Discovering Hidden Structures in Complex Networks

Roman Vershynin

University of Michigan

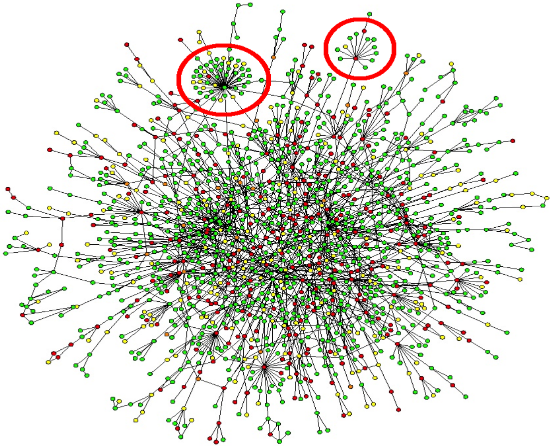
SPARS 2015 University of Cambridge March 2015

Network Science is highly interdisciplinary.

+ finance + technology + . . .

Many networks have fascinating structure.

Some structures are apparent, local.

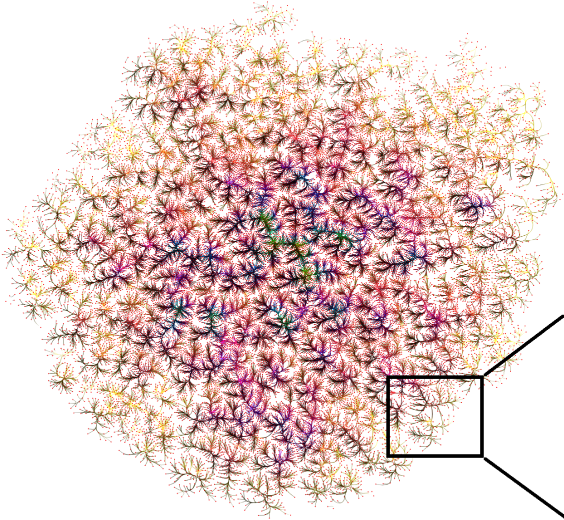


Protein interaction network

[A.-L. Barabási & Z. Oltvai, Nature Reviews Genetics 5, 101–113, Feb. 2004]

Many networks have fascinating structure.

Some structures are apparent, local.

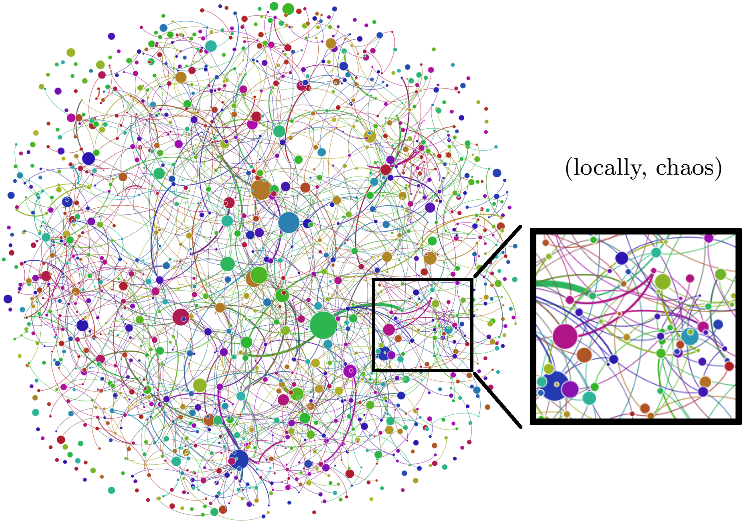


The Internet

(C. Hurter et al., Eurographics Conference on Visualization 2012)

Many networks have fascinating structure.

Other structures are latent, global...



(locally, chaos)

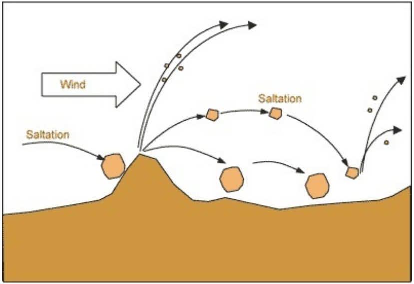
Collaboration network of economists

(AER, JPE, Econometrica, RES, QJE. www.cloudycnet.net)

Many networks have fascinating structure.

... just like in nature:

global structure



local chaos

Basic Questions

- How can we find latent structures in real networks?
- How can we explain and model these structures?

Mathematical perspective

Model large networks as random graphs. (Edges drawn at random.)

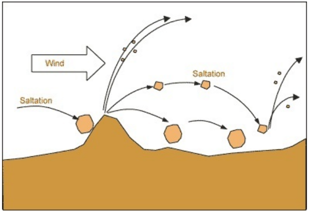
A leap of faith.

Mathematical perspective

Model large networks as random graphs. (Edges drawn at random.)

A leap of faith.

Similar to **statistical physics**: model *complex* systems as *random* ones.
Randomness at the microscopic level averages out at the macroscopic level:

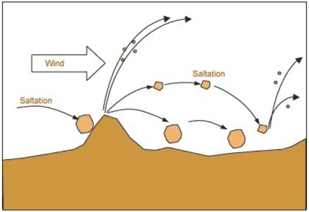


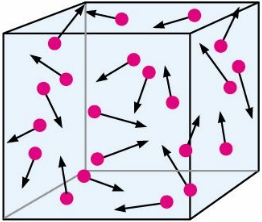
Mathematical perspective

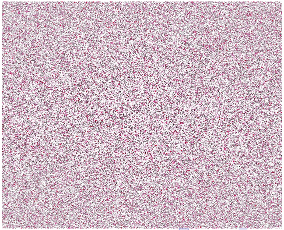
Model large networks as random graphs. (Edges drawn at random.)

A leap of faith.

Similar to **statistical physics**: model *complex* systems as *random* ones.
Randomness at the microscopic level averages out at the macroscopic level:







Random graphs: Erdös-Rényi model $G(n, p)$

Edges drawn independently at random, with probability $p \in [0, 1]$.

Random graphs: Erdös-Rényi model $G(n,p)$

Edges drawn independently at random, with probability $p \in [0,1]$.

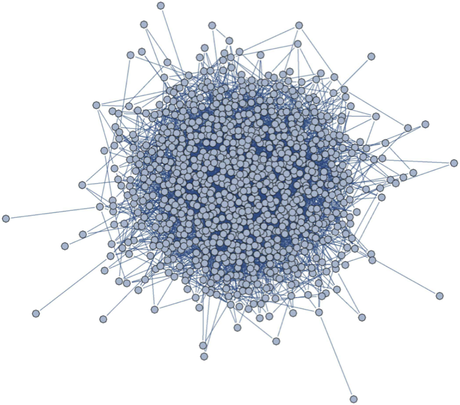
[Paul Erdös, Alfred Rényi 59]: the birth of random graph theory.

Random graphs: Erdös-Rényi model $G(n, p)$

Edges drawn independently at random, with probability $p \in [0, 1]$.

[Paul Erdös, Alfred Rényi 59]: the birth of random graph theory.

$G(n, p)$ with $n = 1000$, $p = 0.00095$



(A. Novozhilov's course in Mathematics of Networks, NDSU)

Inhomogeneous Erdös-Rényi model $G(n, (p_{ij}))$

Edges are still independent, but can have **different** probabilities p_{ij}.

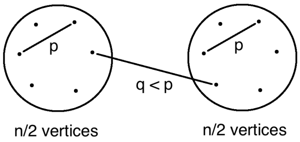
Allows to model networks with structure = communities (clusters).

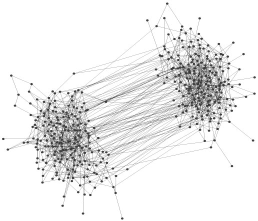
Inhomogeneous Erdös-Rényi model $G(n, (p_{ij}))$

Edges are still independent, but can have **different** probabilities p_{ij}.

Allows to model networks with structure = communities (clusters).

Example. **Stochastic block model** with two communities $G(n, p, q)$:
Edges *within* each community: probability p; *across* communities: probability $q < p$.

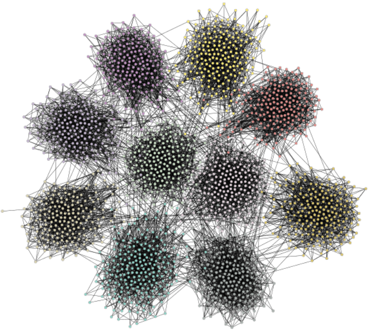




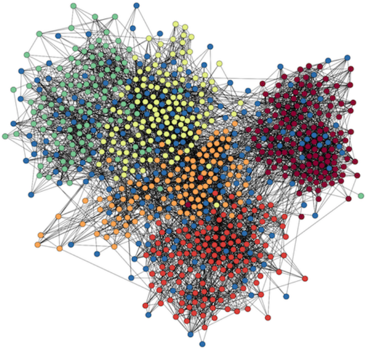
Inhomogeneous Erdös-Rényi model $G(n,(p_{ij}))$

Multiple communities are possible to model, too:

Stochastic block model



Real data (aggression network of students)



(UC Davis Center for Visualization)

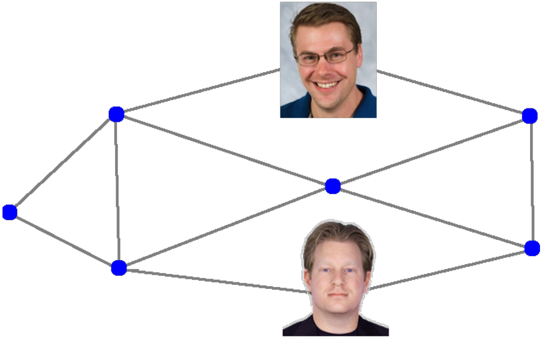
Network Model Recovery

Model Recovery Problem. Observe one instance of a network from $G(n, (p_{ij}))$. Recover the model, i.e. the connection probabilities p_{ij}.

