The Minimum Wiener Connector Problem

Natali Ruchansky, Francesco Bonchi, David García-Soriano, Francesco Gullo, Nicolas Kourtellis

ACM SIGMOD/PODS
31 MAY - 4 JUNE 2015

BOSTON UNIVERSITY
YAHOO! LABS

Infected Patients

Who is the culprit?

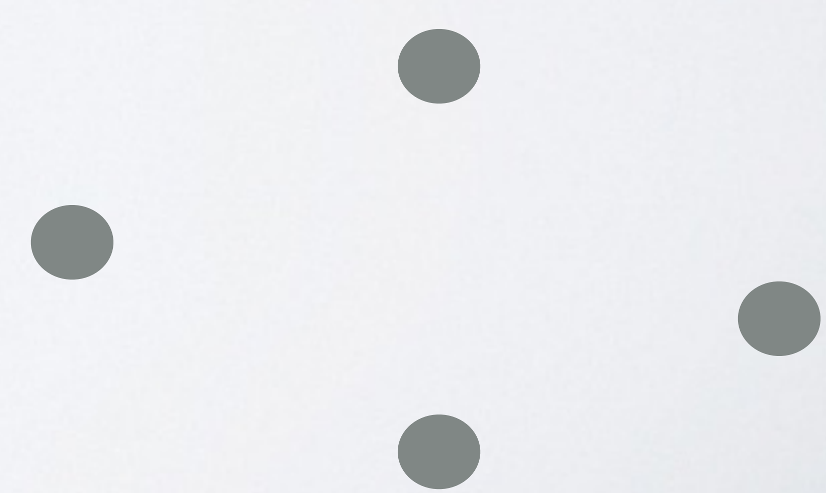
Proteins

Which other proteins participate in pathways?

General Problem

Given a graph $G = (V, E)$ and a set of query vertices $Q \subseteq V$, find a *small* subgraph H of G that "*explains*" the connections existing among Q.

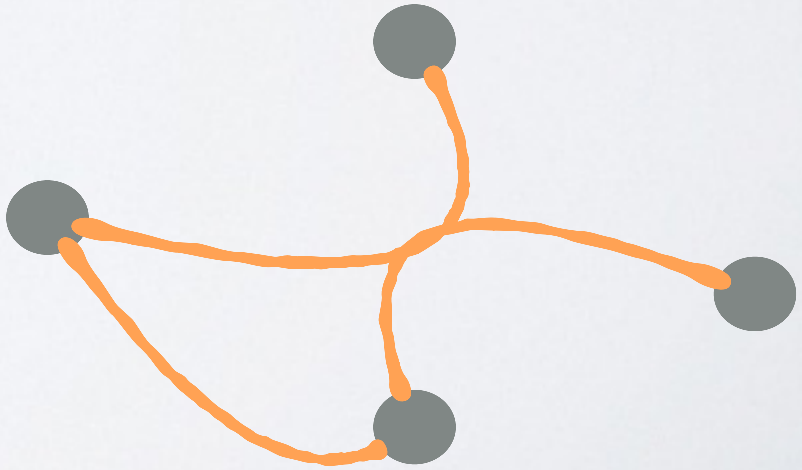
Call this query-dependent graph, H, a ***connector***.



General Problem

Given a graph $G = (V, E)$ and a set of query vertices $Q \subseteq V$,
find a *small* subgraph H of G that
"*explains*" the connections existing among Q.

Call this query-dependent graph, H, a ***connector***.



Related Work

Random-walk

Run a random walk from each query node.

Identify a neighborhood of each node.

Combine neighborhoods.

Search

Search for a subgraph that best meets objective.

Many parameters

Large solutions

Steiner Tree

Find the smallest tree that connects query nodes.

No interpretation

Motivating Observation

A natural sense of closeness in graphs is captured by **short paths.**

Objective

We define a new problem where the objective is to:

minimize the sum of pairwise shortest-path-distances

between nodes **in the connector *H***.

If $d(u, v)$ is the shortest-path distance, we want:

$$\text{minimize} \sum_{(u,v)\in H} d(u, v)$$

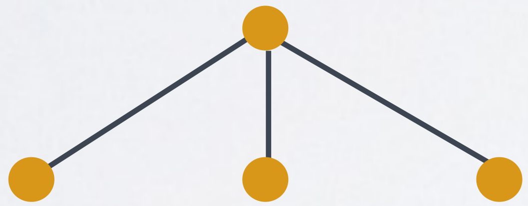
In fact this quantity is called the **Wiener Index**.

Wiener Intuitions

Path is **largest**:

3+2+1+2+1+1= 10

Clique/Star is **smallest**:



2+2+1+2+1+1= 9

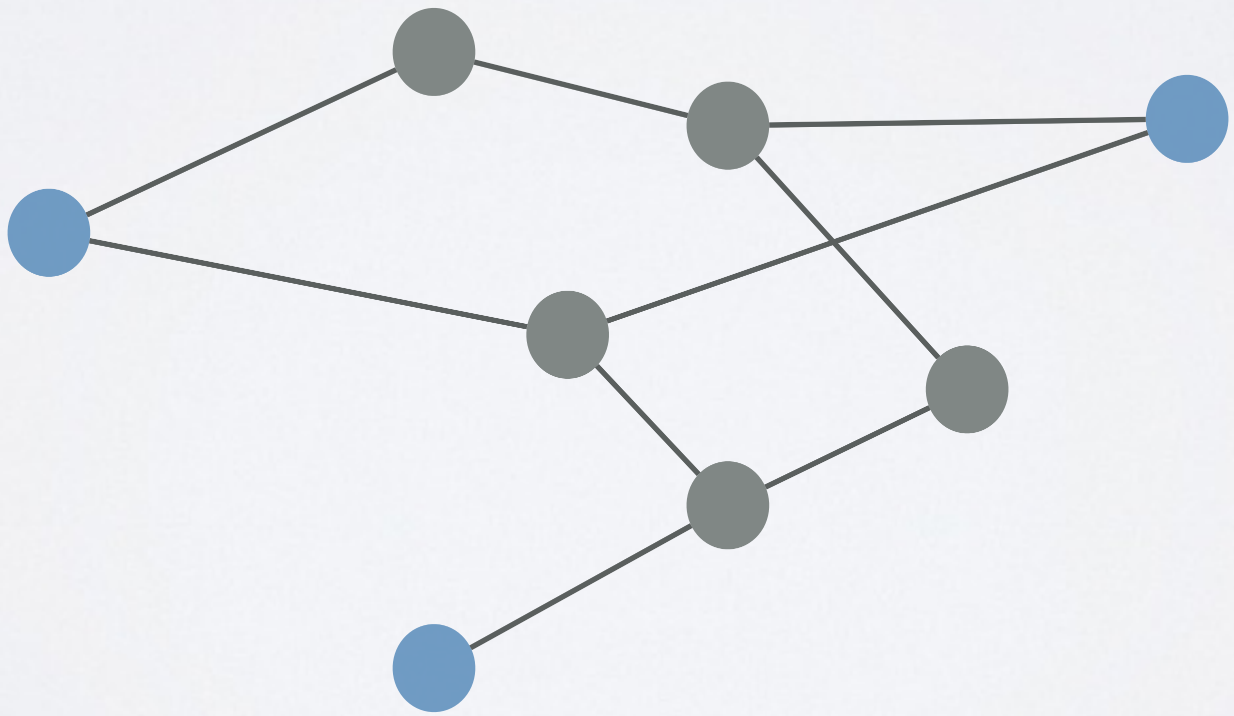
Favors star-shape, closeness.

Provides a numerical feedback of connectedness.

Minimum Wiener Connector

Given a graph $G = (V, E)$ and a query set $Q \subseteq V$, find a connector H^* for Q in G with smallest Wiener index.

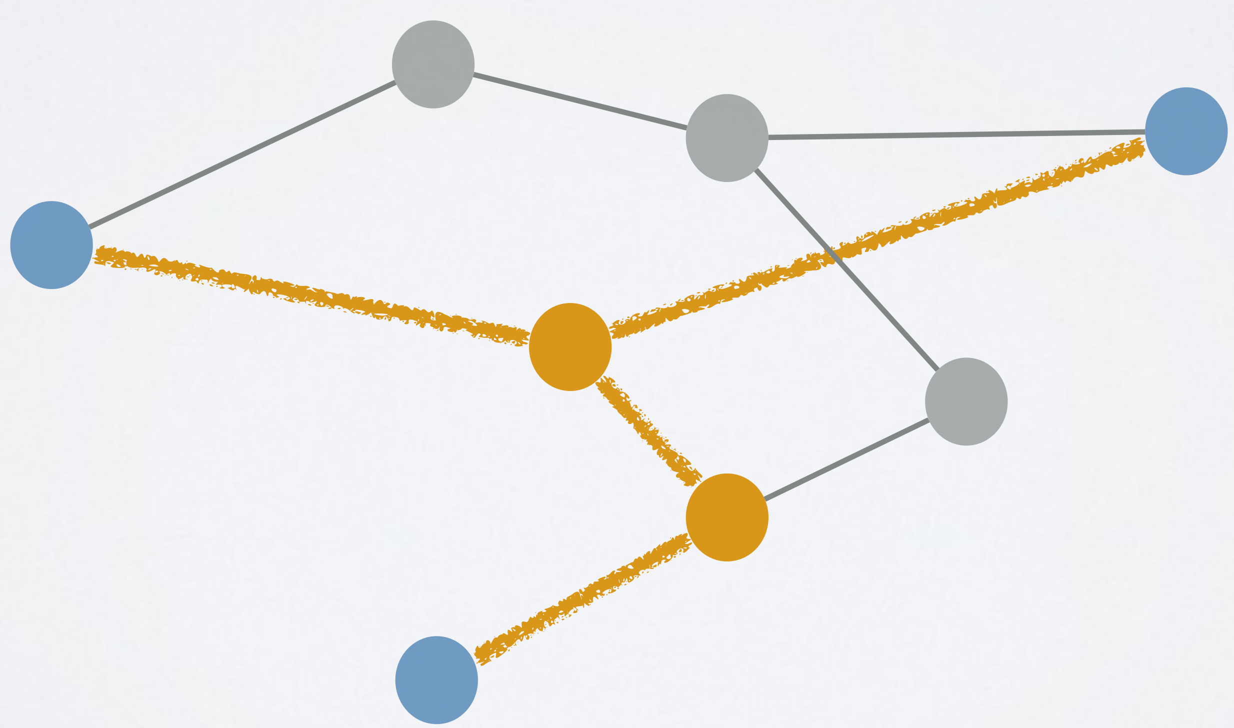
Call H^* the **minimum Wiener connector.**



Minimum Wiener Connector

Given a graph $G = (V, E)$ **and a query set** $Q \subseteq V$**, find a connector** H^* **for** Q **in** G **with smallest Wiener index.**

Call H^* the **minimum Wiener connector.**



Minimum Wiener Connector

Given a graph $G = (V, E)$ **and a query set** $Q \subseteq V$**, find a connector** H^* **for** Q **in** G **with smallest Wiener index.**

Call H^* the **minimum Wiener connector.**

There is no explicit size constraint, but rewriting

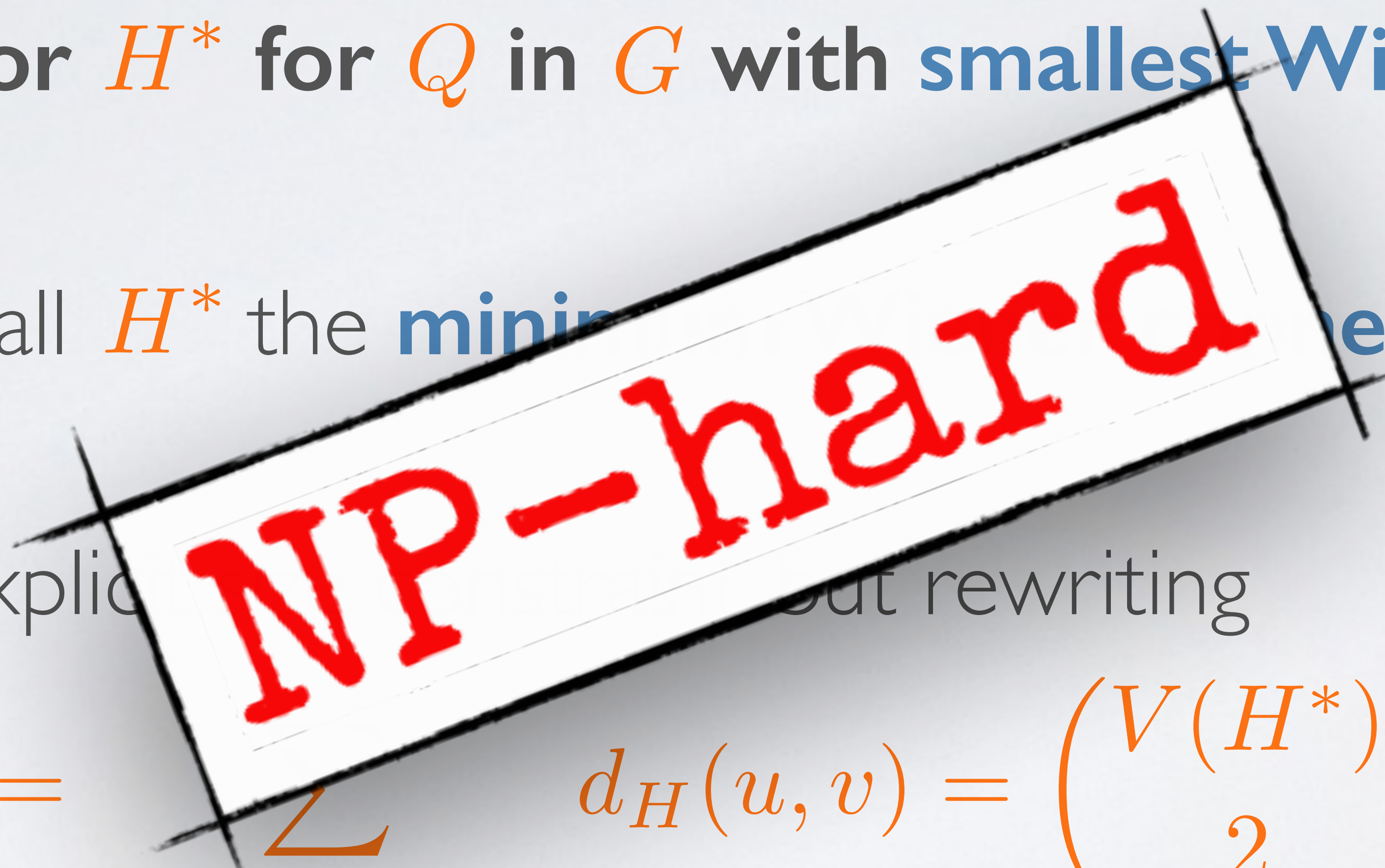
$$W(H^*) = \sum_{\{u,v\} \subseteq V(H^*)} d_H(u, v) = \binom{V(H^*)}{2} * \text{average } d$$

uncovers a tradeoff between **size** and **average distance**

Minimum Wiener Connector

Given a graph $G = (V, E)$ **and a query set** $Q \subseteq V$**, find a connector** H^* **for** Q **in** G **with smallest Wiener index.**

