLI STUDI DI PADOVA

UMAP example in R ☺

#metoo 2018 semantic network

umap(*A*, n_neighbors=30, metric='cosine', min_dist=0.5)

UMAP example in R ☺

enhancing the role of communities

umap($\boldsymbol{A} \cdot (1 + \alpha\, \boldsymbol{C}^T \boldsymbol{C}\,)$, n_neighbors=30, metric='cosine', min_dist=0.5)

$\alpha = 9$

adjacency matrix $\boldsymbol{A}$ whose elements inside a community are multiplied by $(1+\alpha)$; communities $\boldsymbol{C}$ identified by Louvain

Università degli Studi di Padova

Gephi

Geo layout + Label adjust + Contraction/Expansion

Takeaways

for your projects

- many layout algorithms are available in Python
- UMAP seems the best, but you never know
- Gephi is an alternative useful tool, but largely based on manual intervention
- use a combination of the two for best results
- a good project has a readable network (or wordcloud) clearly showing the role of communities