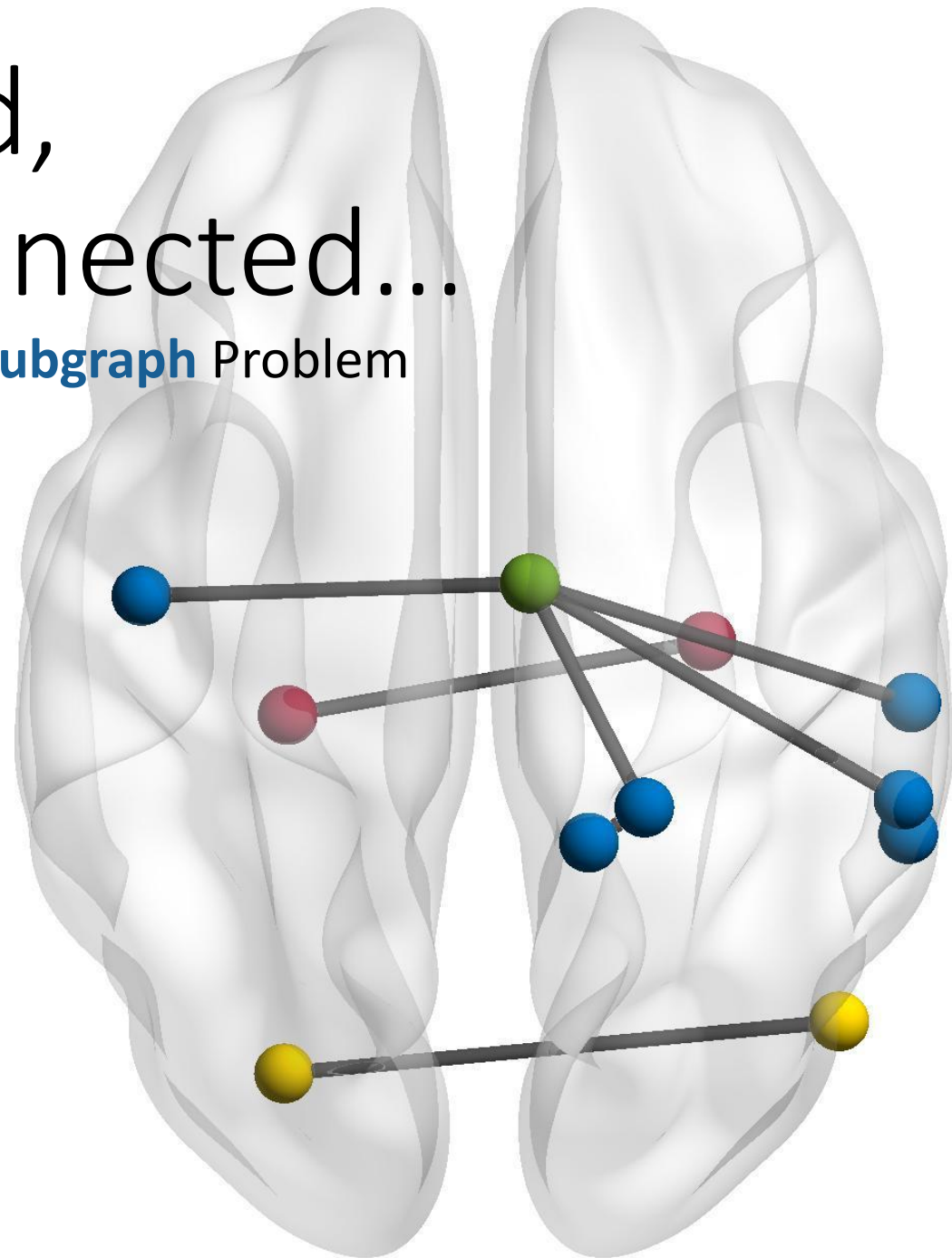


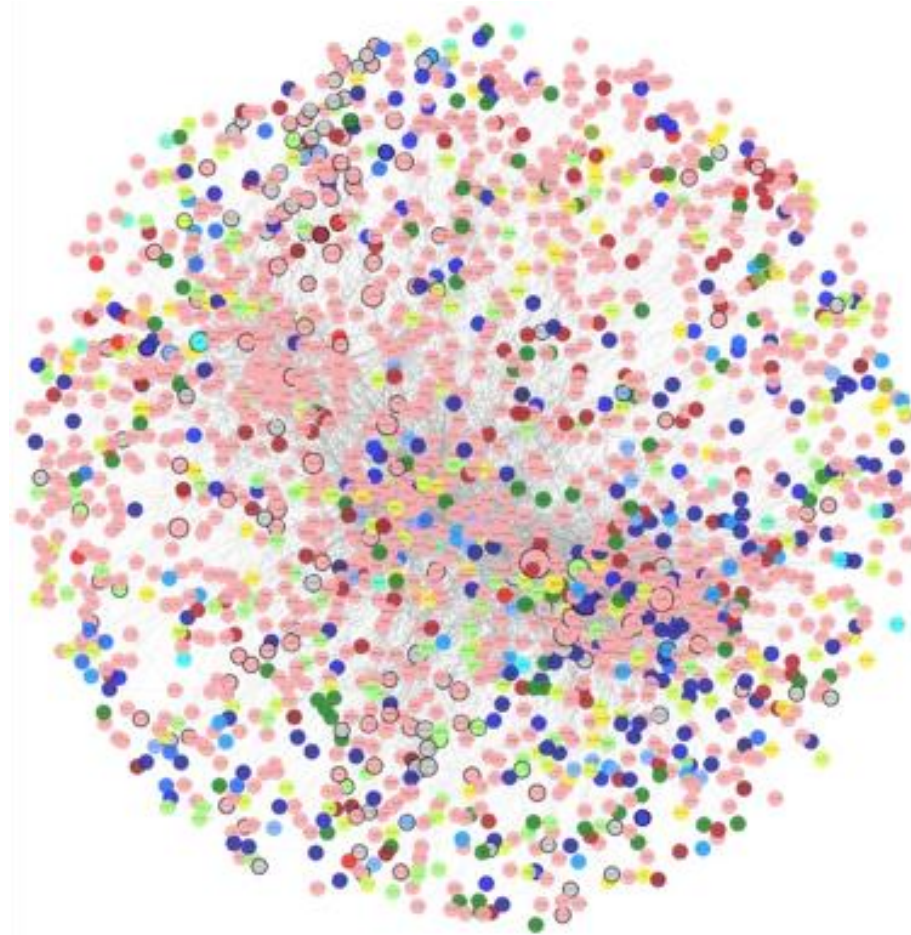
# To be connected, or not to be connected...

That is the **Minimum Inefficiency Subgraph** Problem

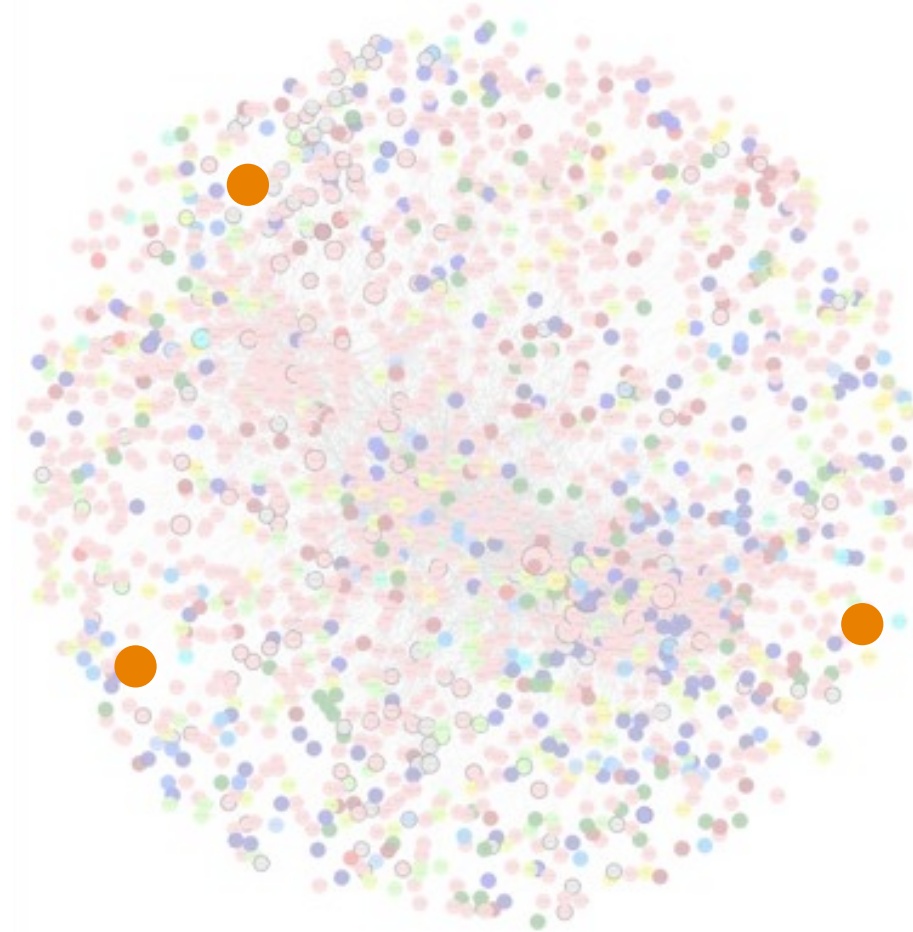
Natali Ruchansky  
Francesco Bonchi  
David Garcia-Soriano  
Francesco Gullo  
Nicolas Kourtellis