Application to real graphs:

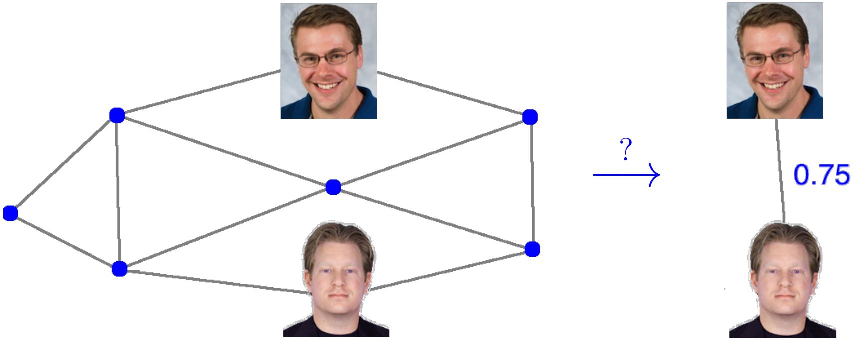
Model Recovery Problem. Observe one instance of a network from $G(n, (p_{ij}))$. Recover the model, i.e. the connection probabilities p_{ij}.

Application to real graphs:



Model Recovery Problem. Observe one instance of a network from $G(n, (p_{ij}))$. Recover the model, i.e. the connection probabilities p_{ij}.

Application to real graphs:



p_{ij} = *"latent bonds"* between vertices.

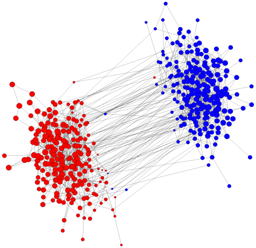
Link prediction.

Network Model Recovery Problem

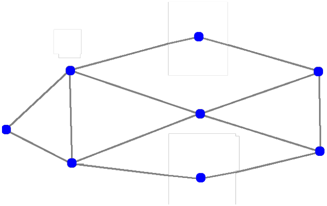
A particular case, for stochastic block models:

Community Detection Problem. Observe a network drawn from the stochastic block model $G(n, p, q)$. Recover the two communities.

$\xrightarrow{?}$

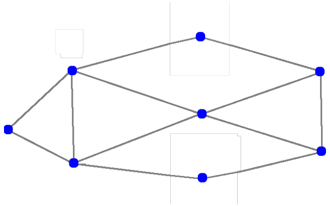


Adjacency matrix A:



$$\longrightarrow \qquad A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Adjacency matrix A:



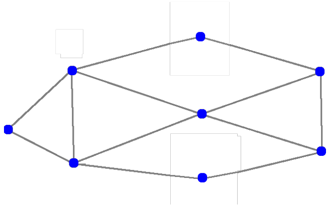
$$\longrightarrow \qquad A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

For inhomogeneous Erdös-Rényi model:

$$A = (\mathrm{Bernoulli}(p_{ij})) \qquad \mathbb{E}\, A = (p_{ij})$$

From graphs to matrices

Adjacency matrix A:



$$\longrightarrow \qquad A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

For inhomogeneous Erdös-Rényi model:

$$A = (\text{Bernoulli}(p_{ij})) \qquad \mathbb{E}\, A = (p_{ij})$$

Model Recovery Problem. Observe A; recover $\mathbb{E}\, A$.

Relation to matrix completion

Evident but not thoroughly explored.

Matrix completion: recover a low-rank matrix from a few randomly chosen entries.

$$\begin{bmatrix} & & & & .1 & & & \\ .7 & & .6 & & & & & .1 \\ & & & .9 & & & & \\ & & .9 & & & .1 & & \\ .1 & & & & & & & .5 \\ & .1 & & & .8 & & & \\ .3 & & & & & & & \\ & & & & & .6 & & \end{bmatrix} \xrightarrow{?} \begin{bmatrix} 1 & .7 & .6 & .7 & .1 & .4 & .3 & .2 \\ .7 & 1 & .6 & .5 & .2 & .1 & .2 & .1 \\ .6 & .6 & 1 & .9 & .4 & .2 & .3 & .3 \\ .7 & .5 & .9 & 1 & .2 & .1 & .3 & .2 \\ .1 & .2 & .4 & .2 & 1 & .8 & .6 & .5 \\ .4 & .1 & .2 & .1 & .8 & 1 & .7 & .6 \\ .3 & .2 & .3 & .3 & .6 & .7 & 1 & .9 \\ .2 & .1 & .3 & .2 & .5 & .6 & .9 & 1 \end{bmatrix}$$

Relation to matrix completion

Evident but not thoroughly explored.

Matrix completion: recover a low-rank matrix from a few randomly chosen entries.

$$\begin{bmatrix} & & & & .1 & & & \\ .7 & & .6 & & & & & .1 \\ & & & .9 & & & & \\ & & .9 & & & .1 & & \\ .1 & & & & & & & .5 \\ & .1 & & & .8 & & & \\ .3 & & & & & & & \\ & & & & & .6 & & \end{bmatrix} \xrightarrow{?} \begin{bmatrix} 1 & .7 & .6 & .7 & .1 & .4 & .3 & .2 \\ .7 & 1 & .6 & .5 & .2 & .1 & .2 & .1 \\ .6 & .6 & 1 & .9 & .4 & .2 & .3 & .3 \\ .7 & .5 & .9 & 1 & .2 & .1 & .3 & .2 \\ .1 & .2 & .4 & .2 & 1 & .8 & .6 & .5 \\ .4 & .1 & .2 & .1 & .8 & 1 & .7 & .6 \\ .3 & .2 & .3 & .3 & .6 & .7 & 1 & .9 \\ .2 & .1 & .3 & .2 & .5 & .6 & .9 & 1 \end{bmatrix}$$

Network model recovery: recover a (low-rank?) matrix $\mathbb{E}\,A = (p_{ij})$ from random measurements $A = (\text{Bernoulli}(p_{ij}))$.

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \xrightarrow{?} \begin{bmatrix} 1 & .7 & .6 & .7 & .1 & .4 & .3 & .2 \\ .7 & 1 & .6 & .5 & .2 & .1 & .2 & .1 \\ .6 & .6 & 1 & .9 & .4 & .2 & .3 & .3 \\ .7 & .5 & .9 & 1 & .2 & .1 & .3 & .2 \\ .1 & .2 & .4 & .2 & 1 & .8 & .6 & .5 \\ .4 & .1 & .2 & .1 & .8 & 1 & .7 & .6 \\ .3 & .2 & .3 & .3 & .6 & .7 & 1 & .9 \\ .2 & .1 & .3 & .2 & .5 & .6 & .9 & 1 \end{bmatrix}$$

Most relevant comparison is to **single-bit matrix completion** [Davenport et al '12].

Existing approaches

Mostly apply to **stochastic block models**.

Insights from Combinatorics, Computer Science, Statistics, Physics:

- combinatorial techniques (min-cut, hierarchical clustering)
- spectral methods – this talk
- statistical inference (likelihood maximization)
- variational methods
- Markov chain Monte Carlo
- belief propagation
- convex optimization
- semidefinite programming – this talk
- ...

Spectral methods

Based on two observations:

(a) eigenstructure(A) $\approx$ eigenstructure($\mathbb{E}\,A$);

Spectral methods

Based on two observations:

(a) eigenstructure$(A) \approx$ eigenstructure$(\mathbb{E}\, A)$;

(b) eigenstructure$(\mathbb{E}\, A)$ reveals **the latent structure** of the network.

Spectral methods

Based on two observations:

(a) eigenstructure(A) $\approx$ eigenstructure($\mathbb{E}\, A$);

(b) eigenstructure($\mathbb{E}\, A$) reveals **the latent structure** of the network.

More on (b) later. By Davis-Kahan theorem, (a) would follow if

$$A \approx \mathbb{E}\, A \text{ in the operator norm.}$$

Spectral methods

Based on two observations:

(a) eigenstructure(A) $\approx$ eigenstructure($\mathbb{E}\, A$);

(b) eigenstructure($\mathbb{E}\, A$) reveals **the latent structure** of the network.

More on (b) later. By Davis-Kahan theorem, (a) would follow if

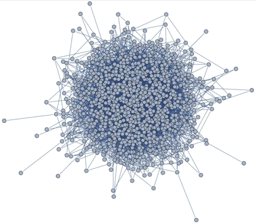
$A \approx \mathbb{E}\, A$ in the operator norm.

Is this true? In other words:

Question. Do random graphs concentrate near their "expected" graphs?

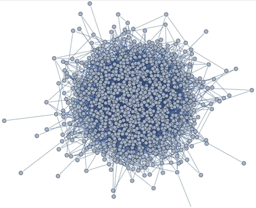
Dense random graphs concentrate

Consider an inhomogeneous Erdös-Rényi random graph $G(n,(p_{ij}))$ with *expected degrees* $np_{ij} \sim d$.



Dense random graphs concentrate

Consider an inhomogeneous Erdös-Rényi random graph $G(n,(p_{ij}))$ with *expected degrees* $np_{ij} \sim d$.

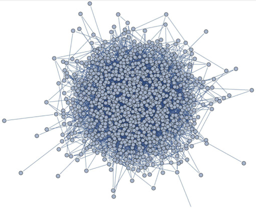


Theorem. *A random graph with expected degrees* $d \gtrsim \log n$ *concentrates:*

$$\|A - \mathbb{E}\, A\| \lesssim \sqrt{d} \quad \textit{w.h.p. while} \quad \|\mathbb{E}\, A\| \sim d.$$

Dense random graphs concentrate

Consider an inhomogeneous Erdös-Rényi random graph $G(n,(p_{ij}))$ with *expected degrees* $np_{ij} \sim d$.