Call H^* the **mini… …ector.**

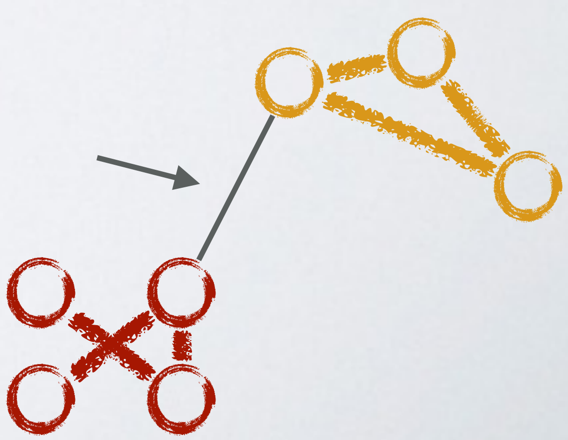


There is no explic… …ut rewriting

$$W(H^*) = \sum_{\{u,v\}\subseteq V(H^*)} d_H(u,v) = \binom{V(H^*)}{2} * \text{average } d$$

uncovers a tradeoff between **size** and **average distance**

Summary Of Results



Summary Of Results

With the Wiener Index as our objective, we propose:

a **constant factor approximation** algorithm that runs in $\tilde{O}(|Q|\ |E|)$

Using this we find solutions that are aside from being **close to optimal**:

small, **meaningful**, and **amenable to visualization**.

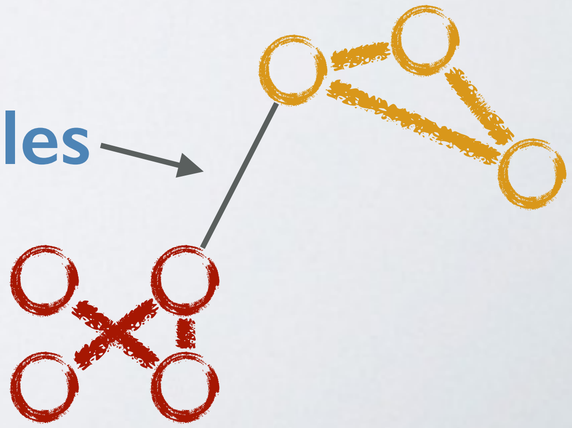
For query nodes belonging to the **same** community:

connector contains nodes of **high centrality**

For query nodes from **different** communities:

connector contains nodes that span **structural holes**

(incident to edges that bridge communities)



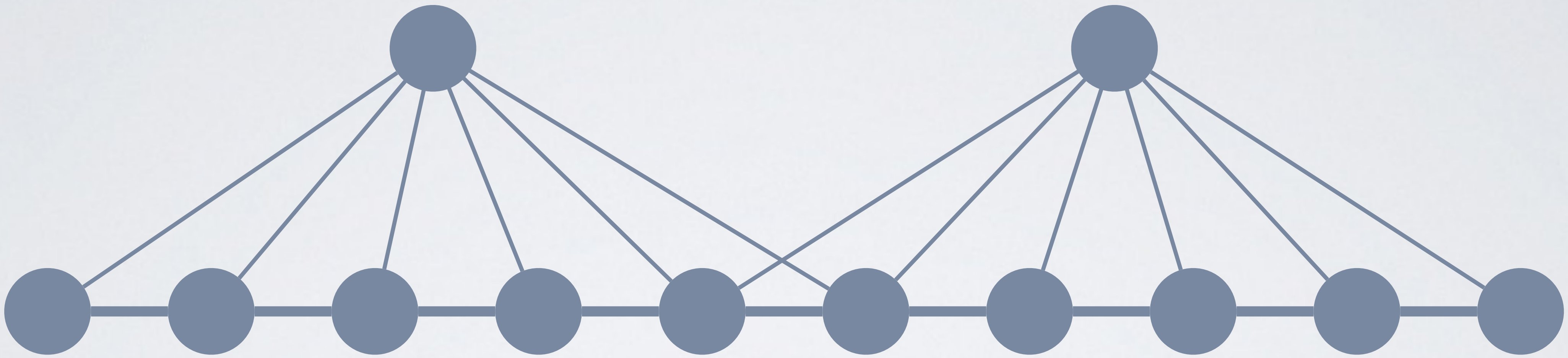
How Do We Find The Minimum Wiener Connector?

No. Not The Steiner Tree

Steiner Tree: Given a graph $G = (V, E)$ and a set of query nodes (terminals) $Q \subseteq V$, find the smallest tree connecting all terminals.

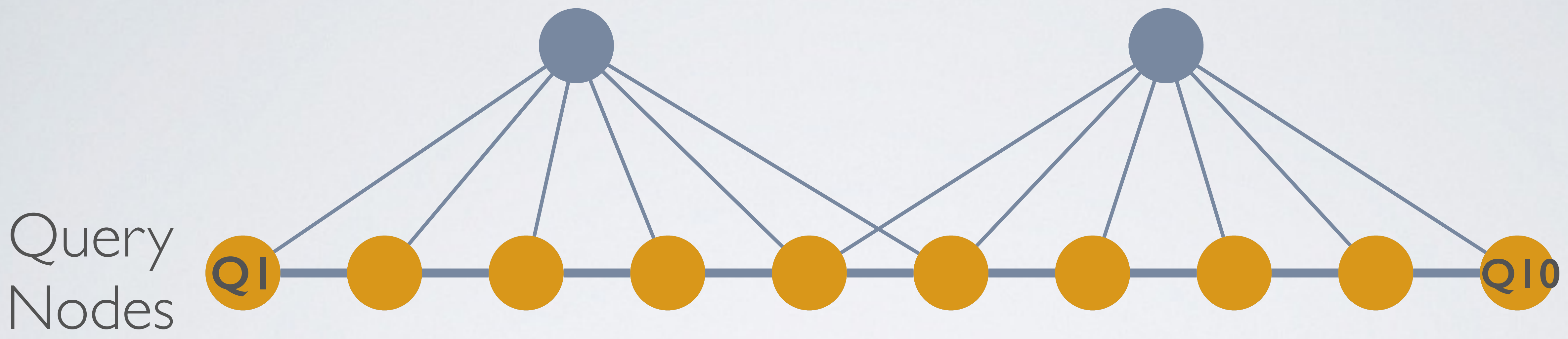
Minimizing the number of edges will **not** necessarily result in the smallest Wiener Index!

Steiner vs Wiener



Optimal Solutions	Steiner Cost	Wiener Cost

Steiner vs Wiener

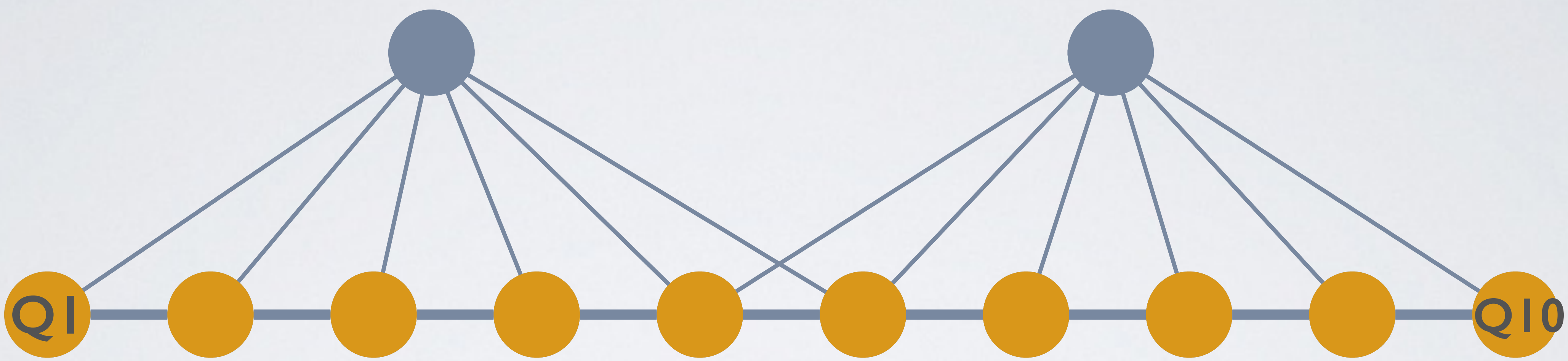


Optimal Solutions

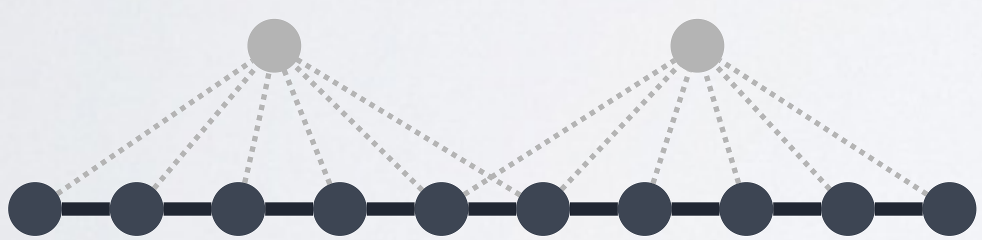
Steiner Cost	Wiener Cost

Steiner vs Wiener

Query Nodes

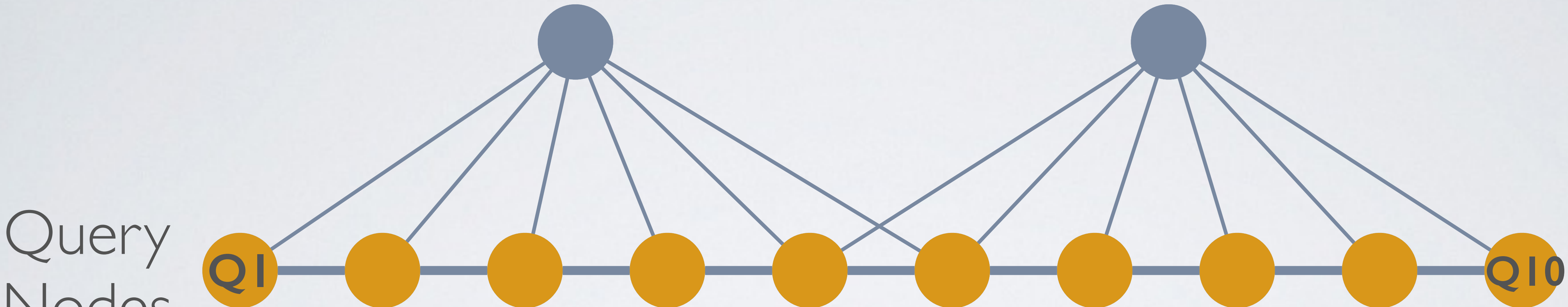


Optimal Solutions

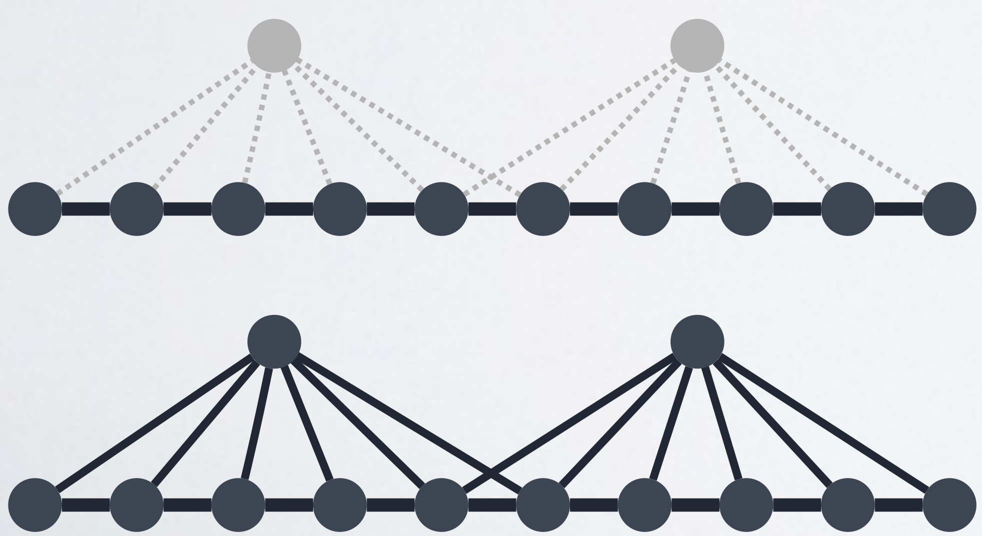


Steiner Cost	Wiener Cost
9	165

Steiner vs Wiener



Optimal Solutions



Steiner Cost	Wiener Cost
9	165
21	**142**

Relaxations

Original Objective **Relaxed Objective**

Reduce to classic Steiner Tree with carefully constructed edge weights.