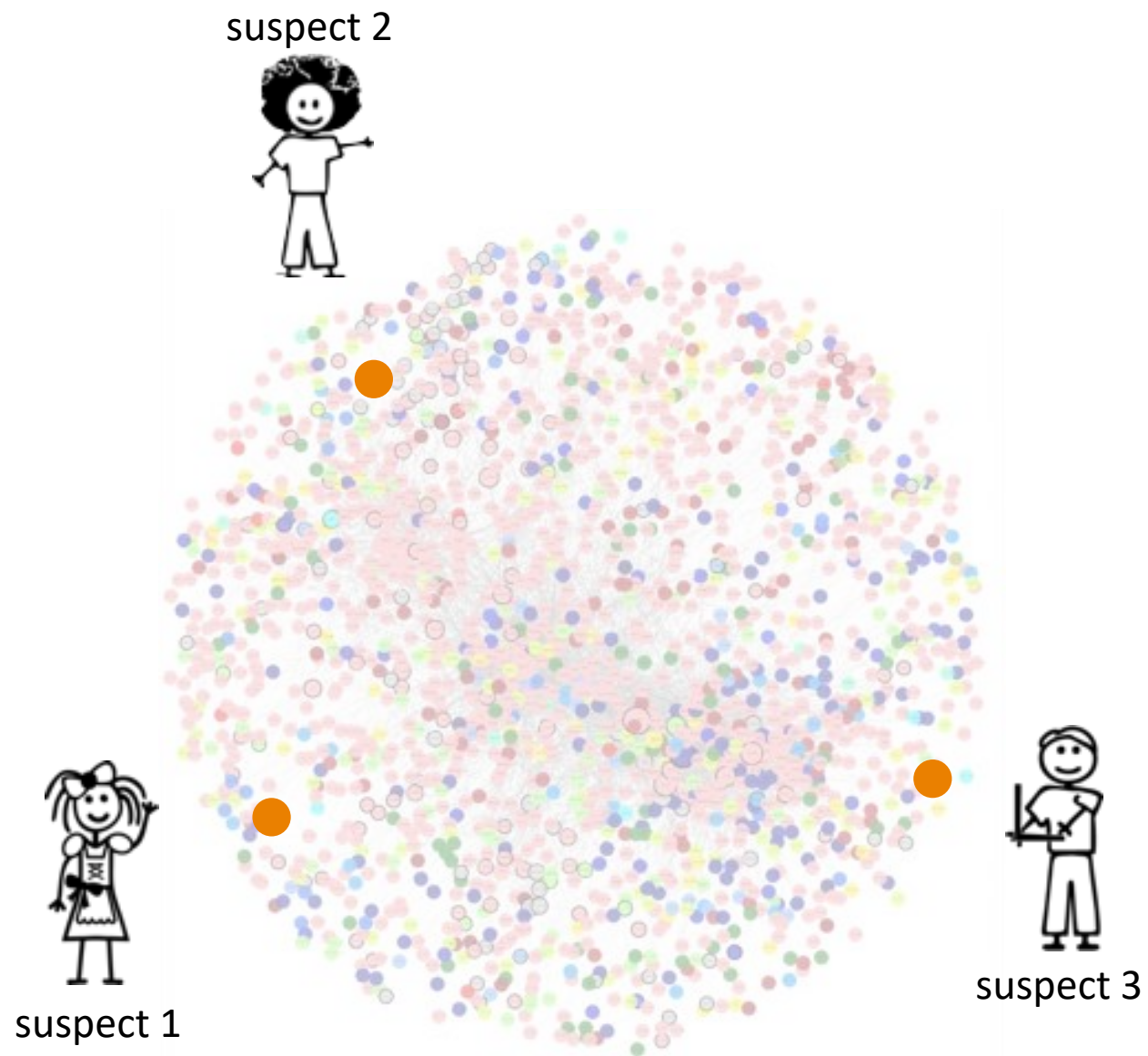
Biologists in Lab X have constructed a large protein-protein interaction network (PPI).



Biologists in Lab X have constructed a large protein-protein interaction network (PPI).



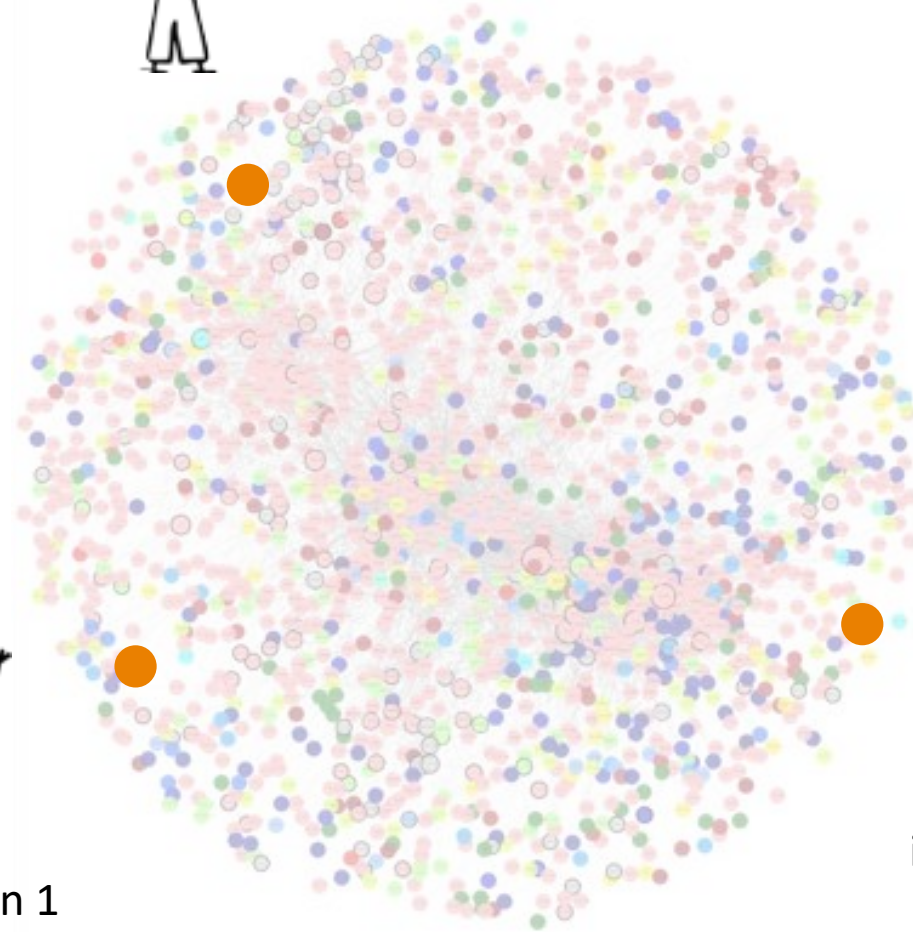
The PI has tasked them with making an amazing discovery about relationship among **specific proteins** P1, P2, and P3.