Theorem. *A random graph with expected degrees* $d \gtrsim \log n$ *concentrates:*

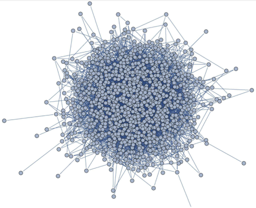
$$\|A - \mathbb{E}\, A\| \lesssim \sqrt{d} \quad \textit{w.h.p. while} \quad \|\mathbb{E}\, A\| \sim d.$$

Proofs:

- [Kahn-Szemeredi 89] → [Feige-Ofek 05, Lei-Rinaldo 13, Chin-Rao-Vu 15]: Simple concentration of $x^\mathsf{T}(A - \mathbb{E}\, A)y$ for *fixed* x, y; then complicated union bound over x, y (tailored the coefficient profiles of x, y).

Dense random graphs concentrate

Consider an inhomogeneous Erdös-Rényi random graph $G(n, (p_{ij}))$ with *expected degrees* $np_{ij} \sim d$.



Theorem. *A random graph with expected degrees $d \gtrsim \log n$ concentrates:*

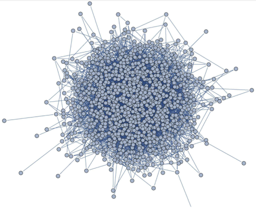
$$\|A - \mathbb{E}\, A\| \lesssim \sqrt{d} \quad w.h.p. \;\; while \quad \|\mathbb{E}\, A\| \sim d.$$

Proofs:

- [Kahn-Szemeredi 89] → [Feige-Ofek 05, Lei-Rinaldo 13, Chin-Rao-Vu 15]: Simple concentration of $x^\mathsf{T}(A - \mathbb{E}\, A)y$ for *fixed* x, y; then complicated union bound over x, y (tailored the coefficient profiles of x, y).
- Other approaches: [Hajek-Wu-Xu 14; Bandeira-van Handel 14; Le-Vershynin 15].

Dense random graphs concentrate

Consider an inhomogeneous Erdös-Rényi random graph $G(n,(p_{ij}))$ with *expected degrees* $np_{ij} \sim d$.



Theorem. *A random graph with expected degrees* $d \gtrsim \log n$ *concentrates:*

$$\|A - \mathbb{E}\,A\| \lesssim \sqrt{d} \quad \textit{w.h.p. while} \quad \|\mathbb{E}\,A\| \sim d.$$

Proofs:

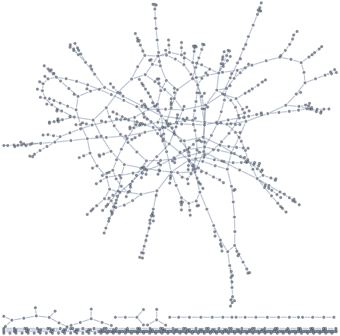
- [Kahn-Szemeredi 89] $\to$ [Feige-Ofek 05, Lei-Rinaldo 13, Chin-Rao-Vu 15]: Simple concentration of $x^\mathsf{T}(A - \mathbb{E}\,A)y$ for *fixed* x, y; then complicated union bound over x, y (tailored the coefficient profiles of x, y).
- Other approaches: [Hajek-Wu-Xu 14; Bandeira-van Handel 14; Le-Vershynin 15].
- Weaker results: [Furedi-Komlos 80] with $d \gtrsim \log^4 n$; [Oliveira 10] with $\|A - \mathbb{E}\,A\| \lesssim \sqrt{d \log n}$ by matrix Bernstein inequality.

Sparse random graphs do not concentrate

Observation. *A random graph* $G(n,p)$ *with expected degrees* $d = np \ll \log n$ *does* *not* *concentrate:*

$$\|A - \mathbb{E}\, A\| \gg \|\mathbb{E}\, A\|.$$

See [Krivelevich-Sudakov 03].

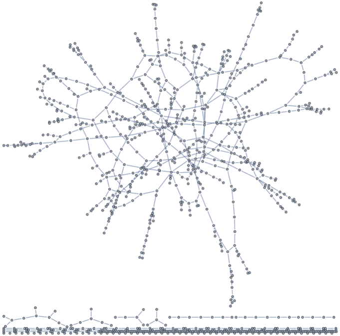


Sparse random graphs do not concentrate

Observation. *A random graph $G(n,p)$ with expected degrees $d = np \ll \log n$ does not concentrate:*

$$\|A - \mathbb{E}\, A\| \gg \|\mathbb{E}\, A\|.$$

See [Krivelevich-Sudakov 03].

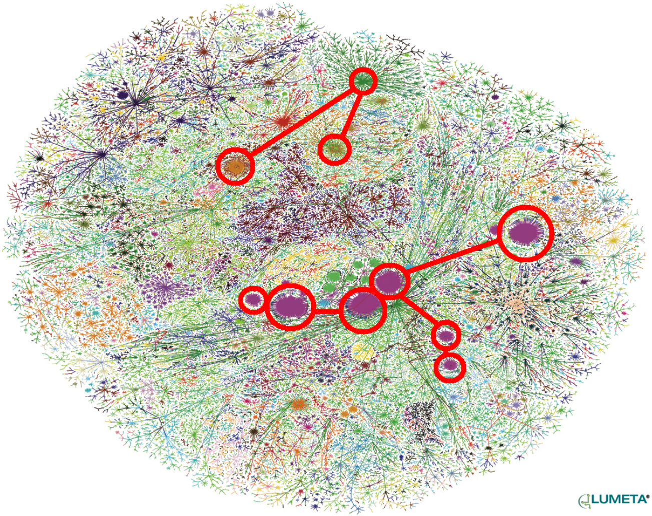


What is wrong with sparse graphs?

The degrees are wild, do not concentrate near d anymore.
High-degree vertices blow up $\|A\|$: some columns of A are too large.

Sparse random graphs do not concentrate

High-degree vertices dominate the picture. Spectral methods reveal only those vertices. *Local information*, no latent structure [Mihail-Papadimitriou 02].



The Internet

Regularization approach

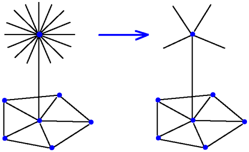
Preprocess the network.

Regularize the high-degree vertices: reweight (or remove) enough edges from them.

Regularization approach

Preprocess the network.

Regularize the high-degree vertices: reweight (or remove) enough edges from them.

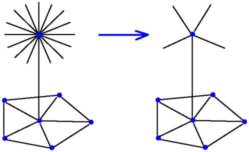


Does this restore concentration?

Regularization approach

Preprocess the network.

Regularize the high-degree vertices: reweight (or remove) enough edges from them.



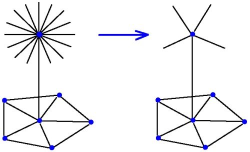
Does this restore concentration?

This is a non-trivial question. (Are these vertices *the only* troublemakers?)

Regularization approach

Preprocess the network.

Regularize the high-degree vertices: reweight (or remove) enough edges from them.



Does this restore concentration?

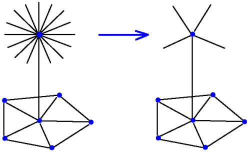
This is a non-trivial question. (Are these vertices *the only* troublemakers?)

- Yes, if we remove **all** high-degree vertices and all their edges [Feige-Ofek 05]. But these vertices hold the network together (hubs)! Their removal can cause network to fall apart.

Regularization approach

Preprocess the network.

Regularize the high-degree vertices: reweight (or remove) enough edges from them.



Does this restore concentration?

This is a non-trivial question. (Are these vertices *the only* troublemakers?)

- Yes, if we remove **all** high-degree vertices and all their edges [Feige-Ofek 05]. But these vertices hold the network together (hubs)! Their removal can cause network to fall apart.
- Yes, in full generality. Any type of regularization helps, as long as it brings down the degrees to $\sim d$. [Le-Levina-V, Le-V 05].

Regularization and concentration: theory

Inhomogeneous E-R random graph with $d = \max np_{ij}$.

Regularize vertices with degrees $> 2d$:
make all degrees $\leq 2d$ by reducing the weights of edges arbitrarily.

Regularization and concentration: theory

Inhomogeneous E-R random graph with $d = \max np_{ij}$.

Regularize vertices with degrees $> 2d$:
make all degrees $\le 2d$ by reducing the weights of edges arbitrarily.

Theorem. *The adjacency matrix A' of the regularized graph concentrates:*

$$\|A' - \mathbb{E}\,A\| \lesssim \sqrt{d} \quad w.h.p.$$

[Le-Levina-V, Le-V 05]; partial case in [Feige-Ofek] (complete removal of vertices).

Regularization and concentration: theory

Inhomogeneous E-R random graph with $d = \max np_{ij}$.

Regularize vertices with degrees $> 2d$:
make all degrees $\leq 2d$ by reducing the weights of edges arbitrarily.

Theorem. *The adjacency matrix A' of the regularized graph concentrates:*

$$\|A' - \mathbb{E}\, A\| \lesssim \sqrt{d} \quad w.h.p.$$

[Le-Levina-V, Le-V 05]; partial case in [Feige-Ofek] (complete removal of vertices).

The graph can be very sparse, $d = O(1)$.

Regularization and concentration: theory

Inhomogeneous E-R random graph with $d = \max np_{ij}$.

Regularize vertices with degrees $> 2d$:
make all degrees $\leq 2d$ by reducing the weights of edges arbitrarily.