Relaxations

Original Objective | **Relaxed Objective**

All pairwise distances → Distances from a root r

Reduce to classic Steiner Tree with carefully constructed edge weights.

Relaxations

Original Objective		**Relaxed Objective**
All pairwise distances	$\rightarrow$	Distances from a root r
Measure distance in H	$\rightarrow$	Measure distance in G

Reduce to classic Steiner Tree with carefully constructed edge weights.

Relaxations

Original Objective	**Relaxed Objective**
All pairwise distances	Distances from a root r
Measure distance in H	Measure distance in G
Product in objective	Linear objective

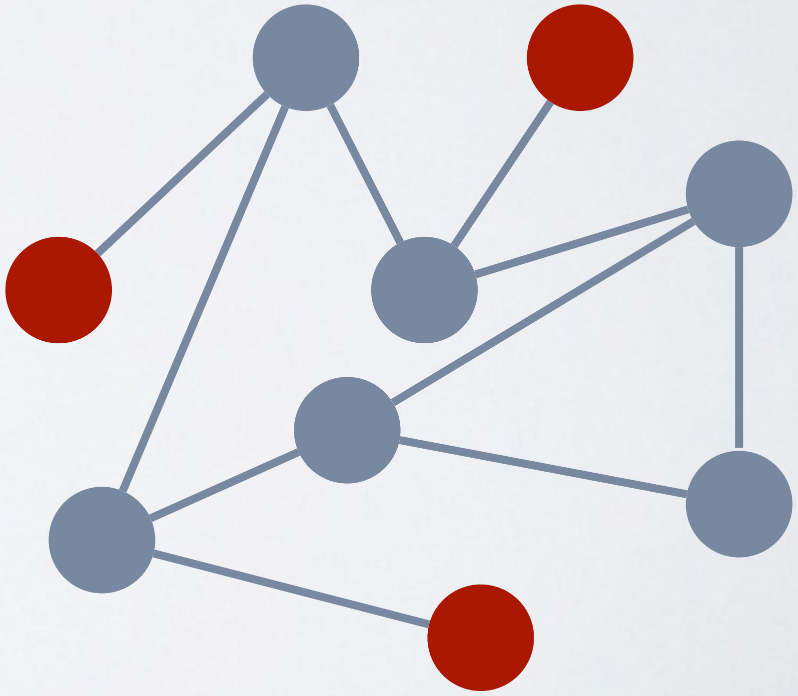
Reduce to classic Steiner Tree with carefully constructed edge weights.

Relaxations

Original Objective	**Relaxed Objective**
All pairwise distances	Distances from a root r
Measure distance in H	Measure distance in G
Product in objective	Linear objective
Node weights	Edge weights

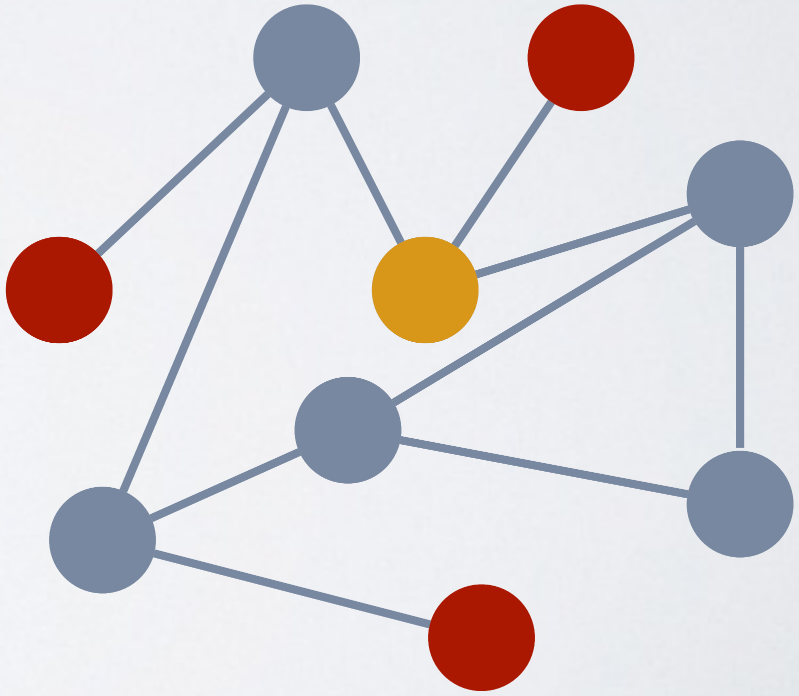
Reduce to classic Steiner Tree with carefully constructed edge weights.

WienerSteiner(G,Q)



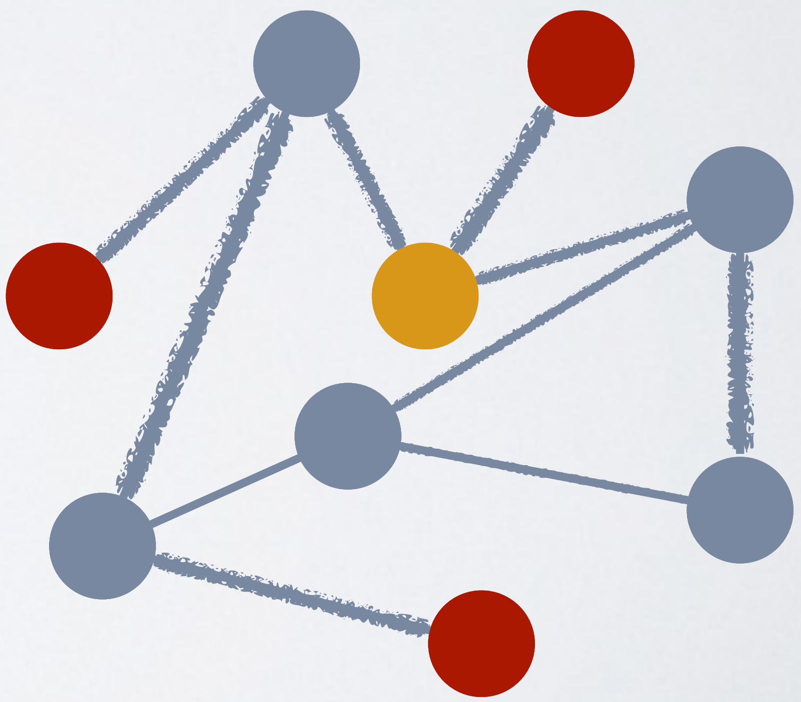
`WienerSteiner(G,Q)`

- For each vertex $r \in V$



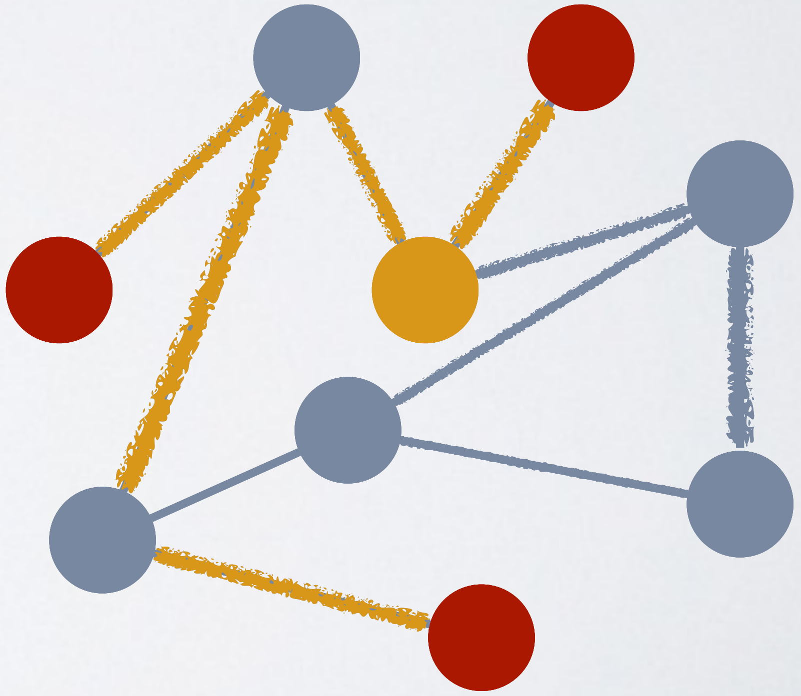
`WienerSteiner(G,Q)`

- For each vertex $r \in V$
  1. Compute $d_G(u, v)$ from r to each vertex u
  2. Construct an edge-weighted graph



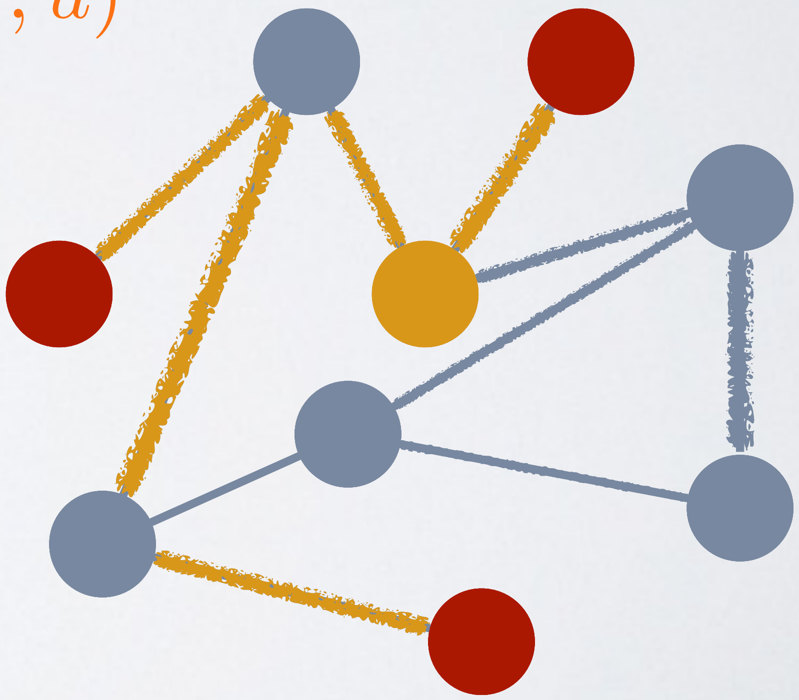
`WienerSteiner(G,Q)`

- For each vertex $r \in V$
    1. Compute $d_G(u, v)$ from r to each vertex u
    2. Construct an edge-weighted graph
    3. Find an approximate Steiner tree S_r^*



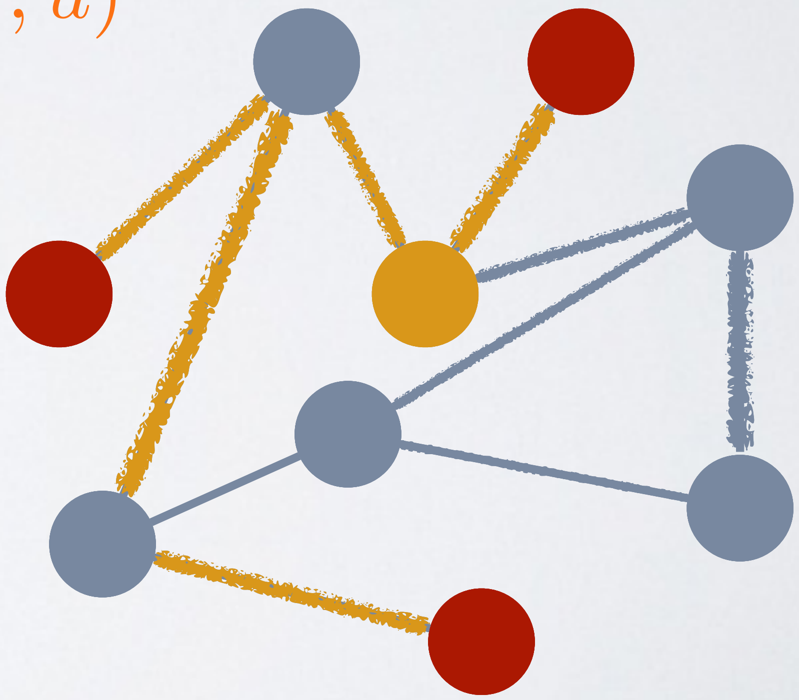
`WienerSteiner(G,Q)`

- For each vertex $r \in V$
    1. Compute $d_G(u, v)$ from r to each vertex u
    2. Construct an edge-weighted graph
    3. Find an approximate Steiner tree S_r^*
    4. Check for paths where $d_G(r, u) < d_{S^*}(r, u)$