Given a set of **subjects in a terrorist network** suspected of organizing an attack. Which other subjects, likely to be involved, should we keep under control?



impression 2



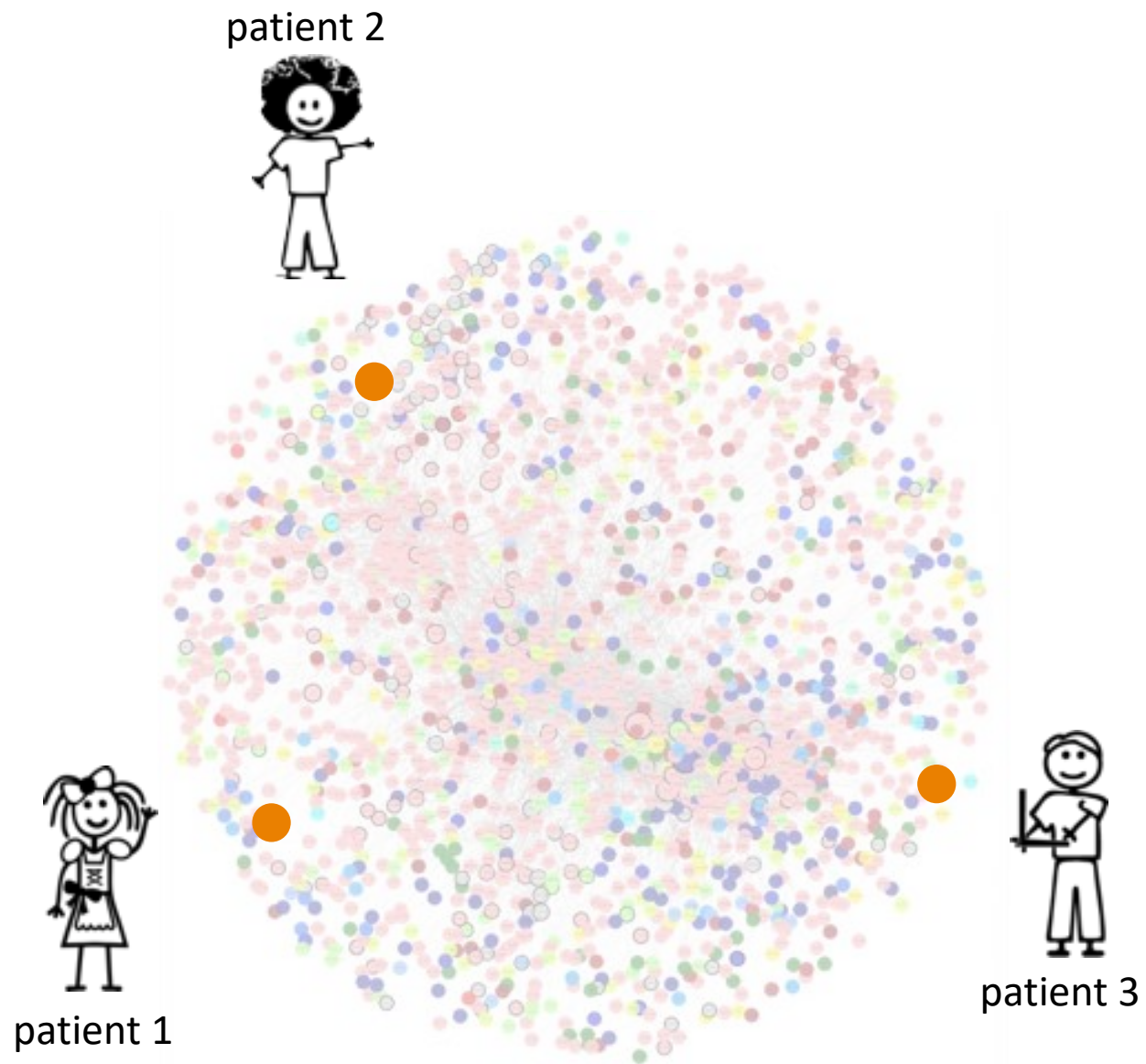
impression 1



impression 3



Given a set of **users who clicked on an ad**, who else should the ad be displayed to?



Given a set of **patients infected** with a viral disease, which other people should we monitor?

# Community search / seed set expansion

- General class of problems of the form:

Given a graph  $G=(V,E)$  and a set of vertices  $Q \subset V$ ,  
find a subgraph  $H$  of  $G$  that “explains” the connections among  $Q$ .  
( $H$  minimizes/maximizes some objective function)

- Several approaches in the literature
  - $H$  must be a **connected** subgraph
  - Mostly based on random-walks
  - Tend to return rather large solutions
  - Solutions get very large when query nodes belong to different communities
  - Have parameters

# The Minimum Wiener Connector Problem

(SIGMOD 2015)

Our proposal: find the **connected** subgraph  $H$  containing  $Q$  and minimizing the **Wiener Index** (the sum of pairwise distances)

$$H^* = \arg \min_{G[S]: Q \subseteq S \subseteq V} \sum_{\{u, v\} \in S} d_{G[S]}(u, v)$$

- Parameter-free
- Returns smaller and denser subgraphs
  - No matter whether the query nodes belong to the same community or not
- Add “**important**” nodes (high **centrality**)
- Efficient algorithm with approximation guarantees

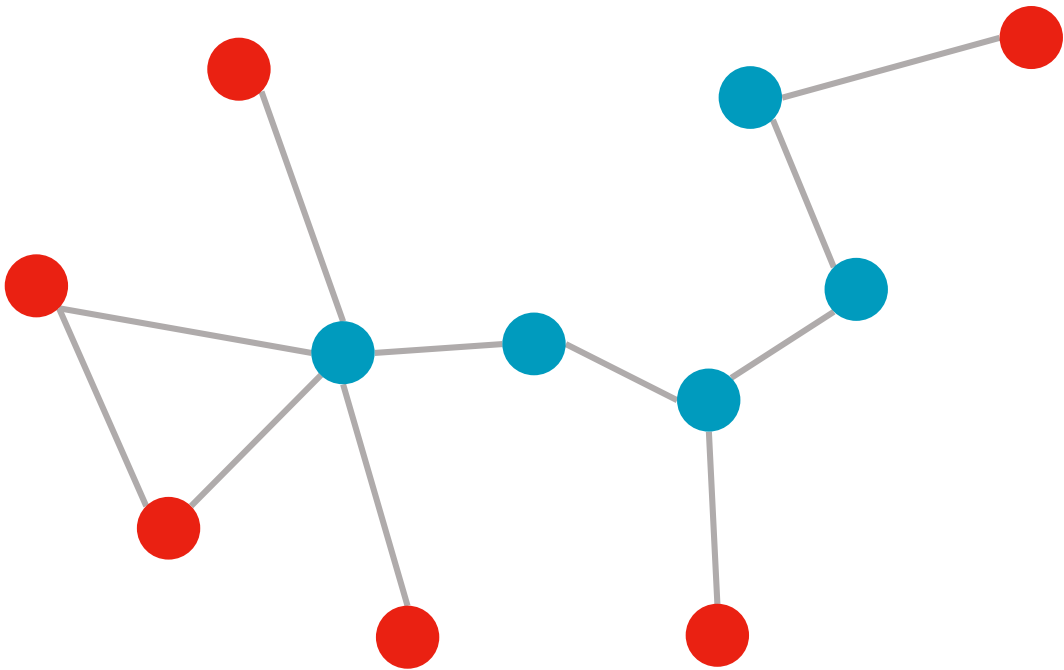


Smaller,  
denser, and  
more central  
vertices

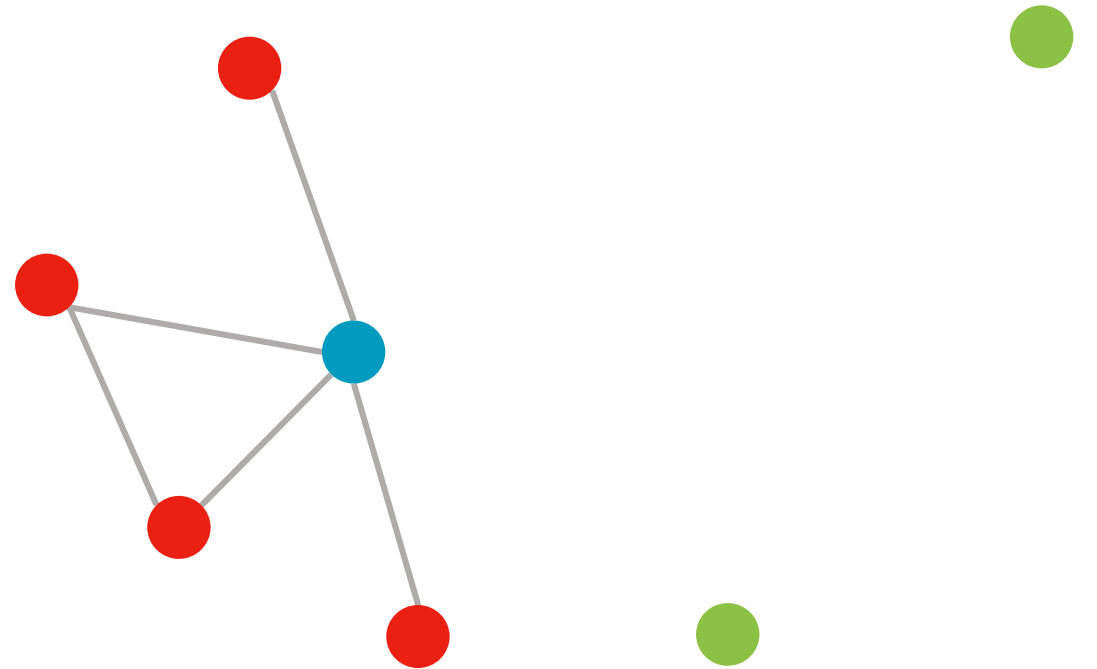
	email	yeast	oregon	astro	dblp	youtube	
$ V[H] $	671	819	9028	12758	11804	17865	CTP
	155	188	4556	1735	7349	5615	CPS
	137	100	1846	598	842	684	PPR
	26	<b>24</b>	26	26	25	19	ST
	<b>24</b>	<b>24</b>	<b>23</b>	<b>23</b>	<b>23</b>	<b>17</b>	WS-Q
$\delta(H)$	0.016	0.016	0.01	<0.01	<0.01	0.01	CTP
	0.047	0.028	0.02	0.019	0.01	<0.01	CPS
	0.029	0.039	0.02	0.07	0.01	0.02	PPR
	0.080	0.088	0.090	0.09	0.08	0.1	ST
	<b>0.093</b>	<b>0.091</b>	<b>0.106</b>	<b>0.13</b>	<b>0.11</b>	<b>0.13</b>	WS-Q
$b_c(H)$	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01	CTP
	0.03	0.02	<0.01	<0.01	<0.01	<0.01	CPS
	0.03	<0.01	<0.01	0.02	0.01	<0.01	PPR
	0.09	0.07	0.10	0.11	0.10	0.13	ST
	<b>0.11</b>	<b>0.11</b>	<b>0.12</b>	<b>0.14</b>	<b>0.12</b>	<b>0.18</b>	WS-Q
$W(H)$	$\approx 750k$	$\approx 2M$	$\approx 137M$	$\approx 292M$	$\approx 400M$	$\approx 1.5G$	CTP
	54 598	69 296	$\approx 50M$	$\approx 8.3M$	$\approx 12.6M$	$\approx 561M$	CPS
	52 222	15 838	$\approx 7.5M$	40 079	$\approx 1.2M$	$\approx 1.3M$	PPR
	1 200	1 259	1 164	1 318	3 371	1 324	ST
	<b>968</b>	<b>931</b>	<b>923</b>	<b>1 007</b>	<b>2 043</b>	<b>956</b>	WS-Q