Theorem. *The adjacency matrix A' of the regularized graph concentrates:*

$$\|A' - \mathbb{E}\, A\| \lesssim \sqrt{d} \quad w.h.p.$$

[Le-Levina-V, Le-V 05]; partial case in [Feige-Ofek] (complete removal of vertices).

The graph can be very sparse, $d = O(1)$.

Proof:

1. simple concentration of A in *cut norm*;
2. upgrade to operator norm on a subgraph by *Grothendieck-Pietsch factorization*;
3. *iteration* to extend the control over all graph. □

By-product: a new graph decomposition.

Eigenvectors reveal the latent structure?

Regularization and concentration: applications

Eigenvectors reveal the latent structure?

Concentration (possibly after regularization) $\Rightarrow$

$$A \approx \mathbb{E}\, A.$$

Davis-Kahan theorem $\Rightarrow$ eigenvectors satisfy

$$v_i(A) \approx v_i(\mathbb{E}\, A).$$

Regularization and concentration: applications

Eigenvectors reveal the latent structure?

Concentration (possibly after regularization) $\Rightarrow$

$$A \approx \mathbb{E}\, A.$$

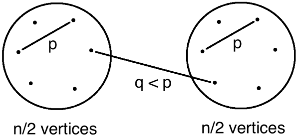
Davis-Kahan theorem $\Rightarrow$ eigenvectors satisfy

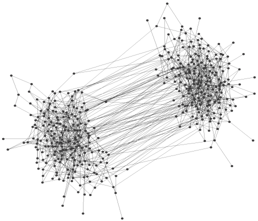
$$v_i(A) \approx v_i(\mathbb{E}\, A).$$

Eigenvectors $v_i(\mathbb{E}\, A)$ carry information about **network structure**.

Eigenvectors reveal the network structure.

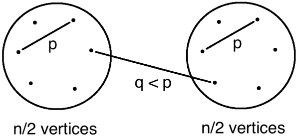
Example. **Community detection** in stochastic block model $G(n, p, q)$.

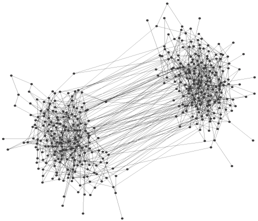




Eigenvectors reveal the network structure.

Example. **Community detection** in stochastic block model $G(n, p, q)$.

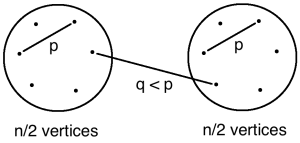


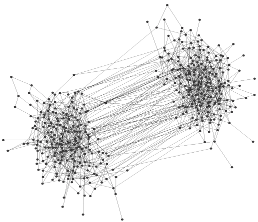


$$\mathbb{E}\,A = \left[\begin{array}{cc|cc} p & p & q & q \\ p & p & q & q \\ \hline q & q & p & p \\ q & q & p & p \end{array}\right] \text{ has rank 2;} \qquad v_1(\mathbb{E}\,A) = \left[\begin{array}{c} 1 \\ 1 \\ \hline 1 \\ 1 \end{array}\right], \quad v_2(\mathbb{E}\,A) = \left[\begin{array}{c} 1 \\ 1 \\ \hline -1 \\ -1 \end{array}\right].$$

Eigenvectors reveal the network structure.

Example. **Community detection** in stochastic block model $G(n, p, q)$.



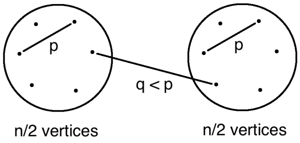


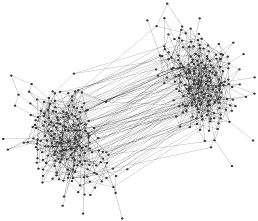
$$\mathbb{E}\,A = \left[\begin{array}{cc|cc} p & p & q & q \\ p & p & q & q \\ \hline q & q & p & p \\ q & q & p & p \end{array}\right] \text{ has rank 2;} \qquad v_1(\mathbb{E}\,A) = \left[\begin{array}{c} 1 \\ 1 \\ \hline 1 \\ 1 \end{array}\right], \quad v_2(\mathbb{E}\,A) = \left[\begin{array}{c} 1 \\ 1 \\ \hline -1 \\ -1 \end{array}\right].$$

$v_2(\mathbb{E}\,A)$ encodes community structure $\Rightarrow$ $v_2(A)$ encodes the structure, too.

Eigenvectors reveal the network structure.

Example. **Community detection** in stochastic block model $G(n, p, q)$.





$$\mathbb{E}\, A = \left[\begin{array}{cc|cc} p & p & q & q \\ p & p & q & q \\ \hline q & q & p & p \\ q & q & p & p \end{array}\right] \text{ has rank 2;} \qquad v_1(\mathbb{E}\, A) = \left[\begin{array}{c} 1 \\ 1 \\ \hline 1 \\ 1 \end{array}\right], \quad v_2(\mathbb{E}\, A) = \left[\begin{array}{c} 1 \\ 1 \\ \hline -1 \\ -1 \end{array}\right].$$

$v_2(\mathbb{E}\, A)$ encodes community structure $\Rightarrow$ $v_2(A)$ encodes the structure, too.

Spectral Clustering Algorithm: given a graph with adjacency matrix A,

- Compute the second leading *eigenvector* of A;
- Recover communities based on the *signs of its coefficients*.

Using eigenvectors: theory.

Corollary **(Community Detection).** Consider the stochastic block model $G(n, p, q)$ with $p = a/n$ and $q = b/n$. Suppose

$$(a-b)^2 \geq C_\varepsilon (a+b).$$

Then the *regularized* spectral clustering algorithm recovers communities up to εn misclassified vertices, and with high probability.

Using eigenvectors: theory.

Corollary **(Community Detection).** Consider the stochastic block model $G(n, p, q)$ with $p = a/n$ and $q = b/n$. Suppose

$$(a-b)^2 \geq C_\varepsilon (a+b).$$

Then the *regularized* spectral clustering algorithm recovers communities up to εn misclassified vertices, and with high probability.

Proof: straightforward consequence of concentration [Le-Levina-V; Le-V 05].

Using eigenvectors: theory.

Corollary **(Community Detection).** Consider the stochastic block model $G(n, p, q)$ with $p = a/n$ and $q = b/n$. Suppose

$$(a-b)^2 \geq C_\varepsilon (a+b).$$

Then the *regularized* spectral clustering algorithm recovers communities up to εn misclassified vertices, and with high probability.

Proof: straightforward consequence of concentration [Le-Levina-V; Le-V 05].

Detection threshold. The condition on is *optimal* up to C_ε, which must $\to \infty$. No algorithm can succeed if

$$(a-b)^2 \leq 2(a+b).$$

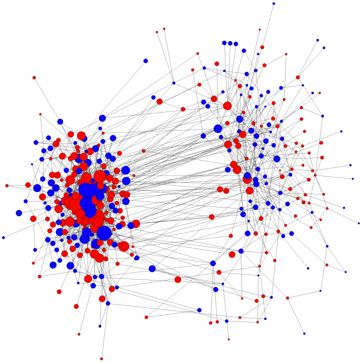
There are algorithms that do *better than random guess* if

$$(a-b)^2 > 2(a+b).$$

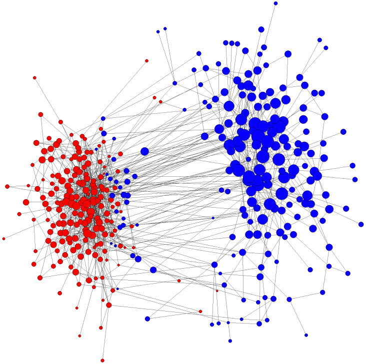
See [Mossel-Neeman-Sly 13-14; Massoulié 13; Bordenave-Lelarge-Massoulié 15].

Performance of regularized spectral clustering

Without regularization



With regularization



$n = 400$ vertices, expected degree 5. Connection probabilities $p = 5/n$ and $b = 0.5/n$.

Application: network visualization by PCA

Further application of

$$\text{eigenstructure}(A) \approx \text{eigenstructure}(\mathbb{E}\, A).$$

Application: network visualization by PCA

Further application of

$$\text{eigenstructure}(A) \approx \text{eigenstructure}(\mathbb{E}\, A).$$

Assume $\mathbb{E}\, A$ has low rank, exactly or approximately.
Then PCA on A should **reveal the latent structure** of the network.

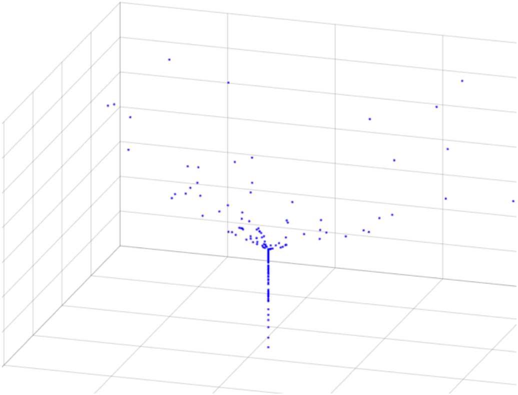
How? project the columns of A onto the space of the 3 leading eigenvectors.

Application: network visualization by PCA

Power grid of U.S.A.