WienerSteiner(G,Q)

- For each vertex $r \in V$
  1. Compute $d_G(u, v)$ from r to each vertex u
  2. Construct an edge-weighted graph
  3. Find an approximate Steiner tree S_r^*
  4. Check for paths where $d_G(r, u) < d_{S^*}(r, u)$
- Pick S_r^* that minimizes $W(S^*)$



Case Studies

Case Study 1: Karate Club

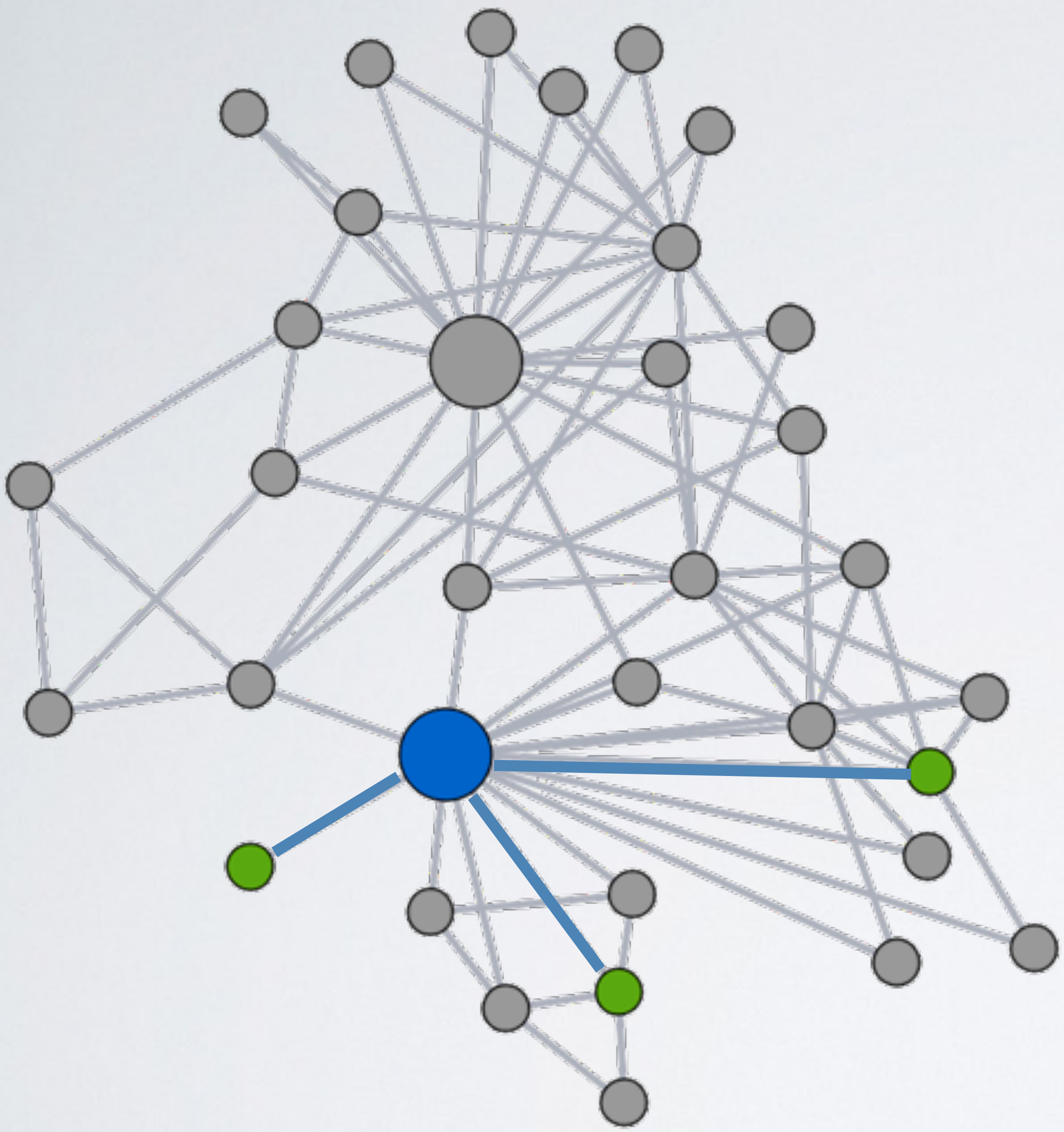
Two clusters around each karate master.

Few nodes with mixed loyalty.

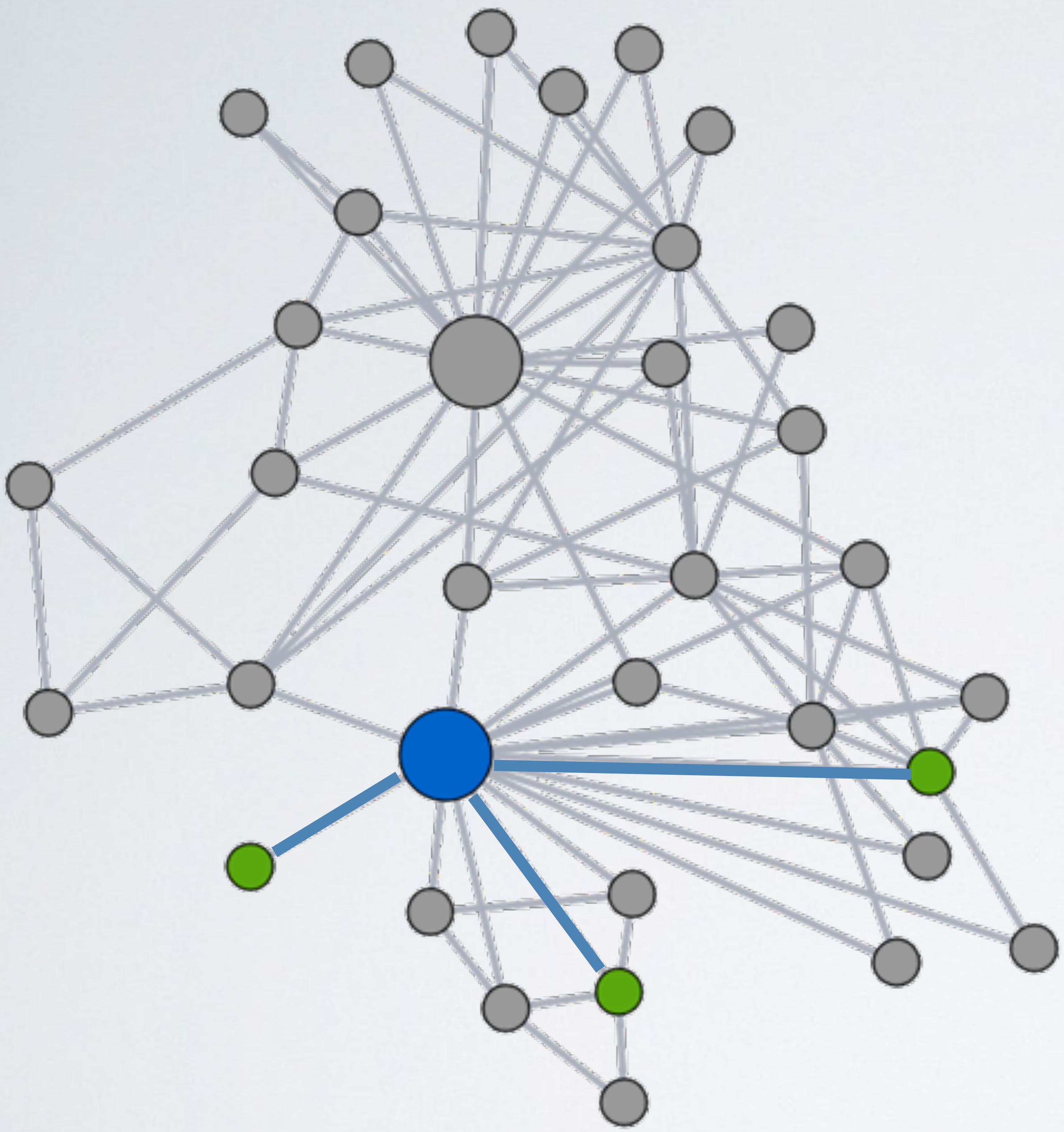
By querying arbitrary nodes, can we learn about their loyalty without any outside meta information?