# Relaxing connectivity

instead of forcing connectivity



relax the constraint



# Desired Properties

## Parsimonious vertex addition

- vertices should be added iff they help forming a more **cohesive** subgraph

## Outlier Tolerance

- query vertices which are far from others should remain disconnected

## Multi-community awareness

- if the query vertices span multiple communities, connectedness should not be imposed among them

# Cohesiveness

- As with the Wiener Connector, we leverage shortest path distances; however, the distance between disconnected vertices is infinite.
- Idea: **use the reciprocal of the shortest-path distance!** This has the useful property of handling disconnection neatly ( $\infty^{-1} = 0$ )

**Network Efficiency** (Latora and Marchiori): 
$$\mathcal{E}(G) = \frac{1}{|V|(|V| - 1)} \sum_{\substack{u, v \in V \\ u \neq v}} \frac{1}{d_G(u, v)}$$

**Harmonic Centrality** (Boldi and Vigna): 
$$c(u) = \sum_{v \in V} \frac{1}{d_G(v, u)}$$



# What about these problem statements?

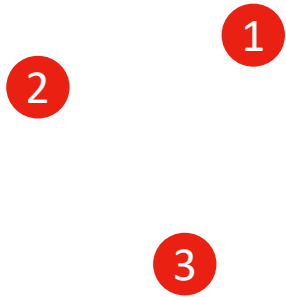
---

Given a graph  $G=(V,E)$  and a set of vertices  $Q \subset V$ , find a (not-necessarily connected) subgraph  $H$  of  $G$ , with  $Q \subset V(H)$  that maximizes network efficiency  $E(H)$

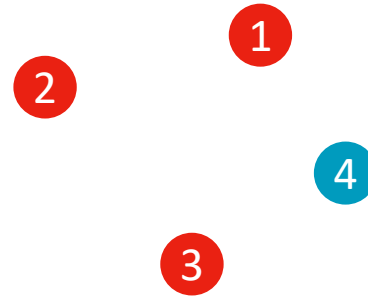
---

Given a graph  $G=(V,E)$  and a set of vertices  $Q \subset V$ , find a (not-necessarily connected) subgraph  $H$  of  $G$ , with  $Q \subset V(H)$  that maximizes the total harmonic centrality  $C(H)$

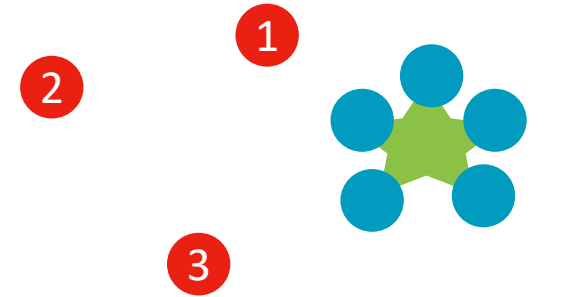
# These do not work...



$C(G[Q])=0$   
 $E(G[Q])=0$



$C(H)=0$   
 $E(H)=0$



$C(H)=9900$   
 $E(H)=0.942$

# Minimize Network Inefficiency

Given a graph  $G=(V,E)$ , we define its inefficiency as:

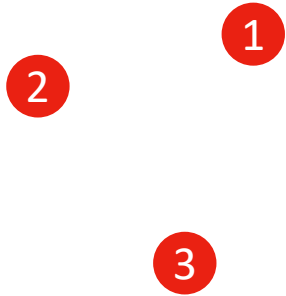
$$\mathcal{I}(G) = \sum_{\substack{u, v \in V \\ u \neq v}} 1 - \frac{1}{d_G(v, u)}$$

Note:

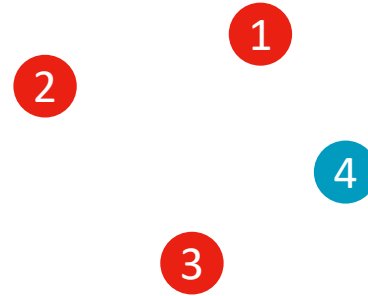
$$\mathcal{E}(G) = C(G)/(n(n-1))$$

$$\mathcal{I}(G) = n(n-1) - C(G)$$

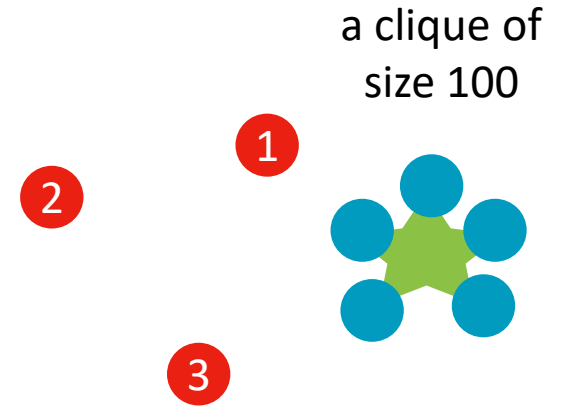
... and this works



$C(G[Q])=0$   
 $E(G[Q])=0$   
 $I(G[Q])=6$



$C(G[Q])=0$   
 $E(G[Q])=0$   
 $I(G[Q])=12$



$C(G[Q])=9900$   
 $E(G[Q])=0.942$   
 $I(G[Q])=606$



# Problem statement and hardness

PROBLEM 1 (MIN-INEFFICIENCY-SUBGRAPH). *Given an undirected graph  $G = (V, E)$  and a query set  $Q \subseteq V$ , find*

$$H^* = \arg \min_{G[S]: Q \subseteq S \subseteq V} I(G[S]).$$

THEOREM 4.1. *MIN-INEFFICIENCY-SUBGRAPH is NP-hard, and it remains hard even on undirected graphs with diameter 3.*

# Greedy Algorithm

Connect

Start with the **Minimum Wiener Connector** for  $Q$

Remove

Remove one vertex at a time until  $Q$  is disconnected

Choose

Choose the intermediate solution  $S$  that minimizes  $I(S)$

# Competitors

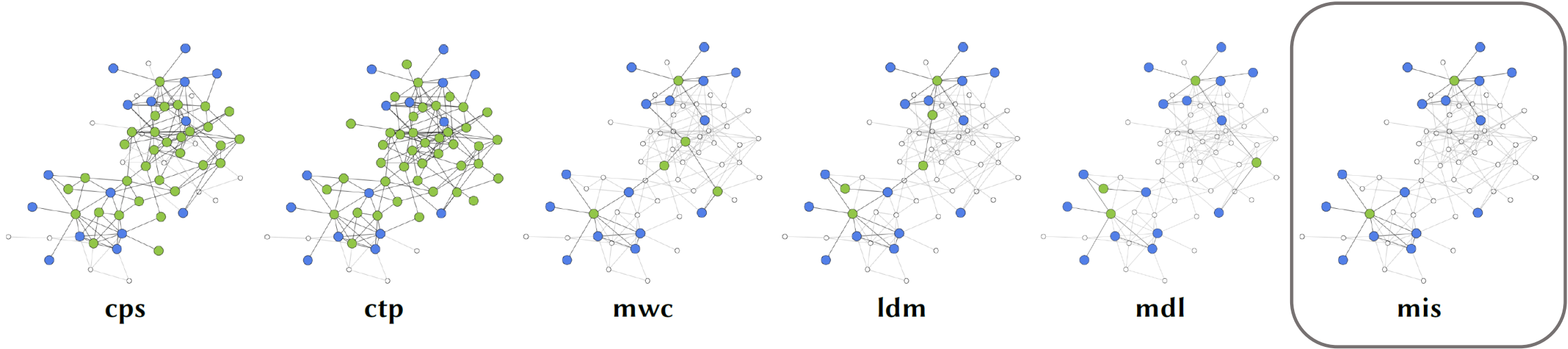


Figure 1: Comparison on the Dolphins social network: **query vertices are in blue**, **added vertices are in green**.