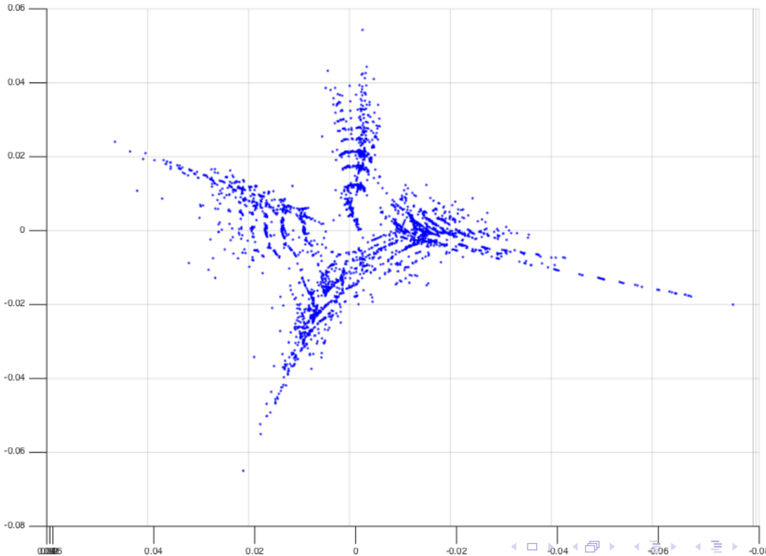
Application: network visualization by PCA

Without regularization:



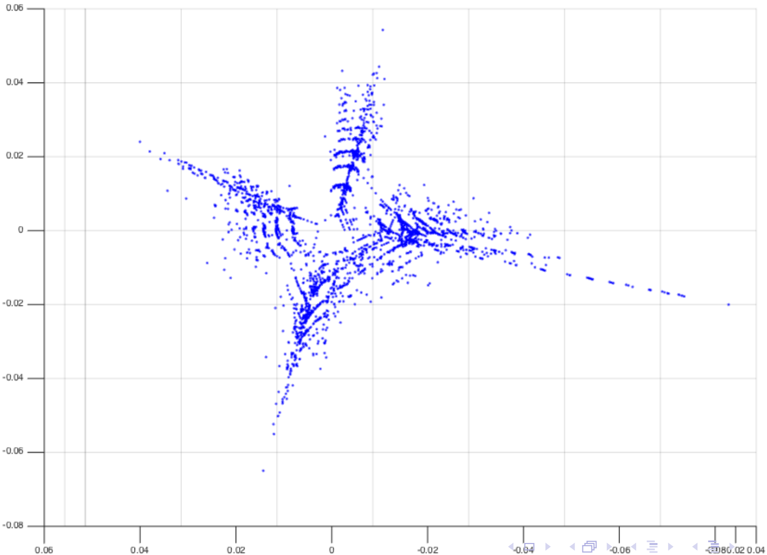
Not very useful...

With regularization:



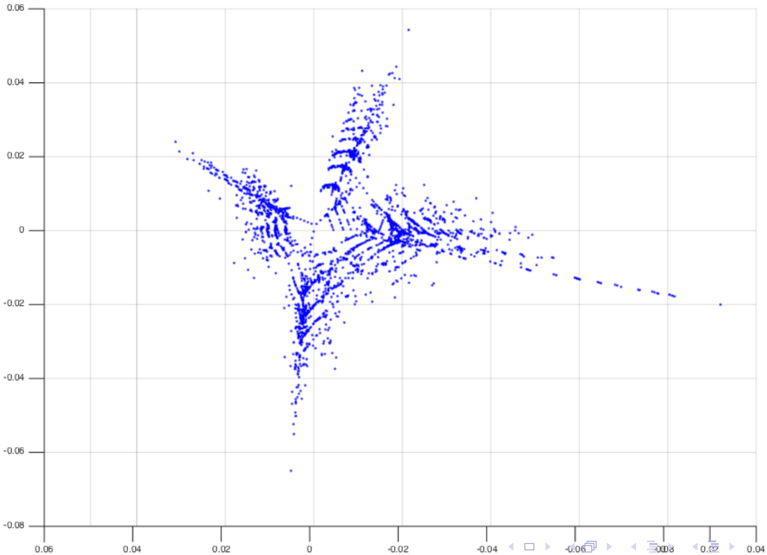
Application: network visualization by PCA

With regularization:



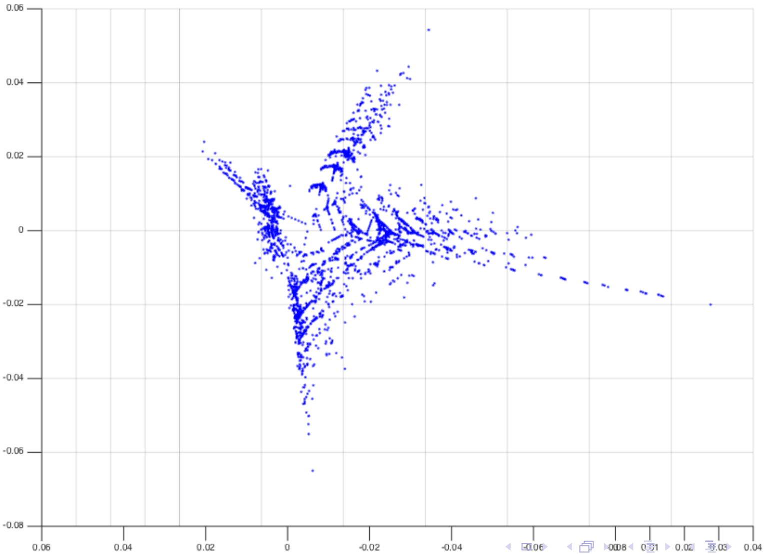
Application: network visualization by PCA

With regularization:



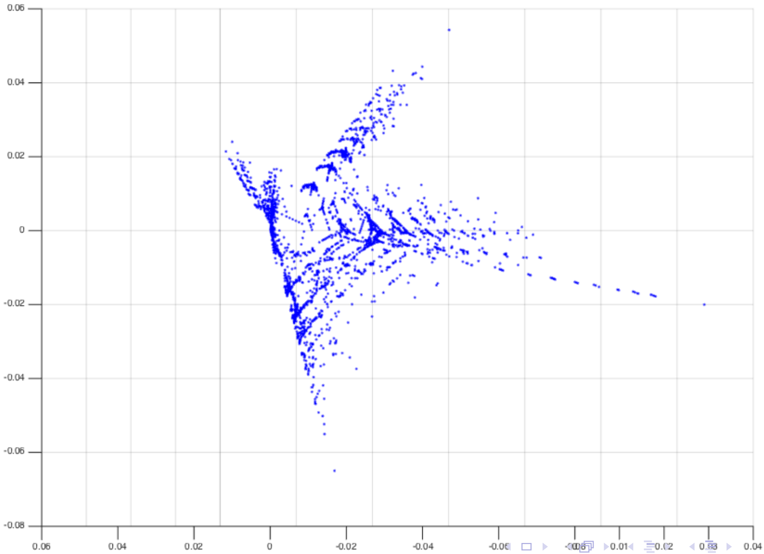
Application: network visualization by PCA

With regularization:

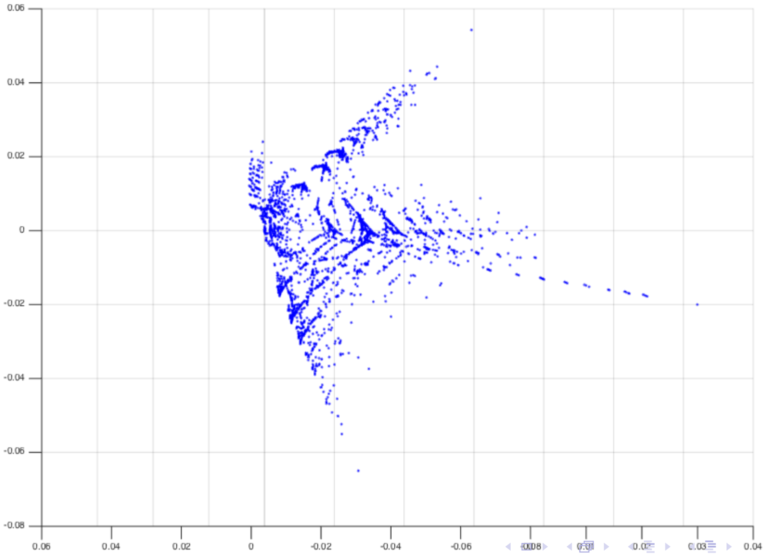


Application: network visualization by PCA

With regularization:

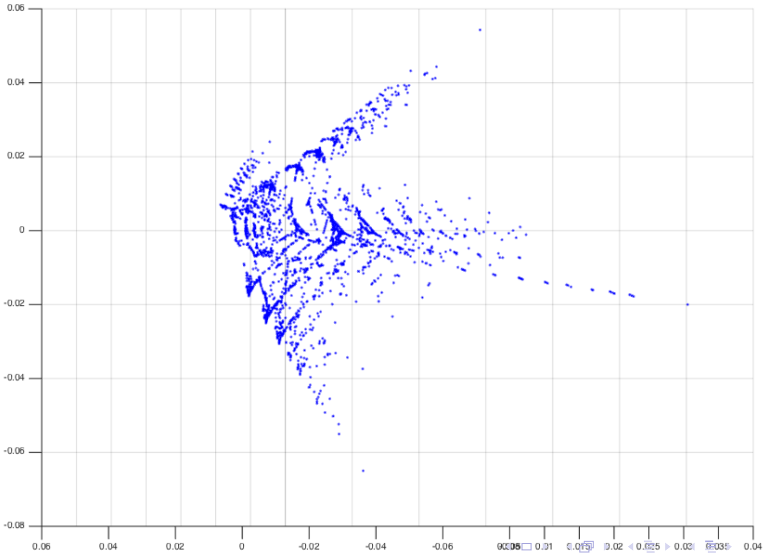


With regularization:

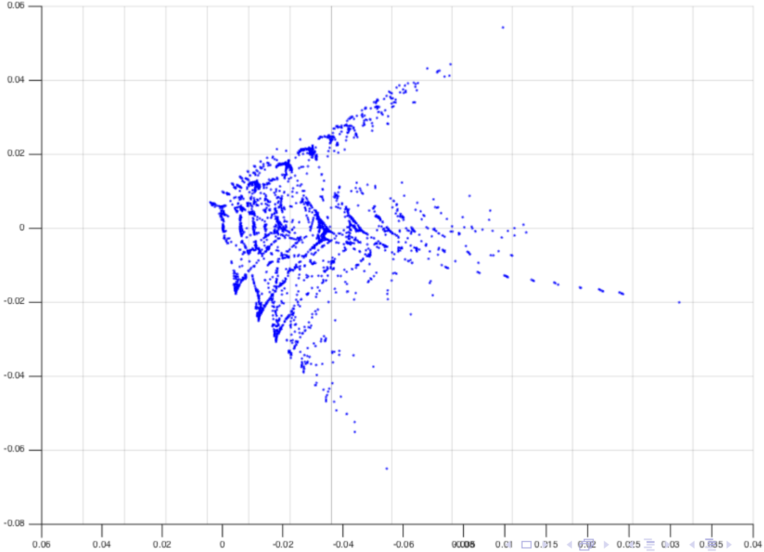


Application: network visualization by PCA

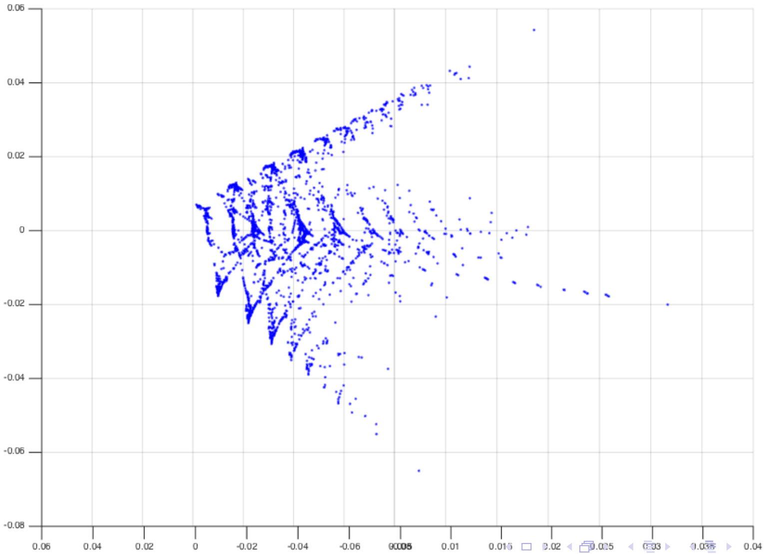
With regularization:



With regularization:

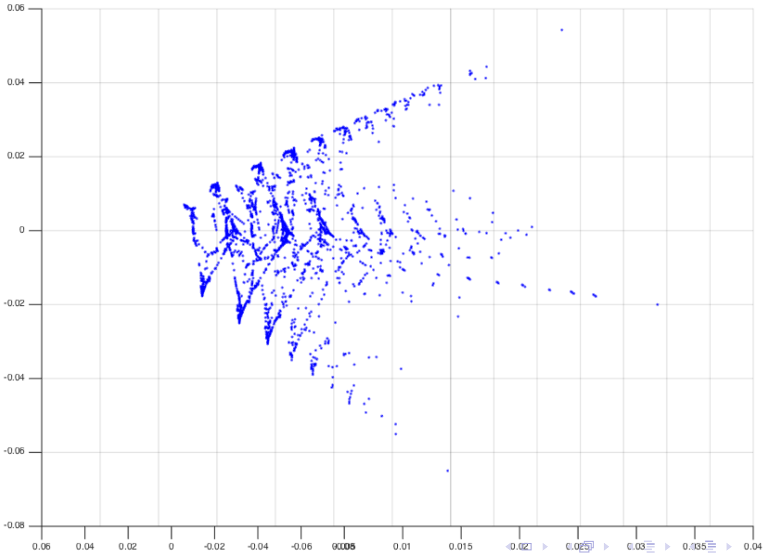


With regularization:



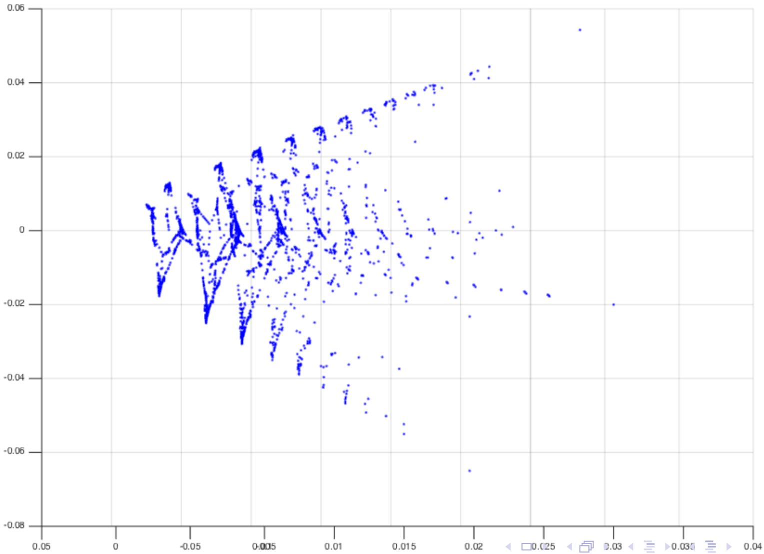
Application: network visualization by PCA

With regularization:



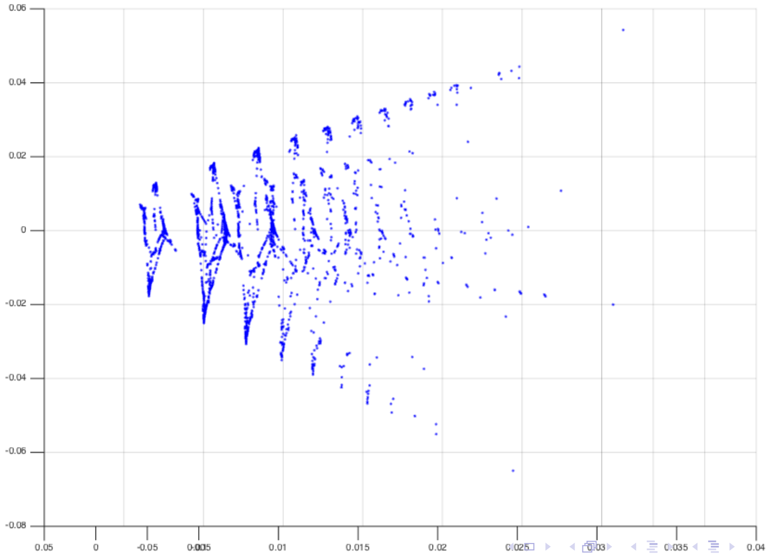
Application: network visualization by PCA

With regularization:



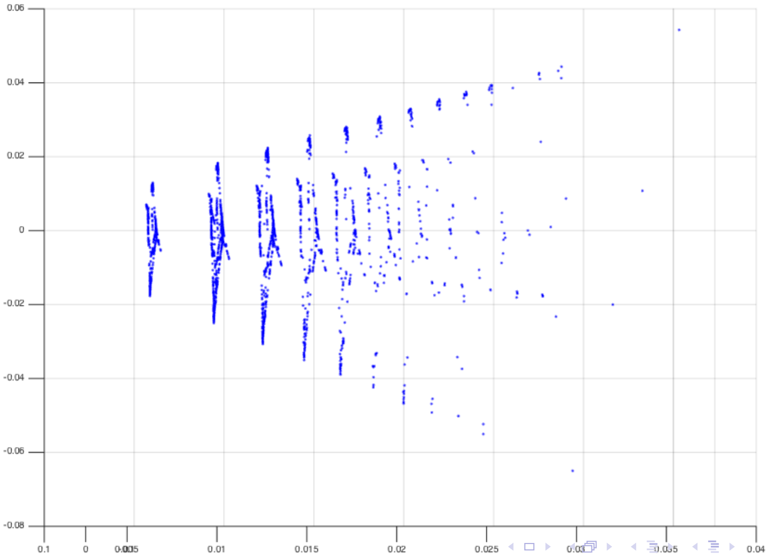
Application: network visualization by PCA

With regularization:



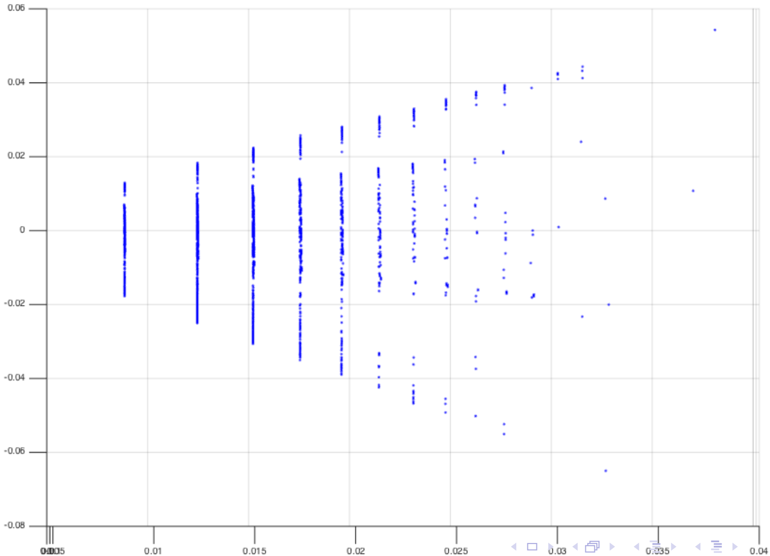
Application: network visualization by PCA

With regularization:



Application: network visualization by PCA

With regularization:



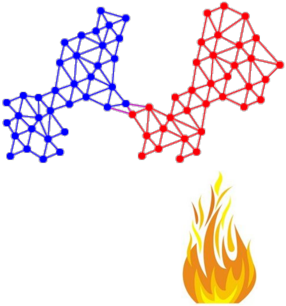
Graph Laplacian

Diffusion approach: heat the graph.