Same Community

Same Community

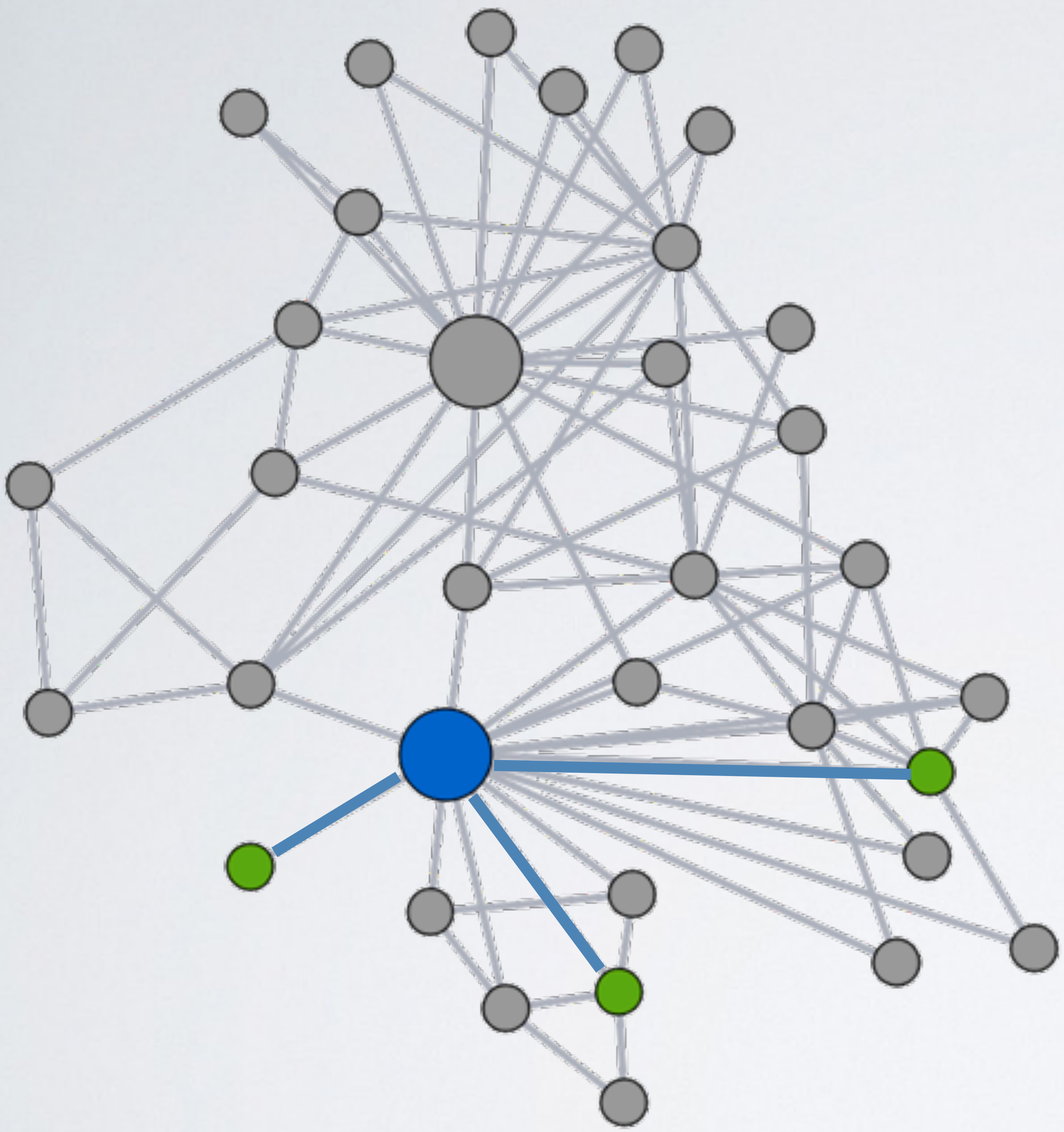


Same Community



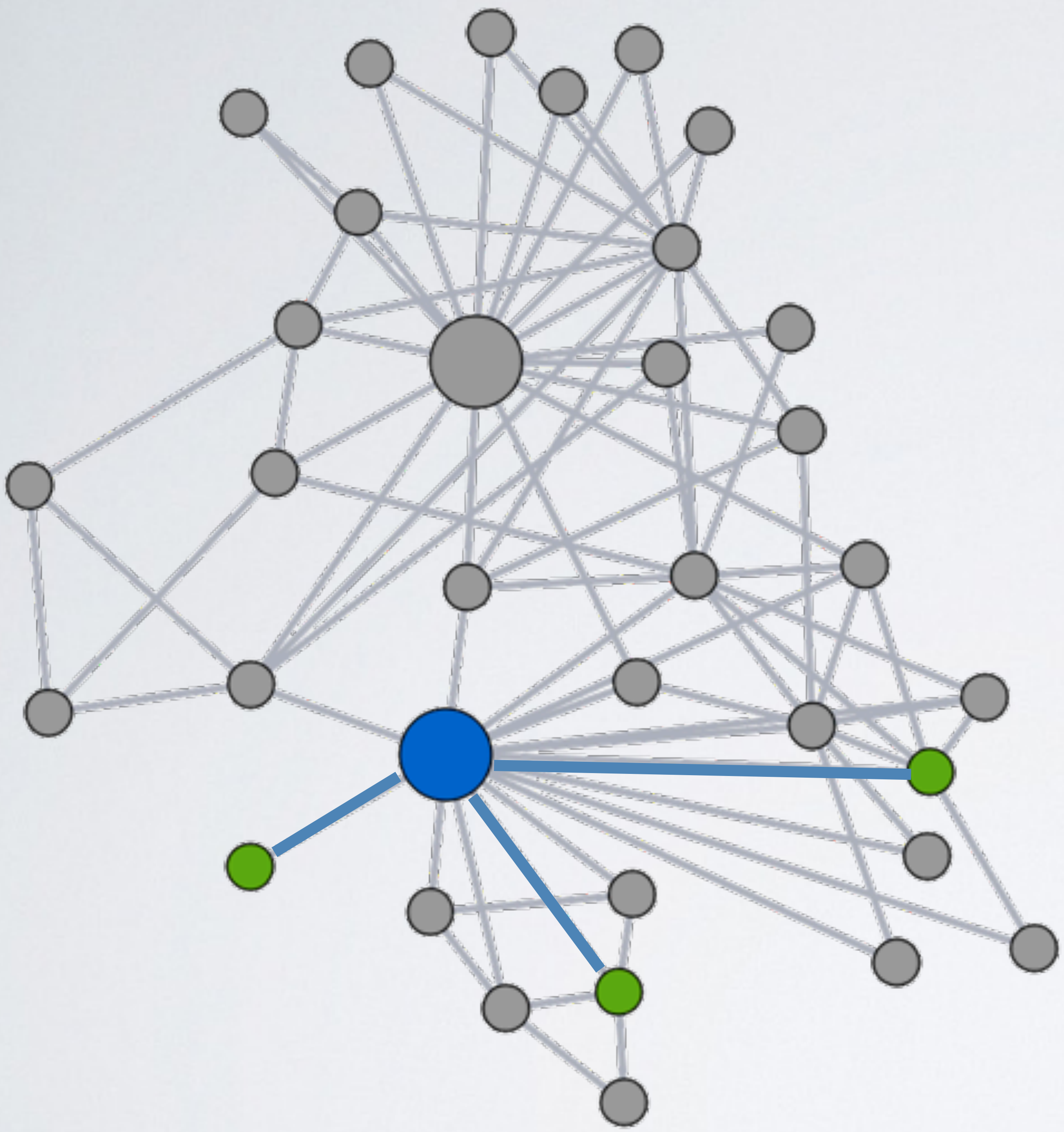
Different Communities

Same Community

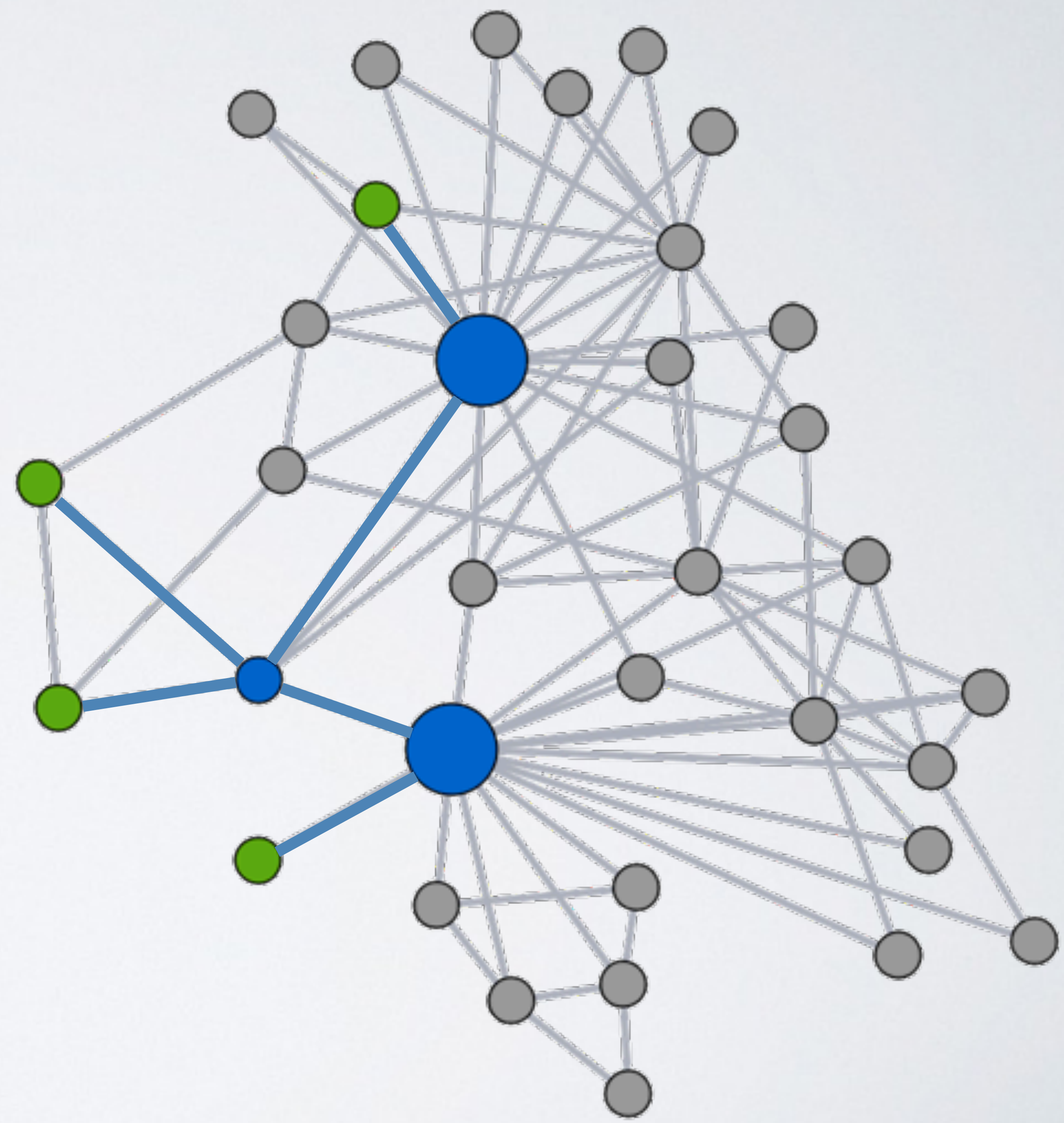


Different Communities

Same Community



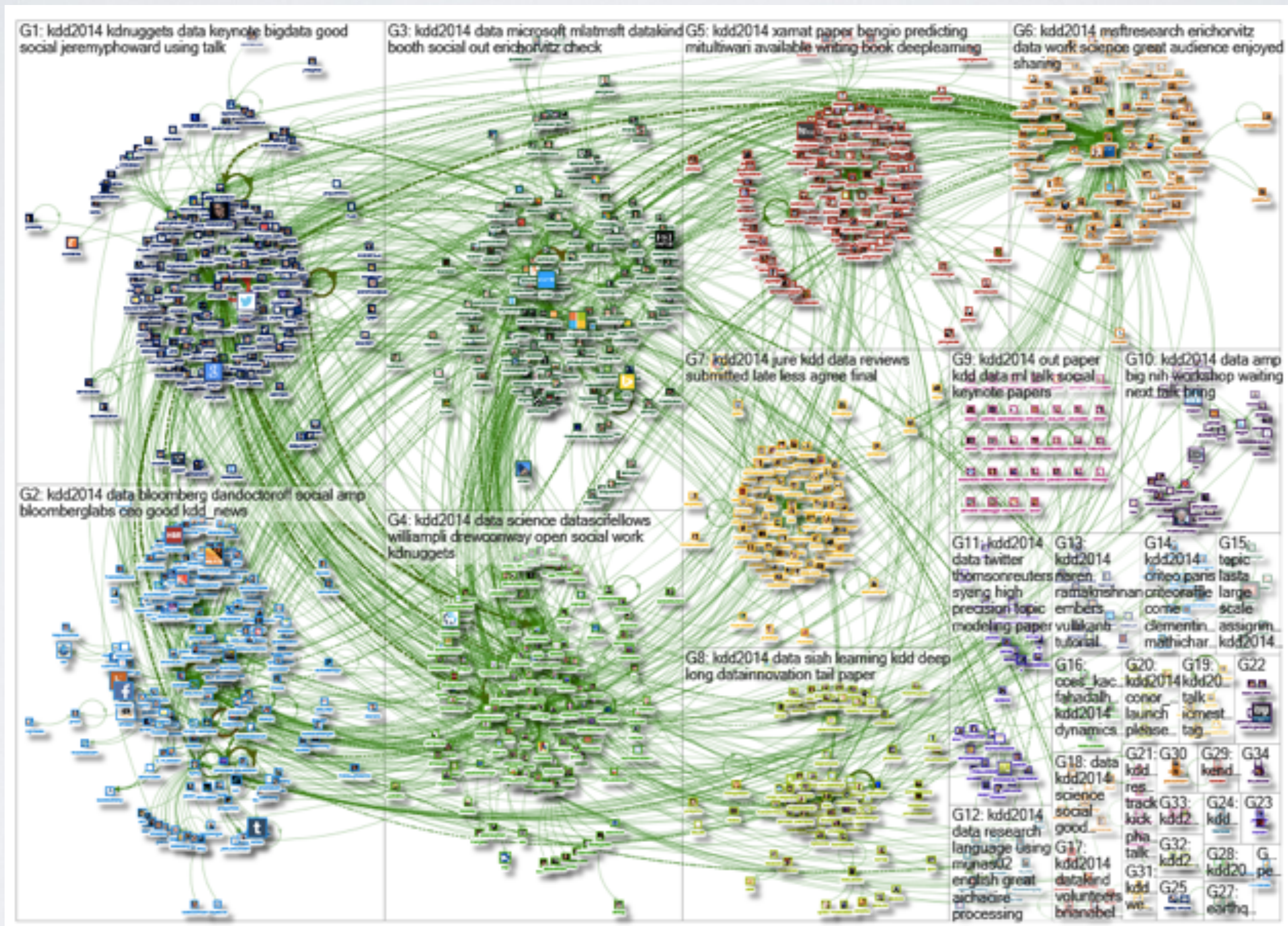
Different Communities



Case Study 2: KDD Tweets

KDD 2014 Tweets

Graph of Twitter users taking part in KDD 2014, with an edge between replies or mentions.



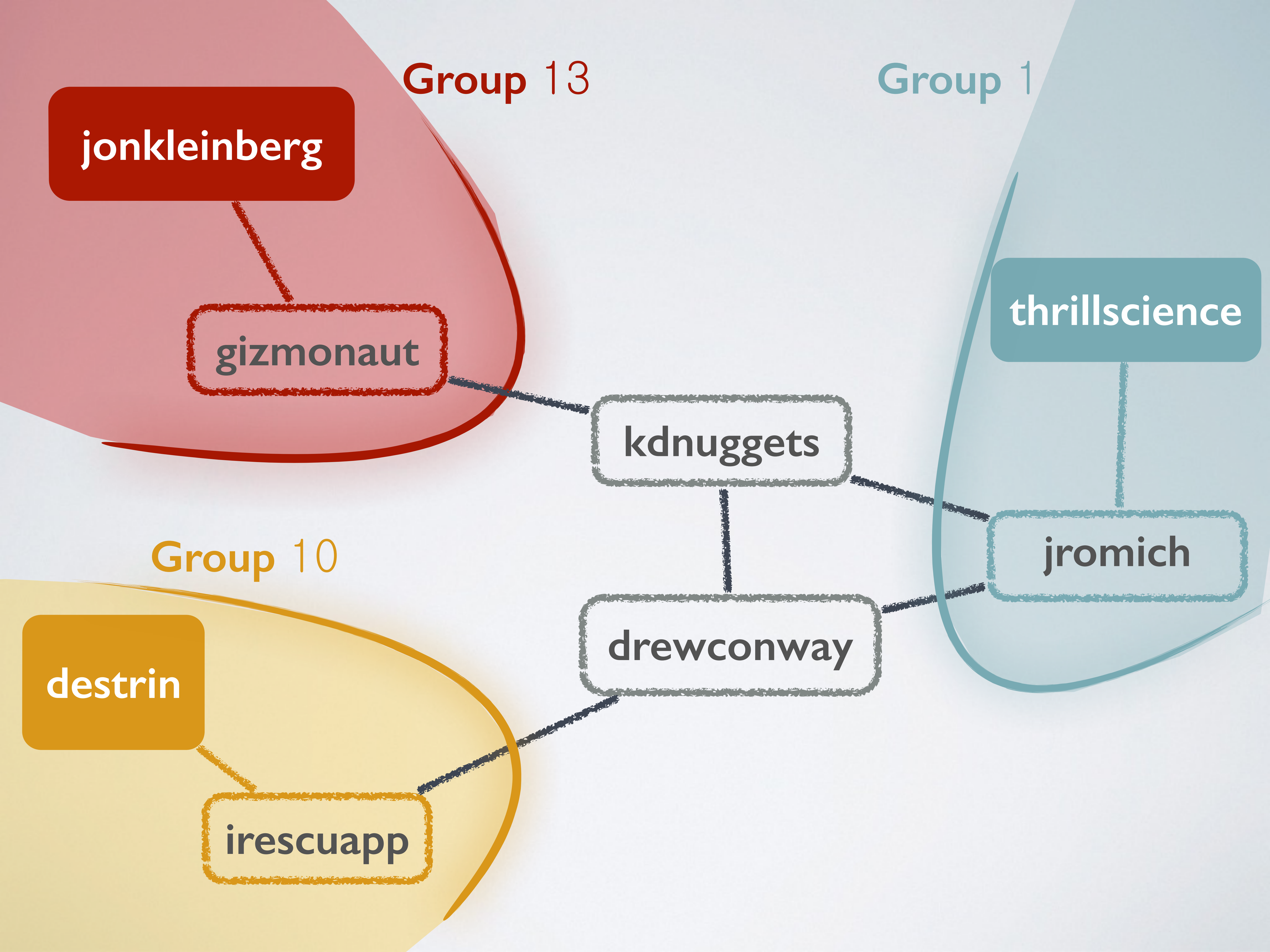
Clustered into 10 communities.