Research Track Paper

### Center-Piece Subgraphs: Problem Definition and Fast Solutions

Hanghang Tong  
Carnegie Mellon University  
htong@cs.cmu.edu

Christos Faloutsos  
Carnegie Mellon University  
christos@cs.cmu.edu

**ABSTRACT**  
Given  $Q$  nodes in a social network (say, authorship network), how can we find the node/author that is the center-piece, and has direct or indirect connections to all, or most of them? For example, this node could be the common author, or someone who started the research area that the  $Q$  nodes belong to. Isomorphic scenarios appear in law enforcement (find the mastermind criminal, connected to all current suspects), gene regulatory networks (find the protein that participates in pathways with all or most of the given  $Q$  proteins), viral marketing and many more.

**Categories and Subject Descriptors**  
H.2.8 (Database Management): Database Applications - Data Mining

**General Terms**  
Application, experimentation

**Keywords**  
Center-piece subgraph, goodness score, KAPLAN

KDD'06

### The Community-search Problem and How to Plan a Successful Cocktail Party

Mauro Sozio<sup>\*</sup>  
Max-Planck-Institut für Informatik  
Saarbrücken, Germany  
msozio@mpi-inf.mpg.de

Aristides Gionis  
Yahoo! Research  
Barcelona, Spain  
gionis@yahoo-inc.com

**ABSTRACT**  
A lot of research in graph mining has been devoted in the discovery of communities. Most of the work has focused in the scenario where communities need to be discovered with only reference to the input graph. However, for many interesting applications one is interested in finding the community formed by a given set of nodes. In this paper we study a query-dependent variant of the community-detection problem, which we call the *community-search problem*: given a graph  $G$ , and a set of *query nodes* in the graph, we seek to find a subgraph of  $G$  that contains the query nodes and it is densely connected.

**1. INTRODUCTION**  
Graphs is one of most ubiquitous data representations, and they find applications in a wide range of areas including biology, physics, social sciences, and information technology. With the increasing availability of very large networks, there is need for designing algorithmic data-analysis tools and for developing applications that exploit the latent structure in the data.

Discovering communities in graphs and social networks has drawn a large amount of attention in recent years [9, 13, 15, 16, 29]. It has been one of them most well-studied problems of graph mining. Most of the work has focused in the scenario where communities need to be discovered in

KDD'10

### The Minimum Wiener Connector Problem

Natali Ruchansky  
Computer Science Dept.  
Boston University, USA  
natali@bu.edu

Francesco Bonchi  
Francesco Gullo  
Yahoo Labs, Barcelona  
(bonchi.davidg.gullo.kourtell}@yahoo-inc.com

David Garcia-Soriano  
Nicolas Kourtellis  
Yahoo Labs, Barcelona

**ABSTRACT**  
The Wiener index of a graph is the sum of all pairwise shortest-path distances between its vertices. In this paper we study the novel problem of finding a minimum Wiener connector: given a connected graph  $G = (V, E)$  and a set  $Q \subseteq V$  of query vertices, find a subgraph of  $G$  that connects all query vertices and has minimum Wiener index.

We show that MIN WIENER CONNECTOR admits a polynomial-time (and hence impractical) exact algorithm for the special case where the number of query vertices is bounded. We show that in general the problem is NP-hard, and has no

SIGMOD'15

### Bump hunting in the dark: Local discrepancy maximization on graphs

Aristides Gionis, Michael Mathioudakis, Antti Ukkonen  
Helsinki Institute for Information Technology HIIT  
Aalto University, Finland  
firstname.lastname@aalto.fi

**Abstract**—We study the problem of discrepancy maximization on graphs given a set of nodes  $Q$  of an underlying graph  $G$ . We aim to identify a connected subgraph of  $G$  that contains many more nodes from  $Q$  than other nodes. This variant of the discrepancy-maximization problem extends the well-known notion of “bump hunting” in the Euclidean space.

We consider the problem under two access models. In the *unrestricted-access model*, the whole graph  $G$  is given as input, while in the *local-access model* we can only retrieve the neighbors of a given node in  $G$  using a possibly slow and costly interface.

We prove that the basic problem of discrepancy maximization on graphs is NP-hard, and empirically evaluate the performance of four heuristics for solving it. For the local-access model we consider three different algorithms that aim to recover a part of  $G$  large enough to contain an optimal solution, while using only a small number of calls to the neighbor-function interface. We perform a thorough experimental evaluation in order to understand the trade offs between the proposed methods and

ICDE'15

### Mining Connection Pathways for Marked Nodes in Large Graphs

Leman Akoglu  
SUNY at Stony Brook  
leman@cs.stonybrook.edu

Jilles Vreeken  
University of Antwerp  
jilles.vreeken@ua.ac.be

Hanghang Tong  
City College, City University of NY  
tong@cs.cuny.cuny.edu

Duen Hong Chau  
Georgia Tech  
polo@gatech.edu

Nikolaj Tatti  
KU Leuven  
nikolaj.tatti@cs.kuleuven.be

Christos Faloutsos  
Carnegie Mellon University  
christos@cs.cmu.edu

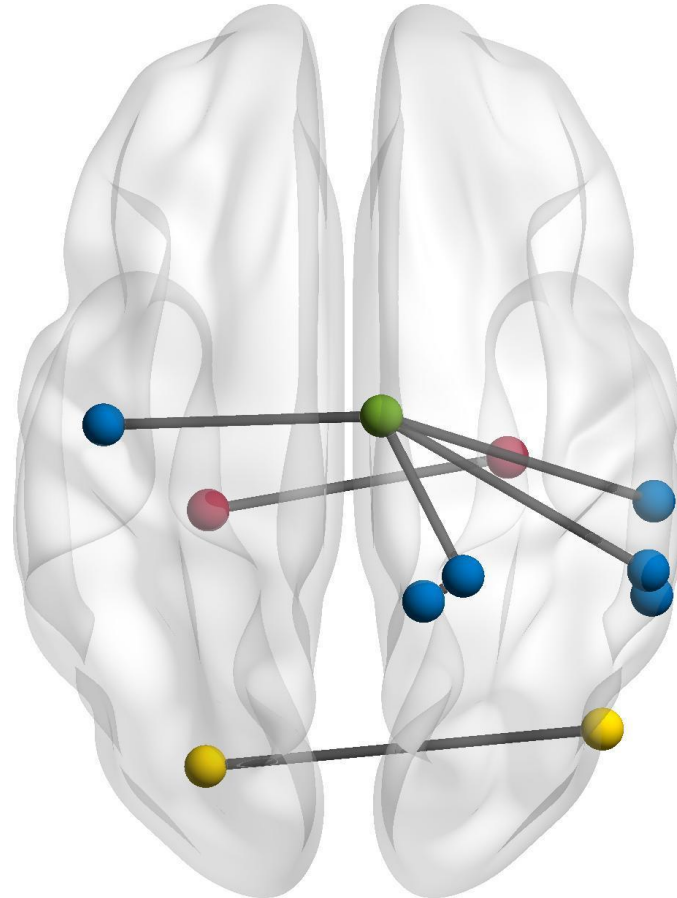
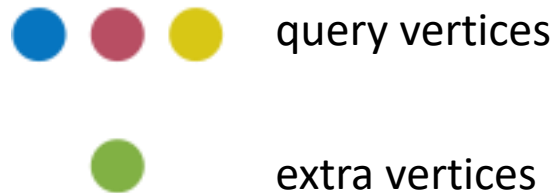
**Abstract**  
Suppose we are given a large graph in which, by some external process, a handful of nodes are marked. What can we say about these nodes? Are they close together in the graph? or, if segregated, how many groups do they form? We approach this problem by trying to find sets of simple connection pathways between sets of marked nodes.