Graph Laplacian

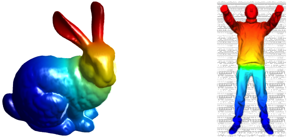
Diffusion approach: heat the graph.

The heat gets trapped in a community $\Rightarrow$ can recover it.



Graph Laplacian

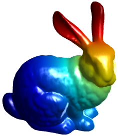
In $\mathbb{R}^2$, the **heat diffusion** is described by the Laplacian $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$.



From Gabriel Peyré's manifold methods class (left); Morpheo research team (right)

Graph Laplacian

In $\mathbb{R}^2$, the **heat diffusion** is described by the Laplacian $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$.

From Gabriel Peyré's manifold methods class (left); Morpheo research team (right)

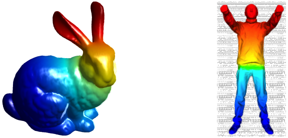
On a graph, the discrete Laplacian is the $n \times n$ matrix

$$\Delta := I - D^{-1/2} A D^{-1/2}$$

where D is the diagonal matrix with the *degrees* on the diagonal.

Graph Laplacian

In $\mathbb{R}^2$, the **heat diffusion** is described by the Laplacian $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$.



From Gabriel Peyré's manifold methods class (left); Morpheo research team (right)

On a graph, the discrete Laplacian is the $n \times n$ matrix

$$\Delta := I - D^{-1/2} A D^{-1/2}$$

where D is the diagonal matrix with the *degrees* on the diagonal.

Adjacency and Laplacian are two most fundamental matrices associated to graphs.

Concentration of Laplacian

Concentration of Laplacian

For **dense graphs** (expected degrees $d \gtrsim \log n$), Laplacian concentrates.

Concentration of Laplacian

For **dense graphs** (expected degrees $d \gtrsim \log n$), Laplacian concentrates.

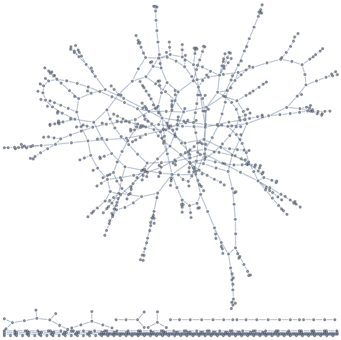
For **sparse graphs** ($d \ll \log n$), fails to concentrate.

Concentration of Laplacian

For **dense graphs** (expected degrees $d \gtrsim \log n$), Laplacian concentrates.

For **sparse graphs** ($d \ll \log n$), fails to concentrate.

What's wrong? Low-degree vertices: isolated vertices, trees. (They get overheated.)



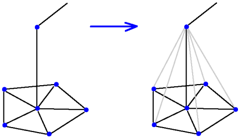
Concentration of Laplacian

Would regularization help?

Concentration of Laplacian

Would regularization help?

Connect low-degree vertices to the rest of the graph by *light weighted edges*; bring up all degrees to $\sim d$.



$$\Longleftrightarrow \qquad A_{ij} \mapsto A_{ij} + d/n$$

Proposed by network scientists [Chaudhuri+ 12, Amini+ 13].

Concentration of regularized Laplacian: theory

Theorem. *The Laplacian* Δ' *of the regularized graph concentrates:*

$$\|\Delta' - \mathbb{E}\,\Delta'\| \lesssim \frac{1}{\sqrt{d}} \quad \textit{while} \quad \|\Delta'\| \sim 1.$$

[Le-Levina-V, Le-V 05].

Concentration of regularized Laplacian: theory

Theorem. *The Laplacian Δ' of the regularized graph concentrates:*

$$\|\Delta' - \mathbb{E}\,\Delta'\| \lesssim \frac{1}{\sqrt{d}} \quad \textit{while} \quad \|\Delta'\| \sim 1.$$

[Le-Levina-V, Le-V 05].

Proof: Deduced from concentration of regularized adjacency matrices.

Concentration of regularized Laplacian: theory

Theorem. *The Laplacian Δ' of the regularized graph concentrates:*

$$\|\Delta' - \mathbb{E}\,\Delta'\| \lesssim \frac{1}{\sqrt{d}} \quad \textit{while} \quad \|\Delta'\| \sim 1.$$

[Le-Levina-V, Le-V 05].

Proof: Deduced from concentration of regularized adjacency matrices.

Application to community detection: use the 2^{nd} eigenvector of the Laplacian. Theoretical performance: same as for adjacency; empirically even better.

Concentration of regularized Laplacian: theory

Theorem. *The Laplacian Δ' of the regularized graph concentrates:*

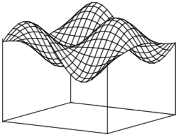
$$\|\Delta' - \mathbb{E}\,\Delta'\| \lesssim \frac{1}{\sqrt{d}} \quad \textit{while} \quad \|\Delta'\| \sim 1.$$

[Le-Levina-V, Le-V 05].

Proof: Deduced from concentration of regularized adjacency matrices.

Application to community detection: use the 2^{nd} eigenvector of the Laplacian. Theoretical performance: same as for adjacency; empirically even better.

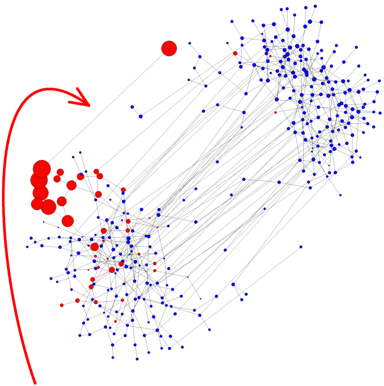
Physical interpretation: Make the graph vibrate; the wave with lowest frequency recovers the communities.



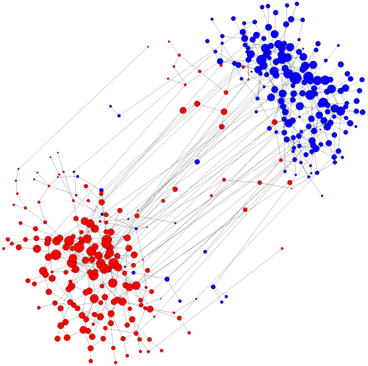
Performance of regularized spectral clustering

Artificial data: sparse stochastic block model

Without regularization



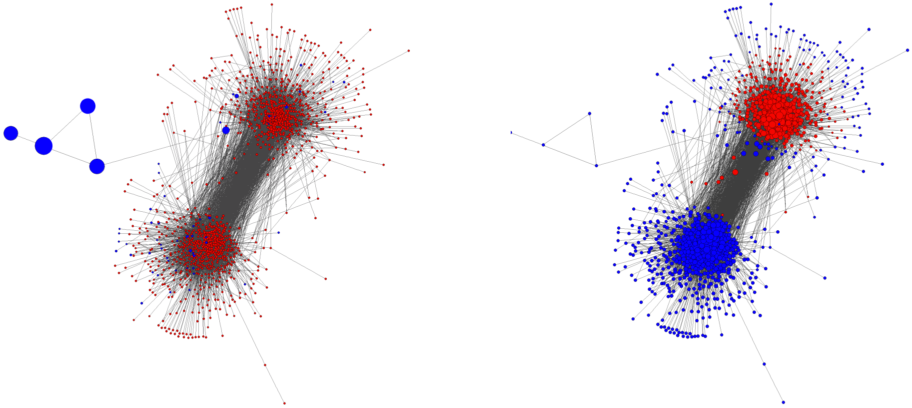
With regularization



This tree gets overheated

Real data: political blogs after 2004 U.S. presidential election [Adamic-Glance 04].

Without regularization — With regularization



1,222 vertices (liberal/conservative); edges = hyperlinks; average degree = 27.

Optimization Methods

Goal: fit the desired type of structure to a given network.

Optimization Methods

Goal: fit the desired type of structure to a given network.

Strongest community structure: *union of cliques*.

How to fit? Maximize correlation between the network and a union of cliques.

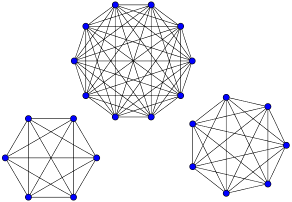
Optimization Methods

Goal: fit the desired type of structure to a given network.

Strongest community structure: *union of cliques*.

How to fit? Maximize correlation between the network and a union of cliques.

Optimization: $\max \langle A, Z \rangle$ where $A =$ adjacency matrix of the network, $Z =$ adjacency matrix of a union of cliques with k edges.



$$Z = \begin{bmatrix} 1 & 1 & 1 & 1 & & & & & \\ 1 & 1 & 1 & 1 & & & & & \\ 1 & 1 & 1 & 1 & & & & & \\ 1 & 1 & 1 & 1 & & & & & \\ & & & & 1 & 1 & 1 & & \\ & & & & 1 & 1 & 1 & & \\ & & & & 1 & 1 & 1 & & \\ & & & & & & & 1 & 1 \\ & & & & & & & 1 & 1 \end{bmatrix}$$

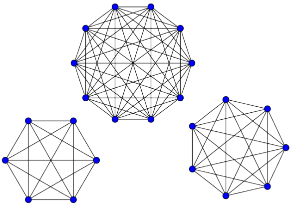
Optimization Methods

Goal: fit the desired type of structure to a given network.

Strongest community structure: *union of cliques.*

How to fit? Maximize correlation between the network and a union of cliques.