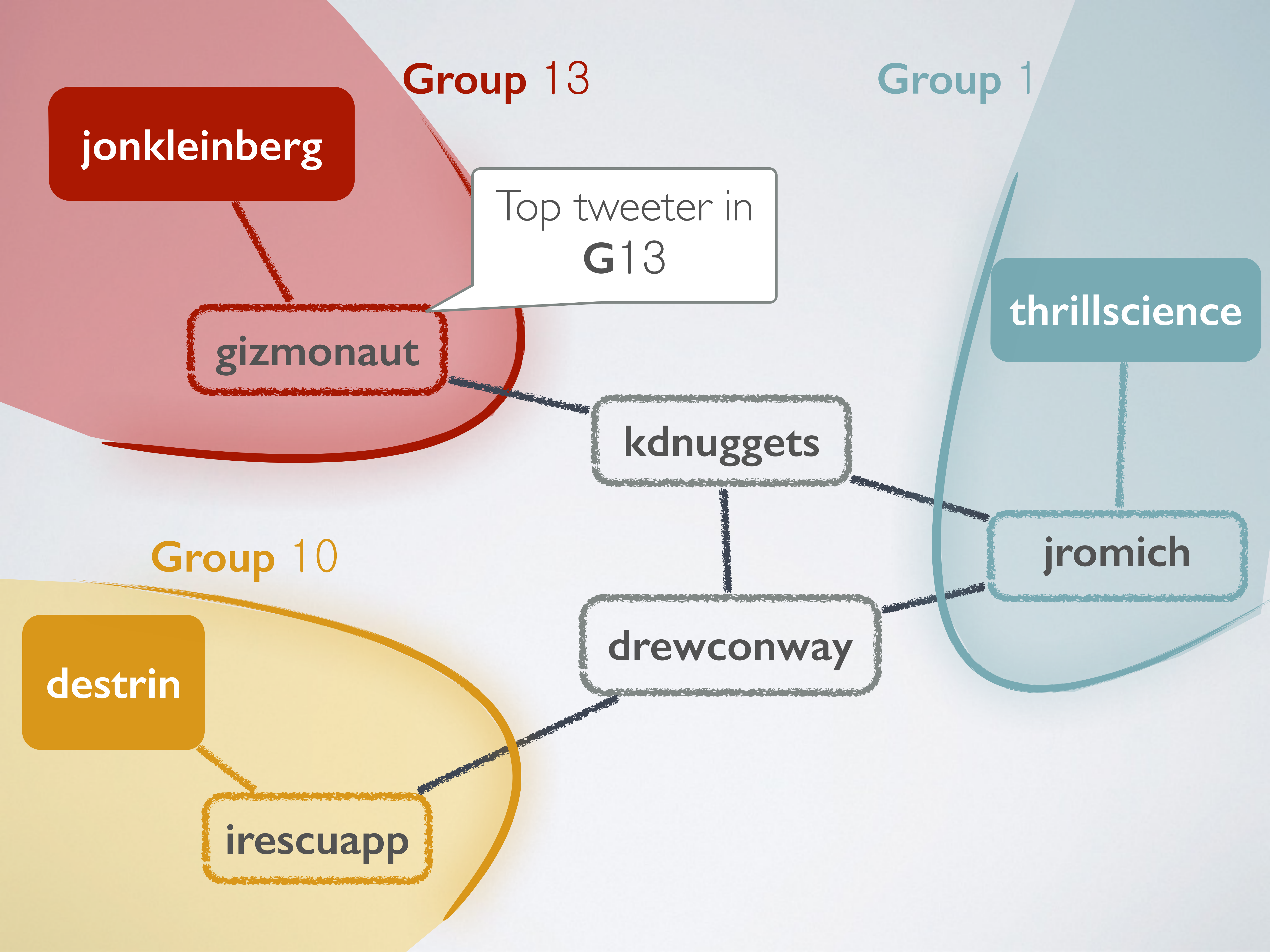
https://nodexlgraphgallery.org/

jonkleinberg

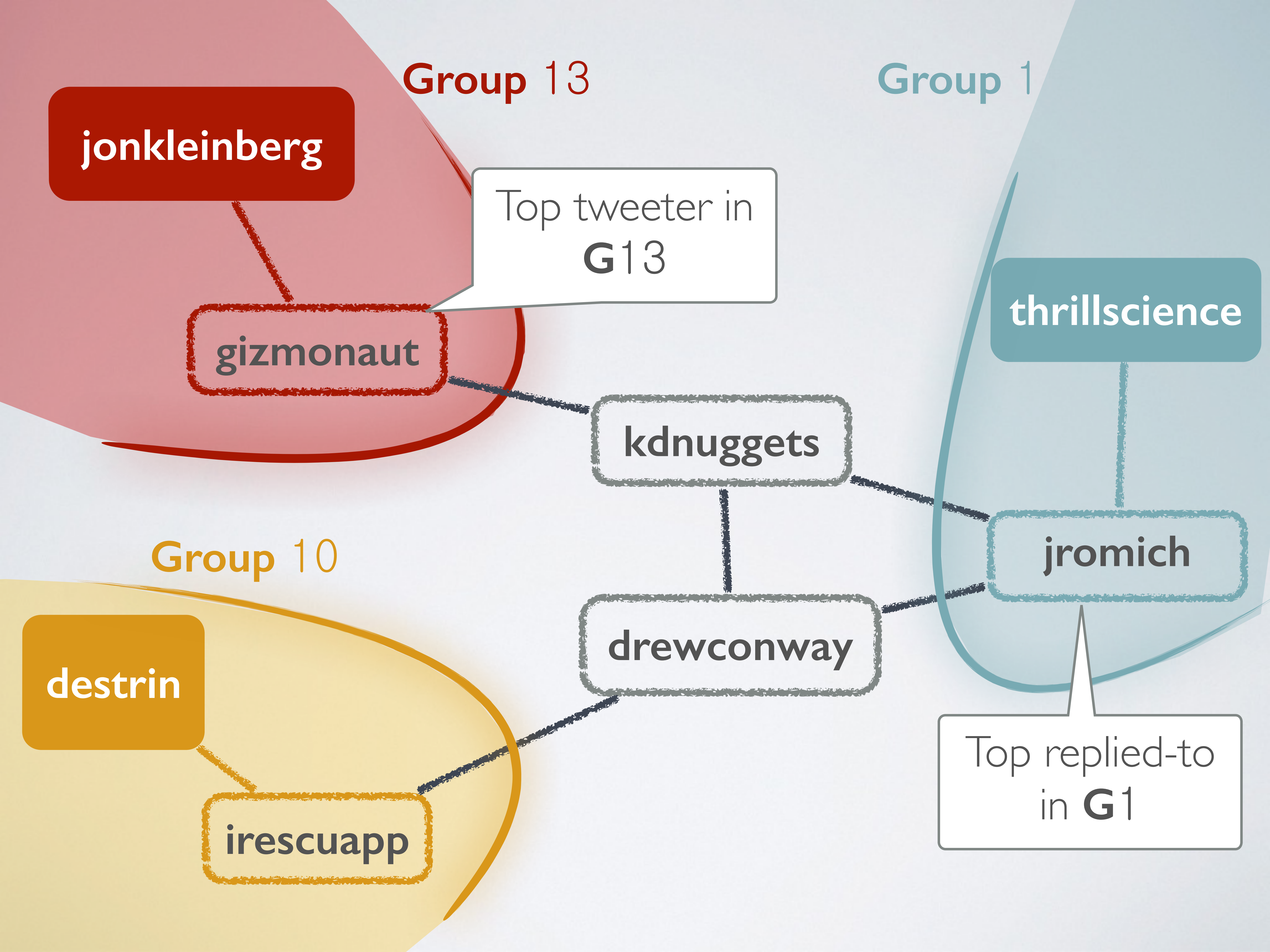
thrillscience

destrin

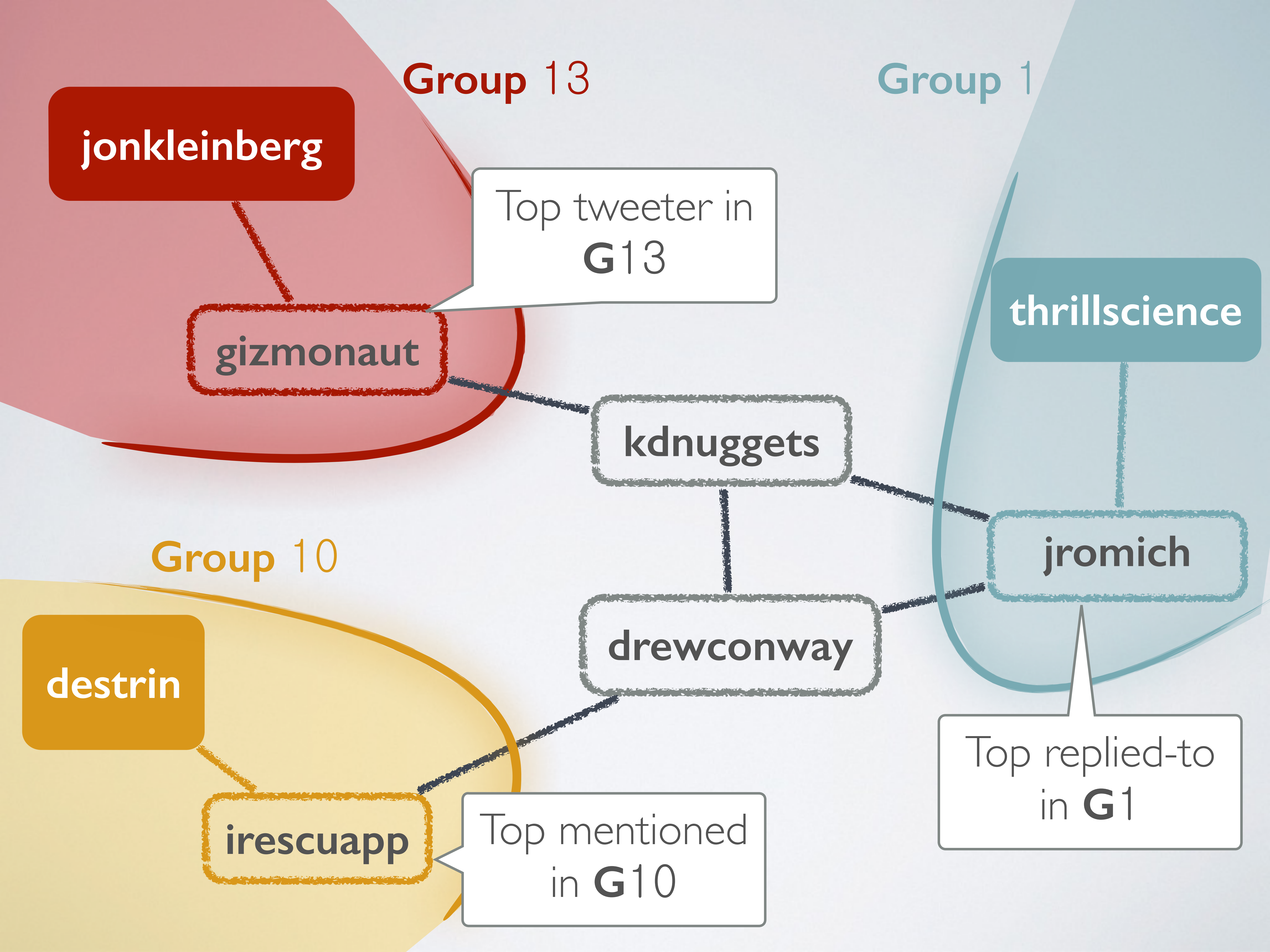

Group 13
Group 1
jonkleinberg
gizmonaut
thrillscience
kdnuggets
jromich
Group 10
drewconway
destrin
irescuapp