We formalize the problem in terms of the Minimum Description Length principle: a pathway is simple when we need only few bits to talk about it, so, following each other

SDM'13

# Brain Co-activation Network

The data is a graph where each vertex is an area of the brain and edges are added according to co-activation in experiments. (The graph is one connected component)

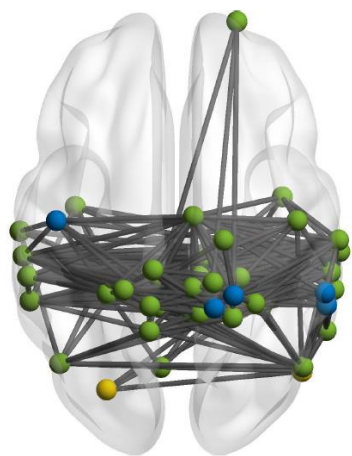


The 3 components in the solution end up corresponding to different functions: **motor**, **visual**, and **emotional**.

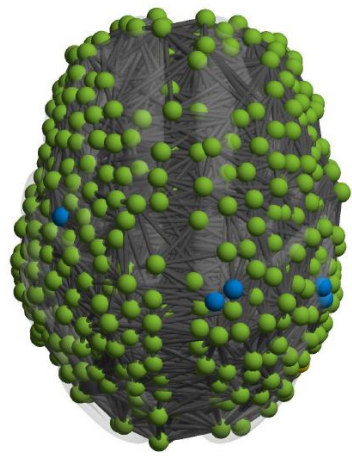
relaxing connectivity highlights three different functional relationships and gives a smaller, more interpretable solution



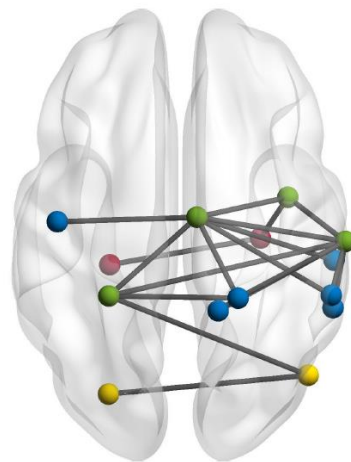
# Brain Co-activation Network: competitors



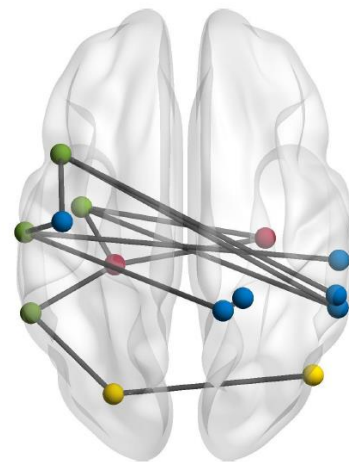
**cps**



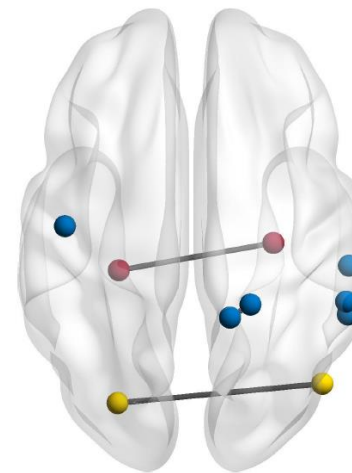
**ctp**



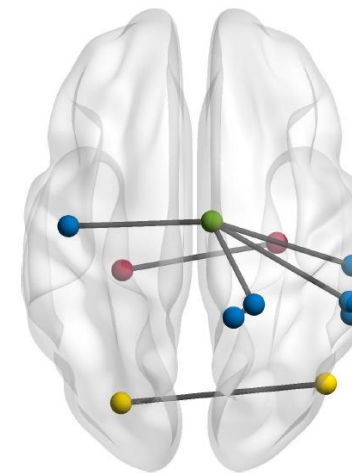
**mwc**



**ldm**



**mdl**



**mis**

# Experimental Results

## Parsimonious vertex addition

- vertices should be added iff they help forming a more **cohesive** subgraph

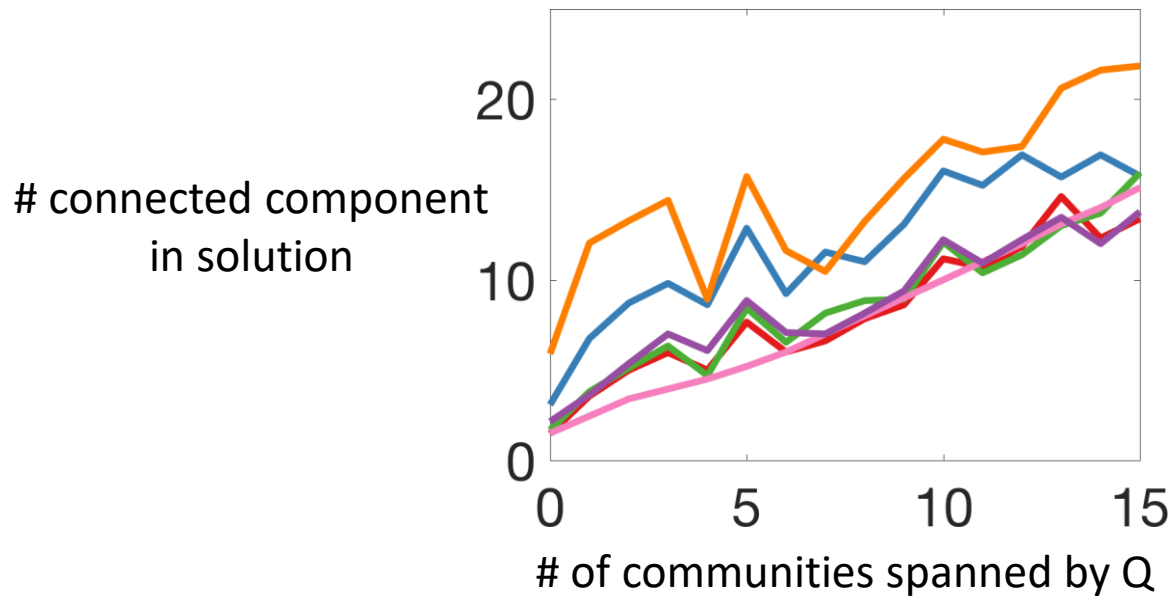
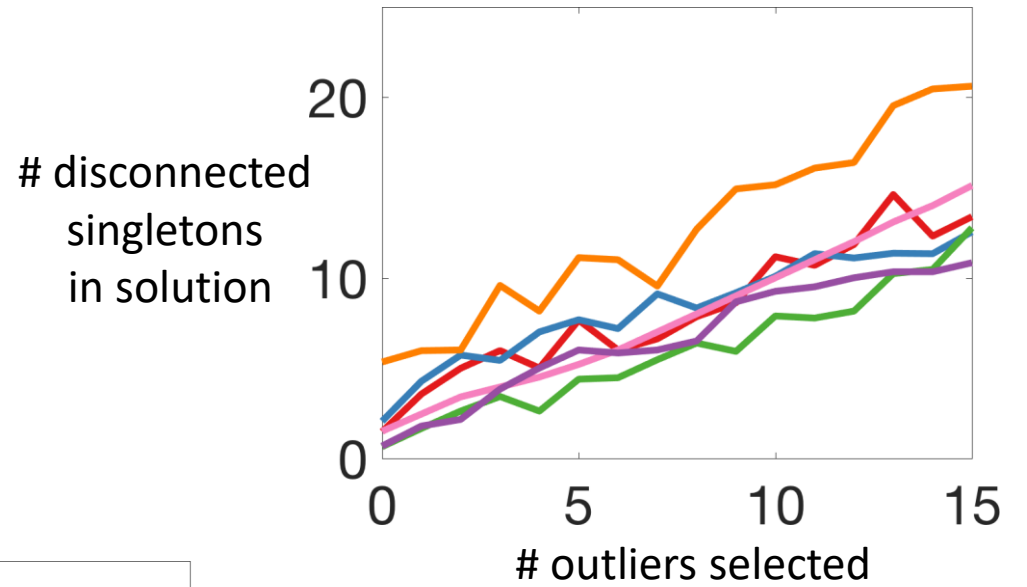
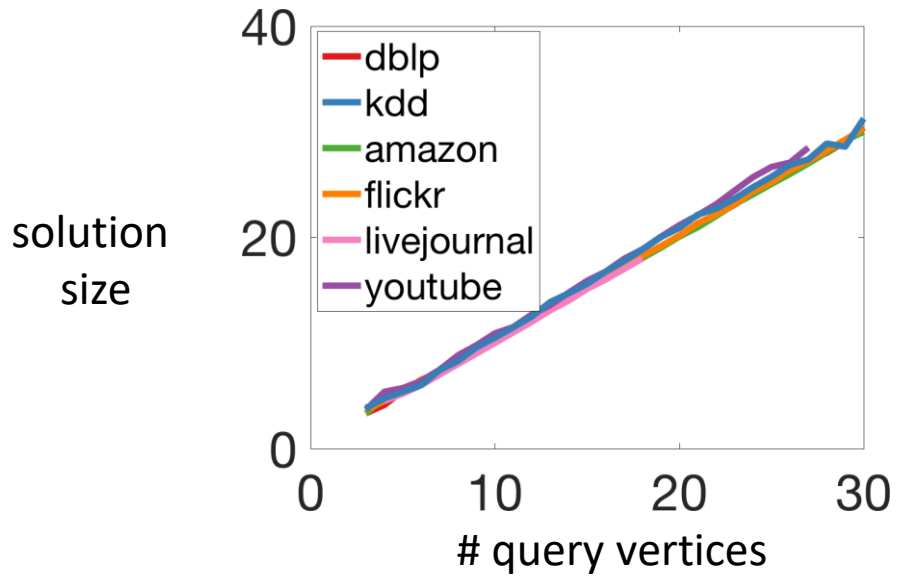
## Outlier Tolerance

- query vertices which are far from others should remain disconnected

## Multi-community awareness

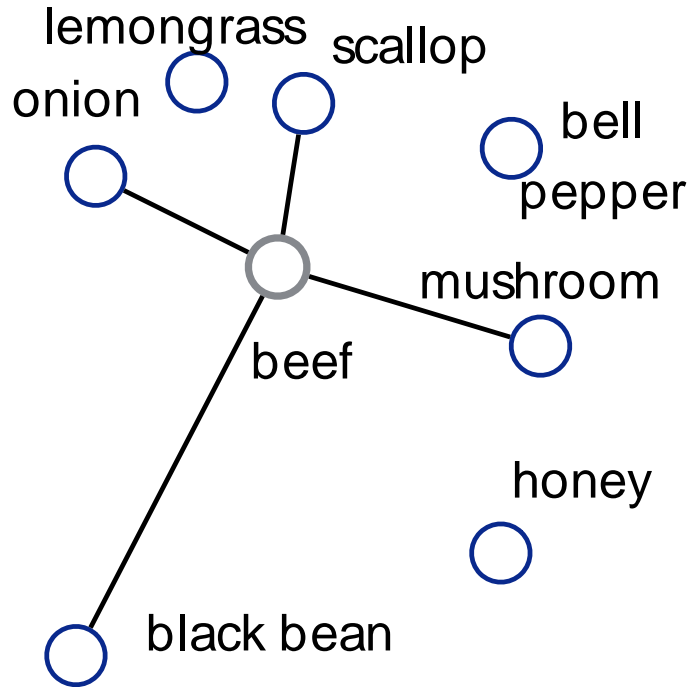
- if the query vertices span multiple communities, connectedness should not be imposed among them