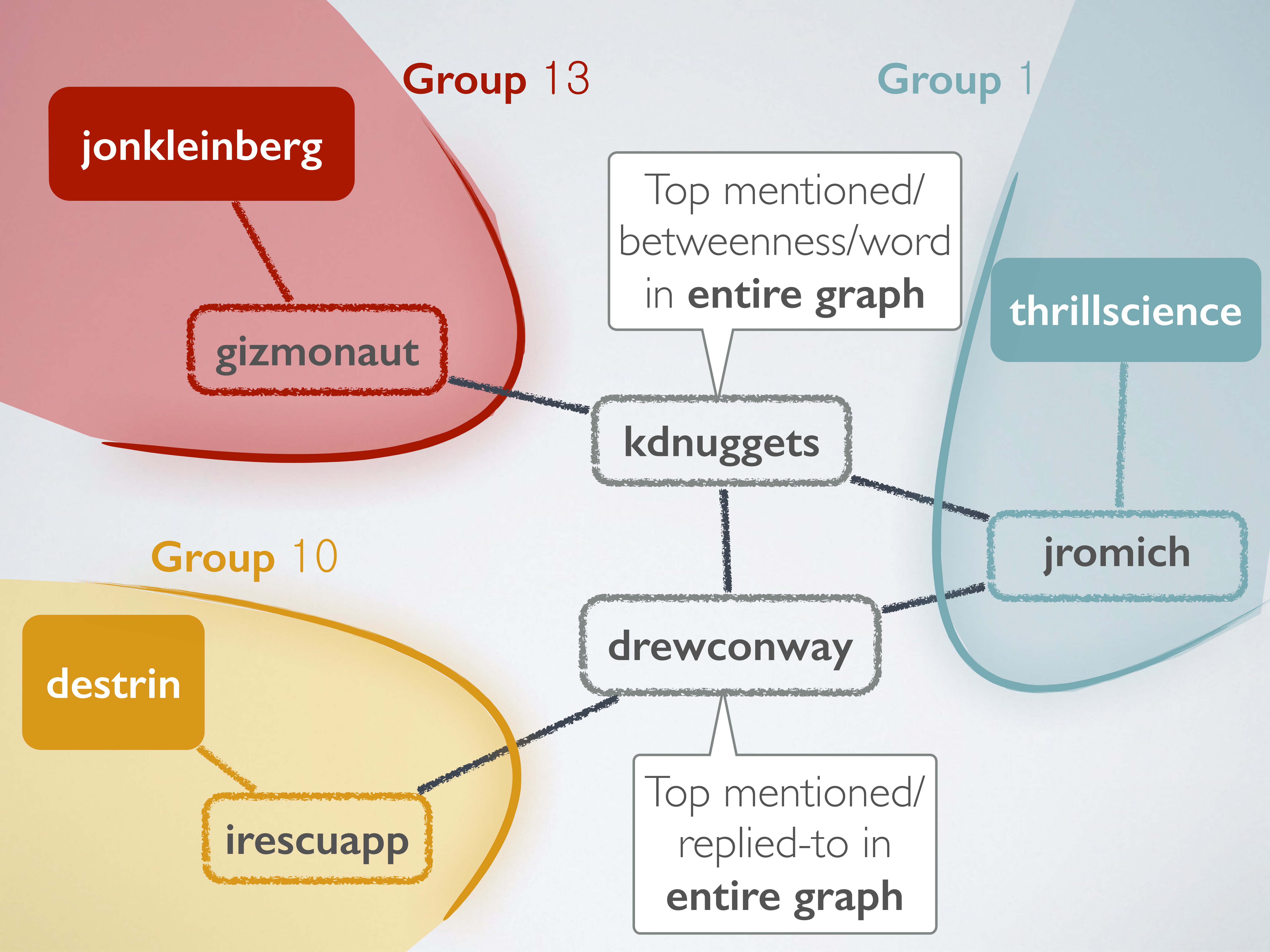


Group 13
Group 1
jonkleinberg
Top tweeter in G13
gizmonaut
thrillscience
kdnuggets
jromich
Group 10
drewconway
destrin
irescuapp



Group 13
Group 1
jonkleinberg
Top tweeter in G13
thrillscience
gizmonaut
kdnuggets
jromich
Group 10
drewconway
destrin
Top replied-to in G1
irescuapp


Group 13
Group 1
jonkleinberg
Top tweeter in G13
thrillscience
gizmonaut
kdnuggets
jromich
Group 10
drewconway
destrin
Top replied-to in G1
irescuapp
Top mentioned in G10


Group 13
Group 1
jonkleinberg
Top mentioned/ betweenness/word in **entire graph**
thrillscience
gizmonaut
kdnuggets
jromich
Group 10
drewconway
destrin
irescuapp
Top mentioned/ replied-to in **entire graph**

Case Study 3: PPI Network

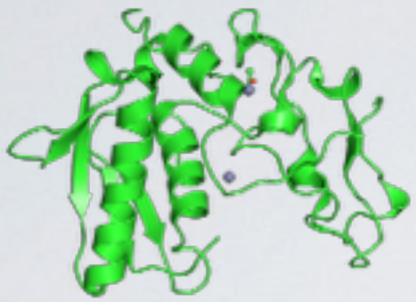
Biology Dataset

Protein-Protein-Interaction (PPI) network collected from BioGrid3 with 15 312 vertices.

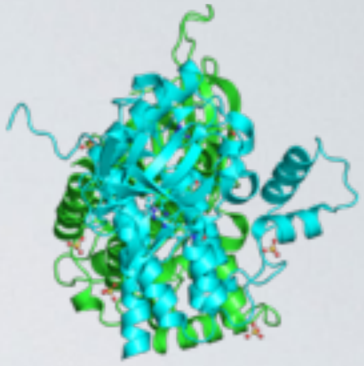
- Do they interact?
- How are they related?
- Which disease are they associated with?
- Which well-known proteins are 'closest' to each?