# Experimental Results

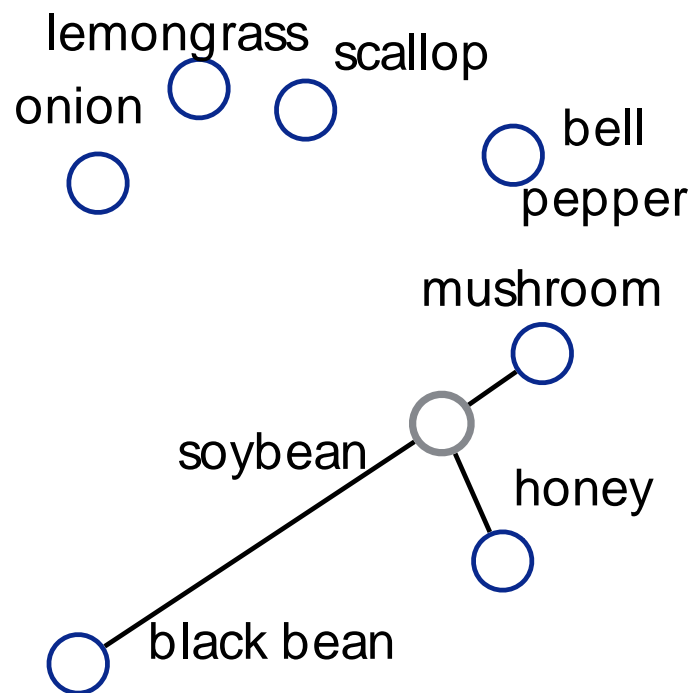


# Cohesive meal creation

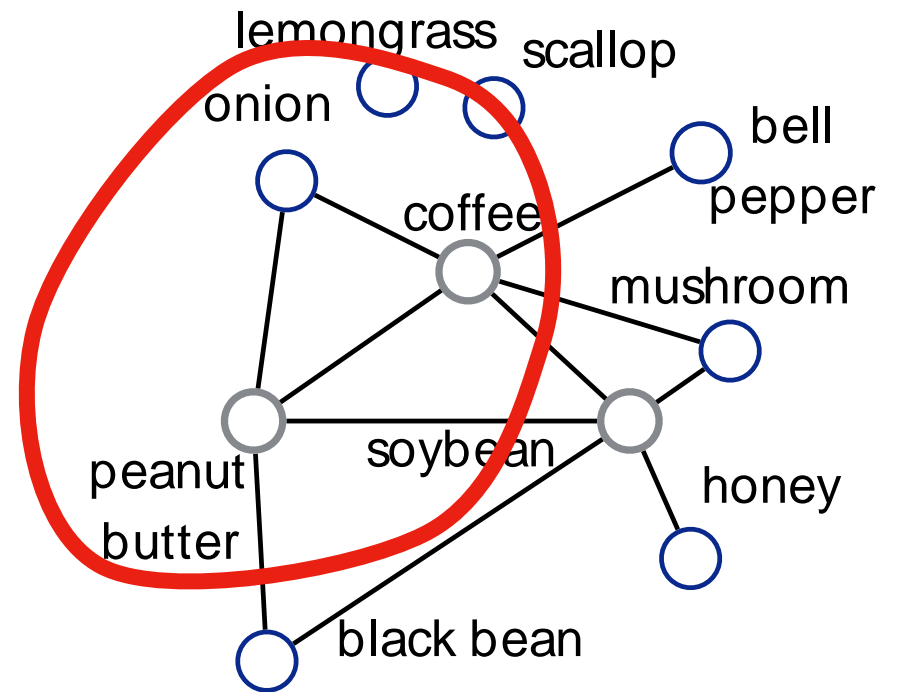
Minimum Inefficiency



MDL-based

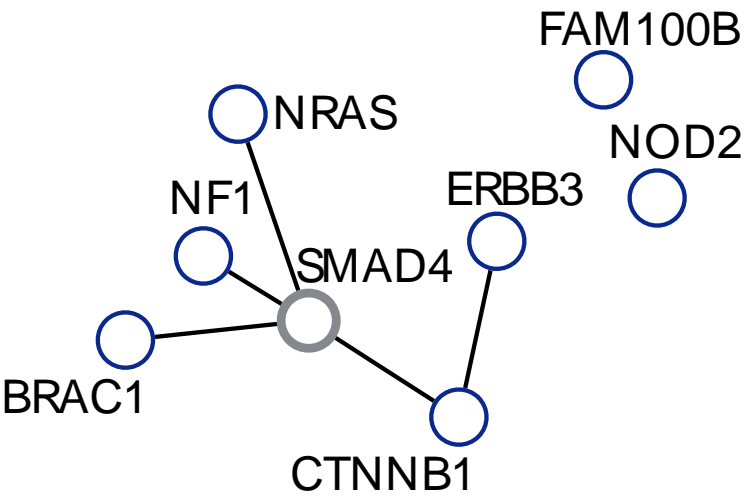


Bump Hunting

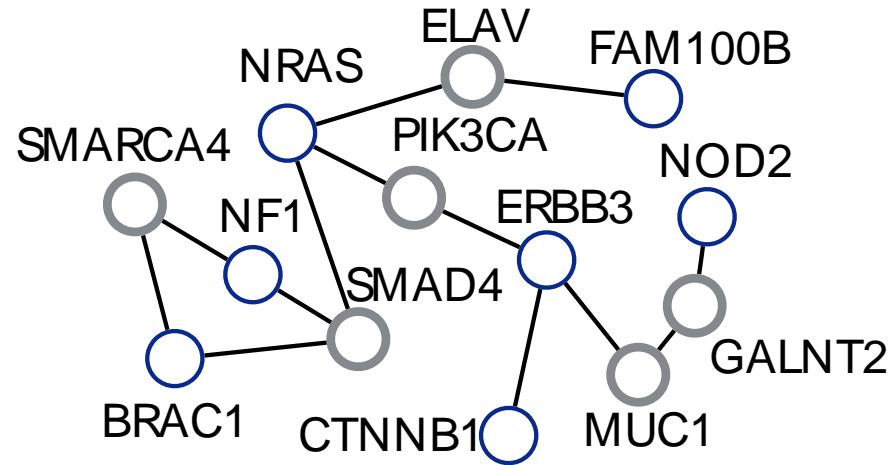


# Biology

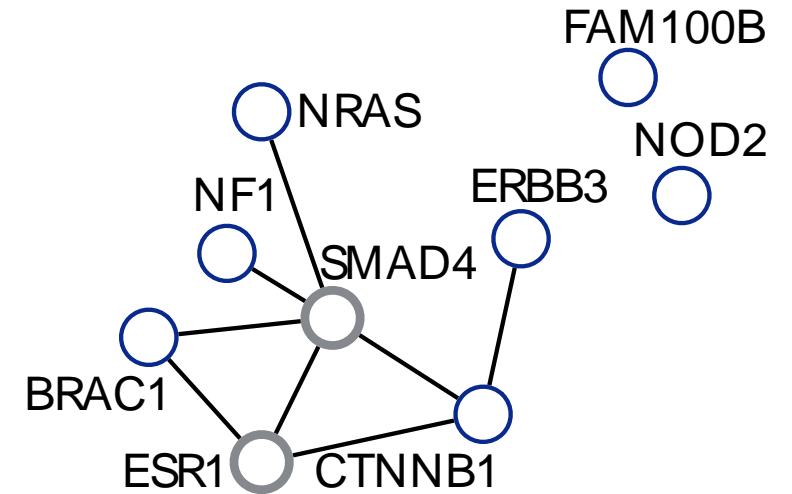
Minimum Inefficiency



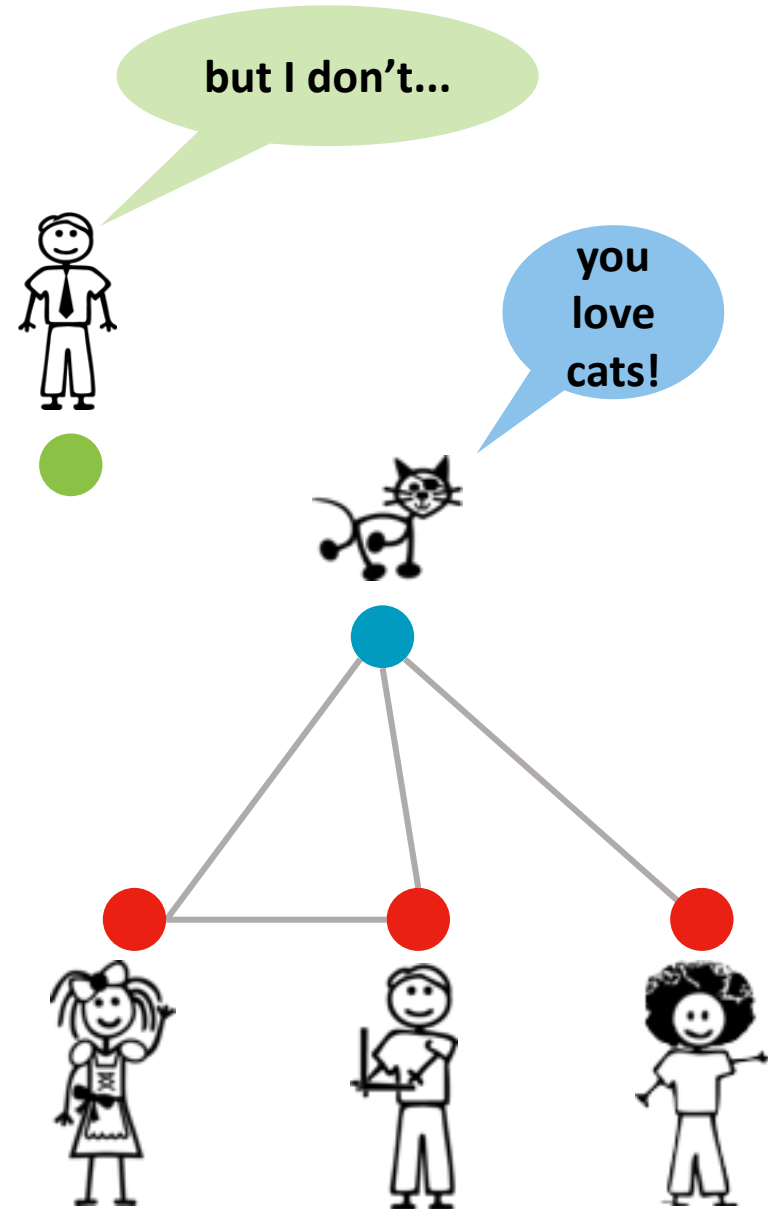
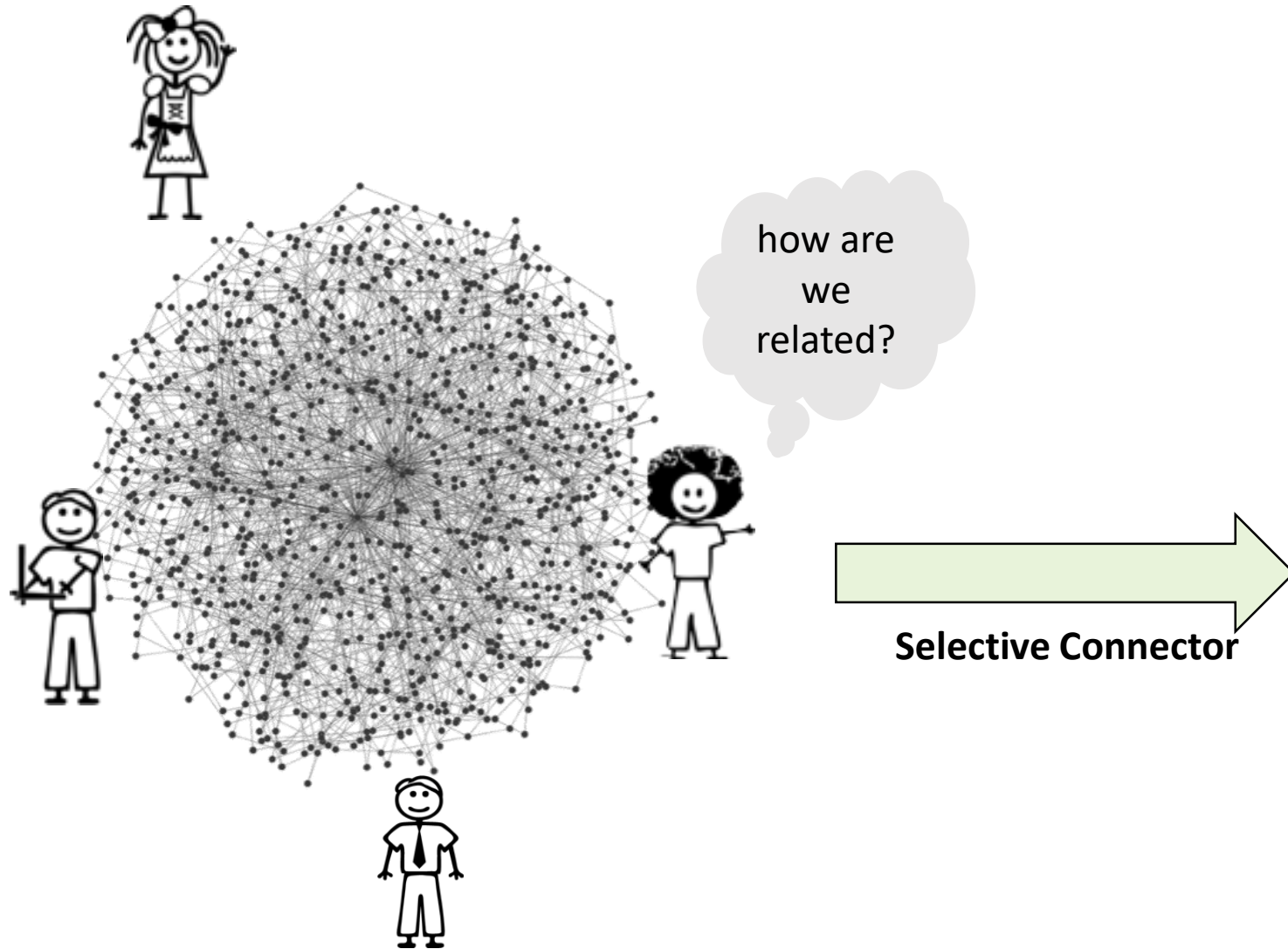
MDL-based



Bump Hunting



# Takeaway



# Thanks!



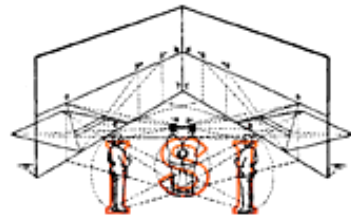
@FrancescoBonchi



[www.francescobonchi.com](http://www.francescobonchi.com)



[francesco.bonchi@isi.it](mailto:francesco.bonchi@isi.it)



INSTITUTE  
FOR SCIENTIFIC INTERCHANGE  
FOUNDATION