https://nodexlgraphgallery.org/

BMP1

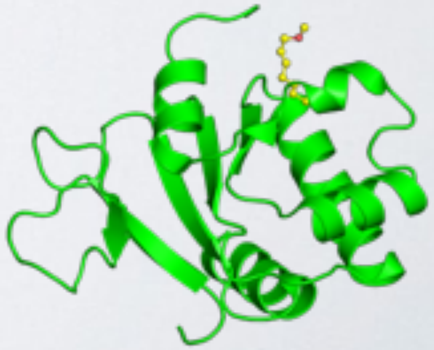


JAK2

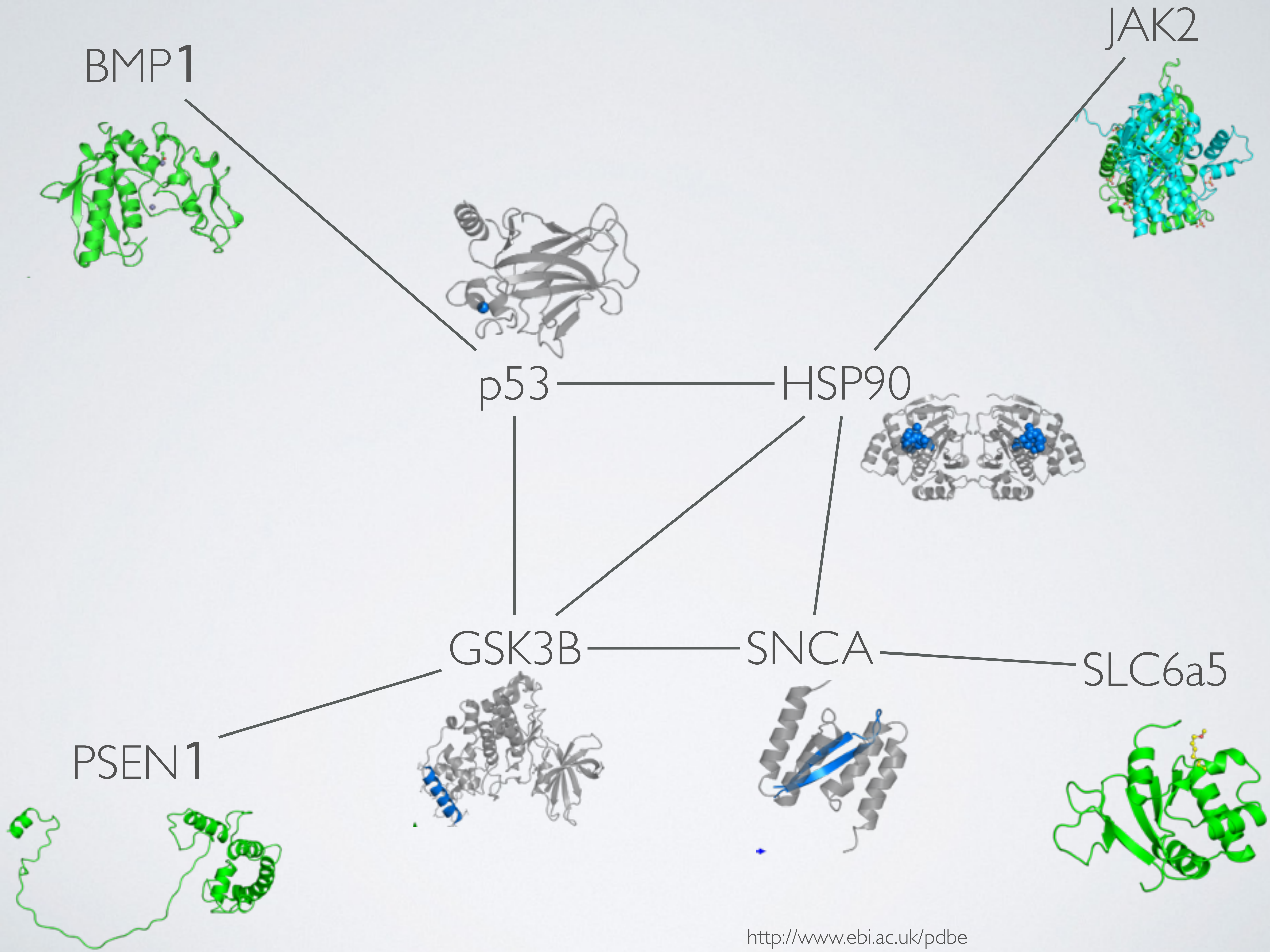


PSEN1

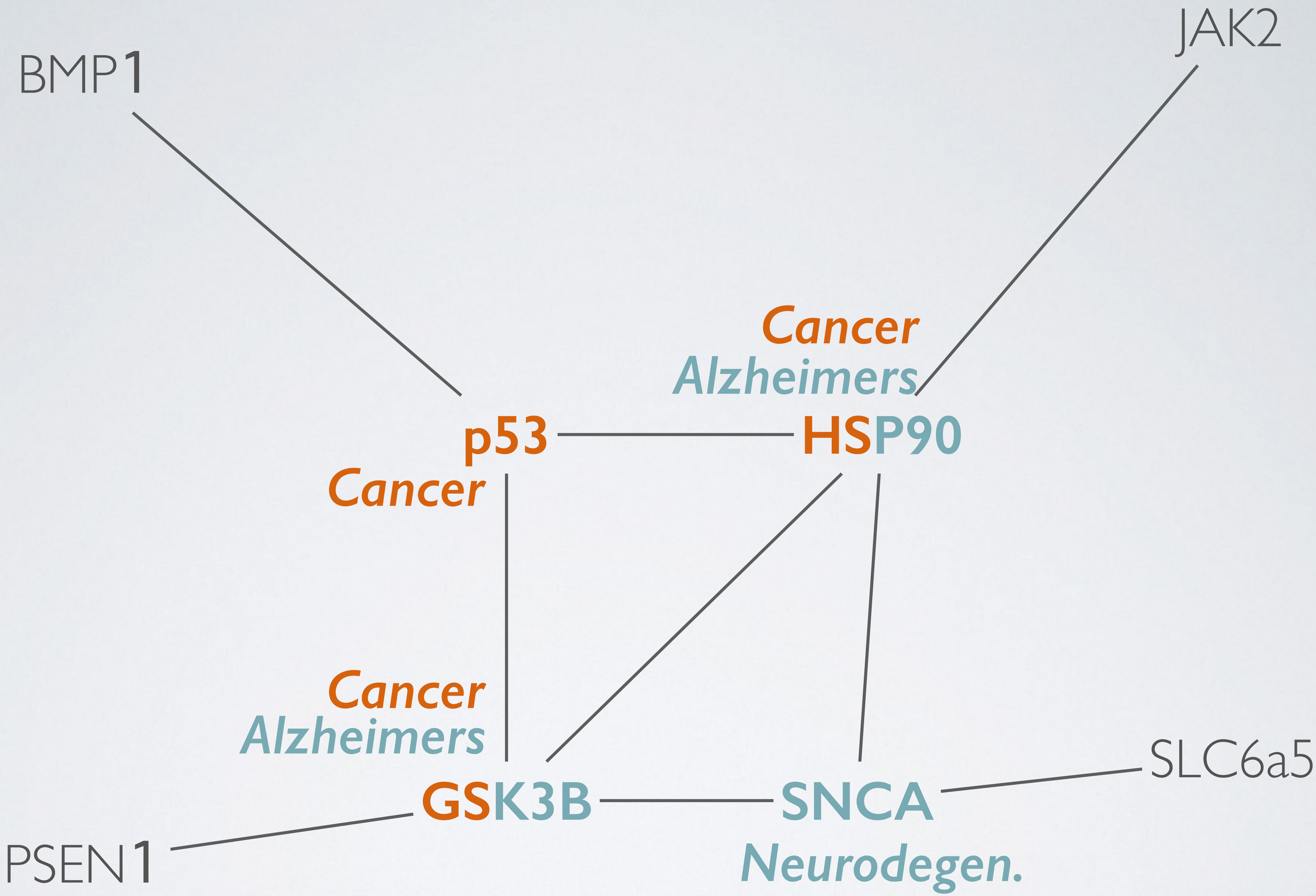
SLC6a5



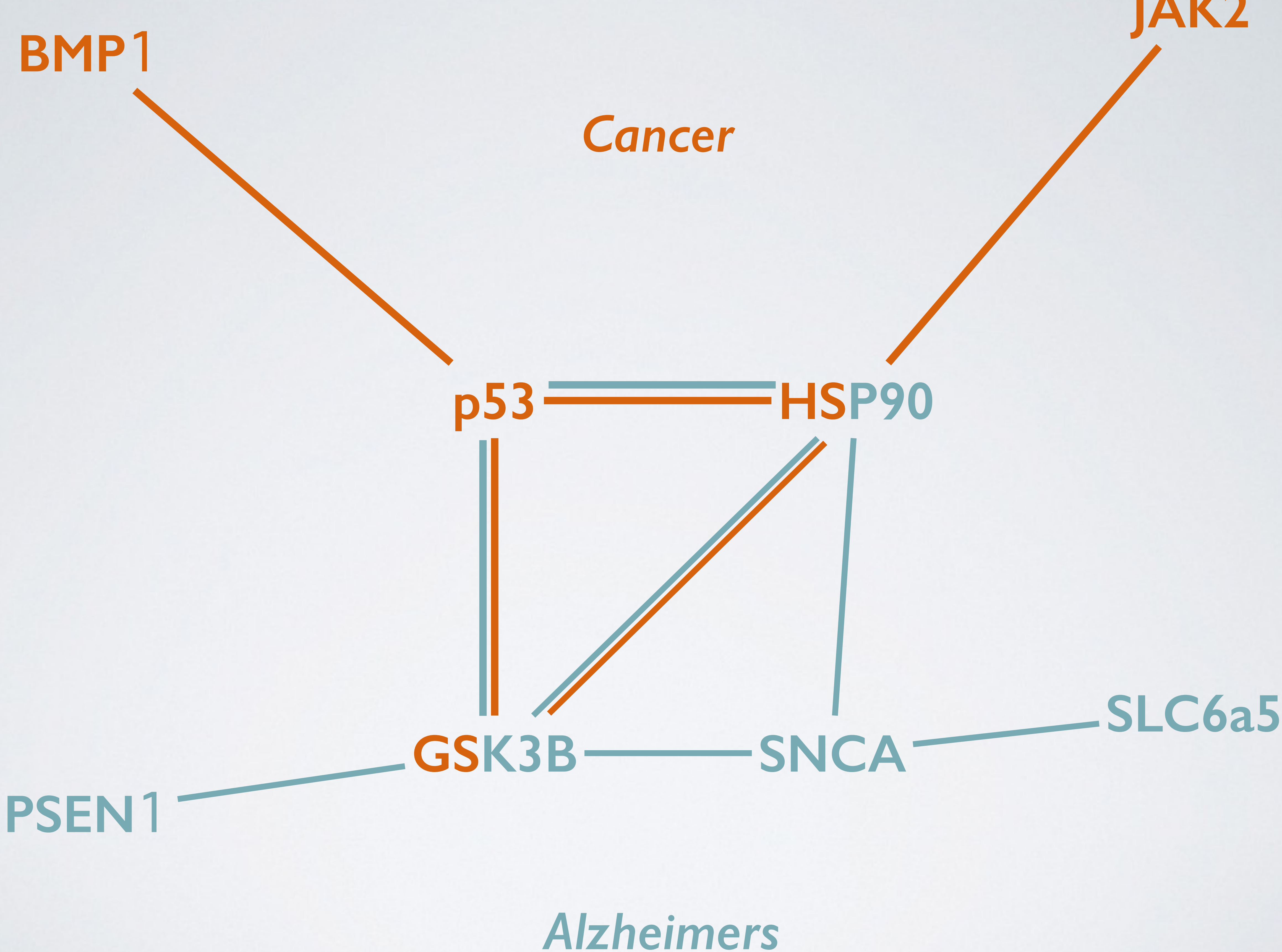
http://www.ebi.ac.uk/pdbe



http://www.ebi.ac.uk/pdbe



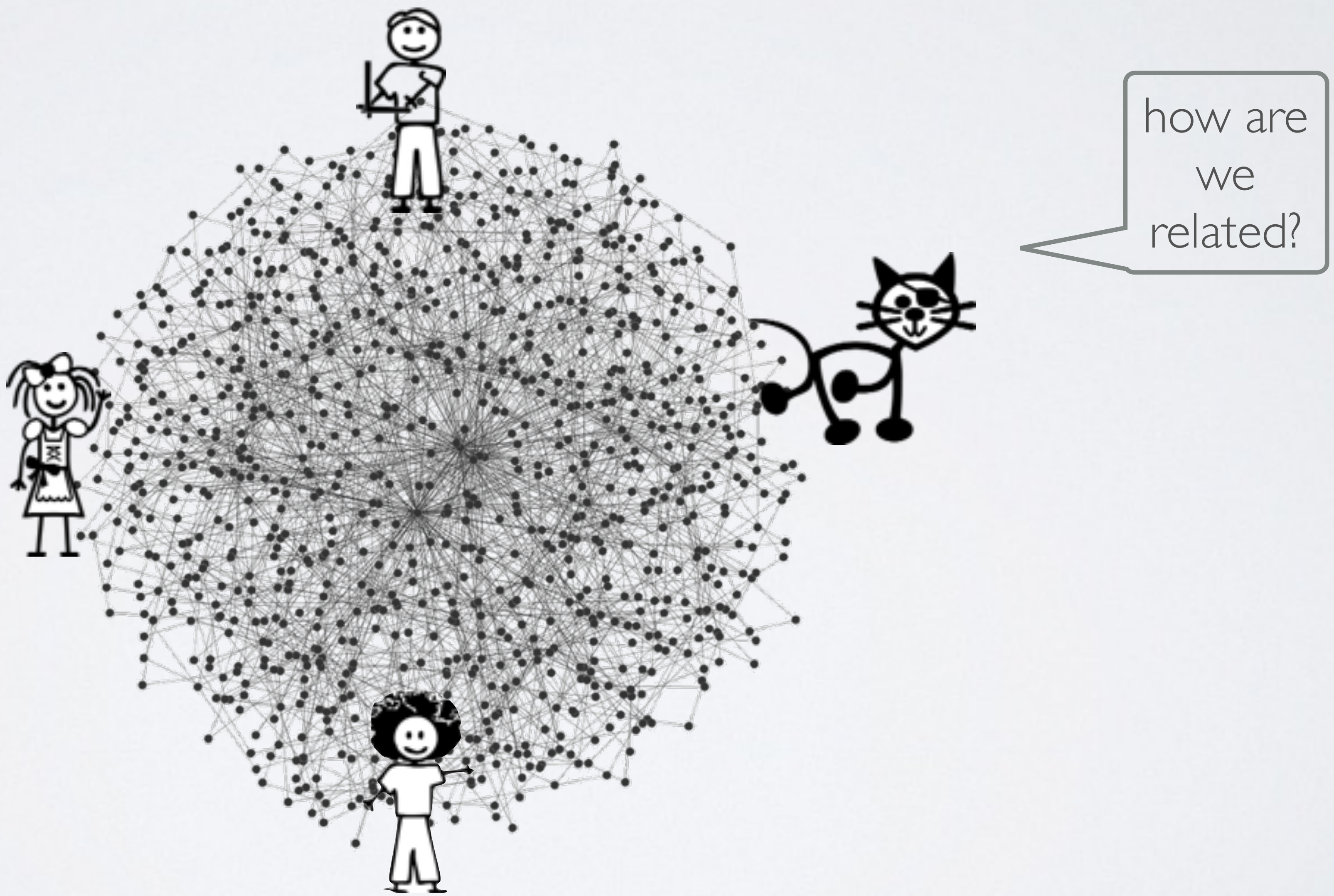
JAK2
BMP1
Cancer
Alzheimers
p53
HSP90
Cancer
Cancer
Alzheimers
SLC6a5
GSK3B
SNCA
PSEN1
Neurodegen.



JAK2
BMP1
Cancer
p53
HSP90
SLC6a5
GSK3B
SNCA
PSEN1
Alzheimers

What Was The Point?

Finding a **connector for a set of query nodes** in a graph is an interesting and relevant problem.

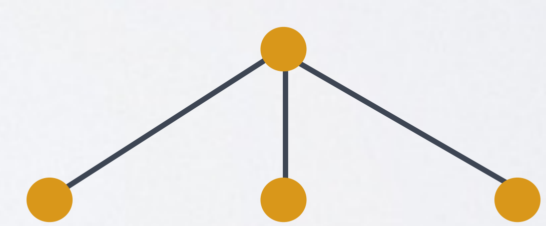


What Was The Point?

Finding a **connector for a set of query nodes** in a graph is an interesting and relevant problem.

The Wiener Index is the **sum of shortest-path distances**, which is intuitive graph measure of closeness.

high



low

What Was The Point?

Finding a **connector for a set of query nodes** in a graph is an interesting and relevant problem.

The Wiener Index is the **sum of shortest-path distances**, which is intuitive graph measure of closeness.

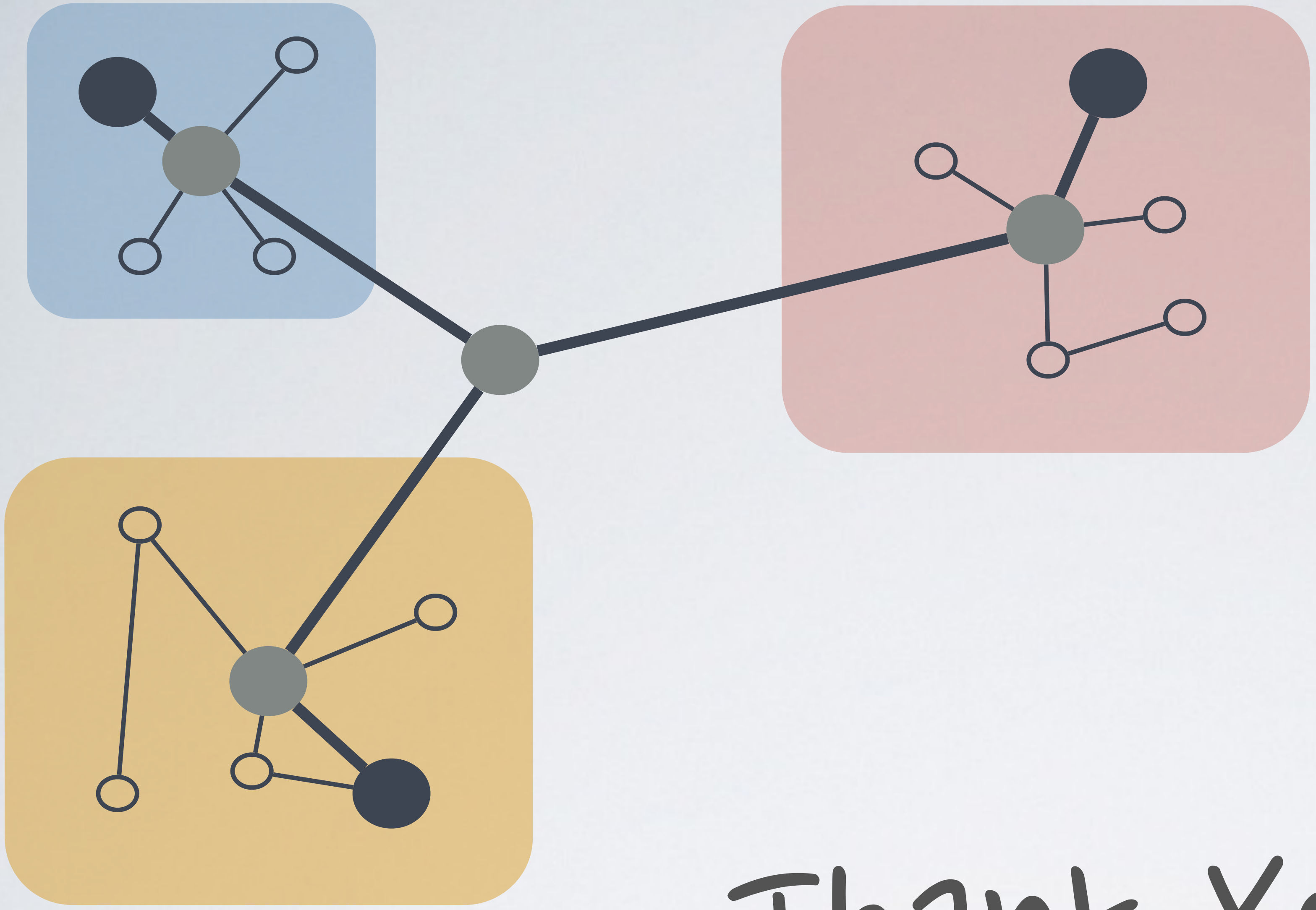
Proposed a constant factor approximation algorithm, that

- finds **small solutions**
- that are **easy to visualize**
- contain **important, central nodes**
- that **convey the relationship** among query nodes
- in a small amount of time.

Further Experiments

- Scalability
- Ground Truth Communities
- Steiner Tree Benchmark Datasets (DIMACS Challenge 2015)
- Comparison to Integer Program
- (and proofs)

Read the paper!



Thank You.

https://en.wikipedia.org/wiki/Wiener_Connector