Optimization: $\max \langle A, Z \rangle$ where $A =$ adjacency matrix of the network, $Z =$ adjacency matrix of a union of cliques with k edges.



$$Z = \begin{bmatrix} 1 & 1 & 1 & 1 & & & & & \\ 1 & 1 & 1 & 1 & & & & & \\ 1 & 1 & 1 & 1 & & & & & \\ 1 & 1 & 1 & 1 & & & & & \\ & & & & 1 & 1 & 1 & & \\ & & & & 1 & 1 & 1 & & \\ & & & & 1 & 1 & 1 & & \\ & & & & & & & 1 & 1 \\ & & & & & & & 1 & 1 \end{bmatrix}$$

Optimization. $\max \langle A, Z \rangle$: $Z \in \{0,1\}^{n \times n}$ is block-diagonal, $\sum Z_{ij} = k$.

Integer optimization problem. NP-hard.

Semidefinite relaxation

Optimization. $\max \langle A, Z\rangle$: $Z \in \{0,1\}^{n\times n}$ is block-diagonal, $\sum Z_{ij} = k$.

Fact. A matrix $Z \in \{0,1\}^{n\times n}$ is block diagonal $\Leftrightarrow$ Z is positive semidefinite.

Semidefinite relaxation

Optimization. $\max \langle A, Z\rangle$: $Z \in \{0,1\}^{n\times n}$ is block-diagonal, $\sum Z_{ij} = k$.

Fact. A matrix $Z \in \{0,1\}^{n\times n}$ is block diagonal $\Leftrightarrow$ Z is positive semidefinite.

A semidefinite relaxation:

SDP. $\max \langle A, Z\rangle$: $Z \in [0,1]^{n\times n}$ is positive semidefinite, $\sum Z_{ij} = k$.

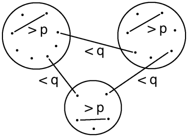
Semidefinite relaxation: theory

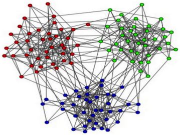
SDP. $\max \langle A, Z\rangle$: $Z \in [0,1]^{n\times n}$ is positive semidefinite, $\sum Z_{ij} = k$.

Semidefinite relaxation: theory

SDP. $\max \langle A, Z \rangle$: $Z \in [0,1]^{n\times n}$ is positive semidefinite, $\sum Z_{ij} = k$.

General stochastic block model: $\forall$ many communities, $\forall$ connection probabilities p_{ij}, *within* communities $> p$; *across* communities $< q$. (Not necessarily low rank!)

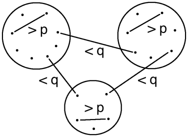


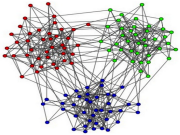


Semidefinite relaxation: theory

SDP. $\max \langle A, Z \rangle$: $Z \in [0,1]^{n \times n}$ is positive semidefinite, $\sum Z_{ij} = k$.

General stochastic block model: $\forall$ many communities, $\forall$ connection probabilities p_{ij}, *within* communities $> p$; *across* communities $< q$. (Not necessarily low rank!)





Theorem (Community Detection by SDP). Consider a general stochastic block model with $p = a/n$ and $q = b/n$. Suppose

$$(a-b)^2 \geq C_\varepsilon (a+b).$$

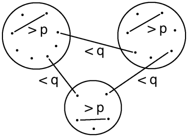
Then the SDP (with k=number of edges) recovers communities up to εn misclassified vertices, and with high probability.

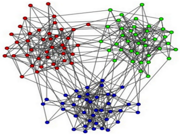
[Guedon-V. 14].

Semidefinite relaxation: theory

SDP. $\max \langle A, Z \rangle$: $Z \in [0,1]^{n \times n}$ is positive semidefinite, $\sum Z_{ij} = k$.

General stochastic block model: $\forall$ many communities, $\forall$ connection probabilities p_{ij}, *within* communities $> p$; *across* communities $< q$. (Not necessarily low rank!)





Theorem (Community Detection by SDP). Consider a general stochastic block model with $p = a/n$ and $q = b/n$. Suppose

$$(a-b)^2 \geq C_\varepsilon (a+b).$$

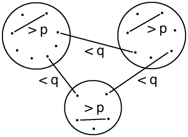
Then the SDP (with k=number of edges) recovers communities up to εn misclassified vertices, and with high probability.

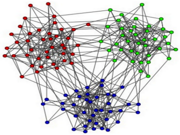
[Guedon-V. 14]. Proof: Grothendieck inequality + concentration in cut norm. $\square$

Semidefinite relaxation: theory

SDP. $\max \langle A, Z \rangle$: $Z \in [0,1]^{n \times n}$ is positive semidefinite, $\sum Z_{ij} = k$.

General stochastic block model: $\forall$ many communities, $\forall$ connection probabilities p_{ij}, *within* communities $> p$; *across* communities $< q$. (Not necessarily low rank!)





Theorem **(Community Detection by SDP).** Consider a general stochastic block model with $p = a/n$ and $q = b/n$. Suppose

$$(a-b)^2 \geq C_\varepsilon (a+b).$$

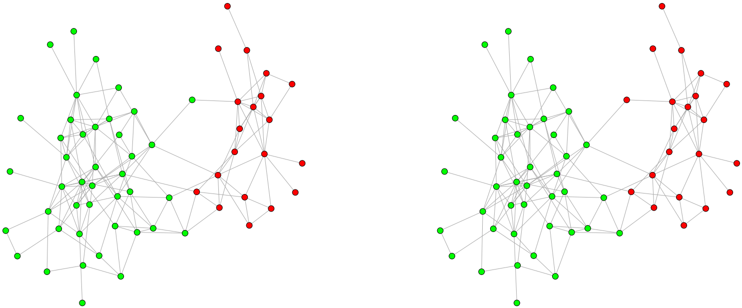
Then the SDP (with k=number of edges) recovers communities up to εn misclassified vertices, and with high probability.

[Guedon-V. 14]. Proof: Grothendieck inequality + concentration in cut norm. □

Exact recovery for *dense* networks ($a, b \geq \log n$); thresholds known [Abbe et al. 14].

Semidefinite relaxation in action

Example. Dolphins in Doubtful Sound, New Zealand [Lusseau et al. 03].



True communities

Communities found by SDP

Semidefinite relaxation in action

Take a closer look at

SDP. $\max \langle A, Z \rangle$: $Z \in [0,1]^{n \times n}$ is positive semidefinite, $\sum Z_{ij} = k$.

Semidefinite relaxation in action

Take a closer look at

SDP. $\max \langle A, Z\rangle$: $Z \in [0,1]^{n\times n}$ is positive semidefinite, $\sum Z_{ij} = k$.

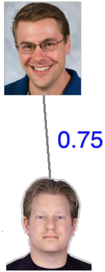
Output: k strongest *"latent bonds"* between vertices.

Semidefinite relaxation in action

Take a closer look at

SDP. $\max \langle A, Z \rangle$: $Z \in [0,1]^{n\times n}$ is positive semidefinite, $\sum Z_{ij} = k$.

Output: k strongest *"latent bonds"* between vertices.

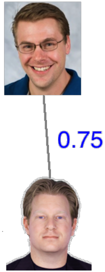


Semidefinite relaxation in action

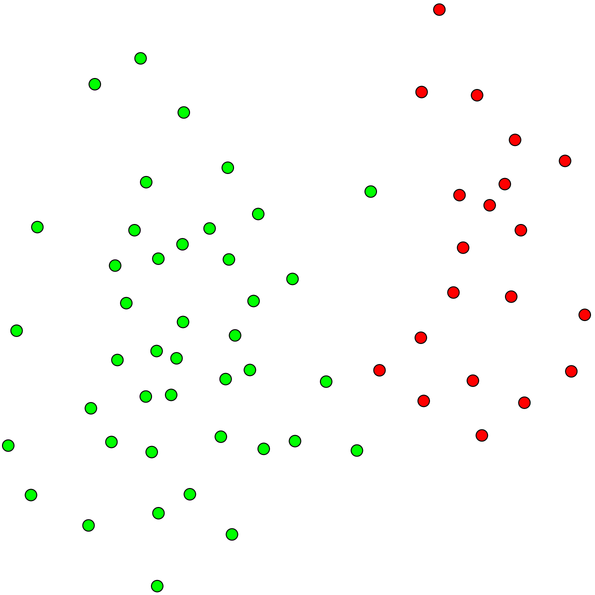
Take a closer look at

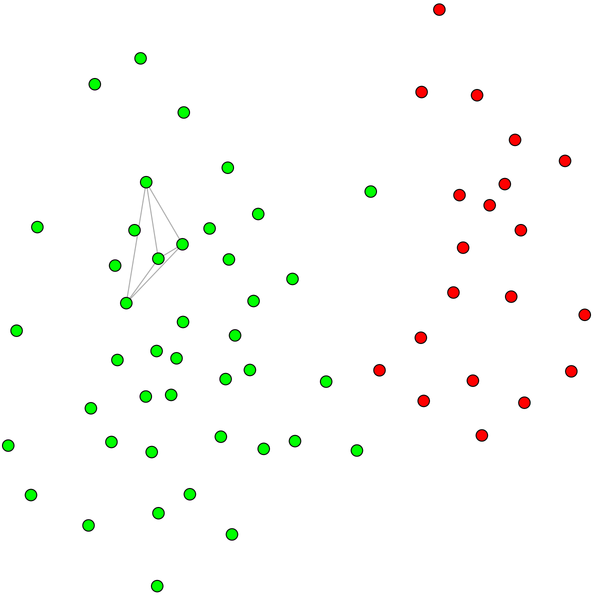
SDP. $\max \langle A, Z \rangle$: $Z \in [0,1]^{n \times n}$ is positive semidefinite, $\sum Z_{ij} = k$.

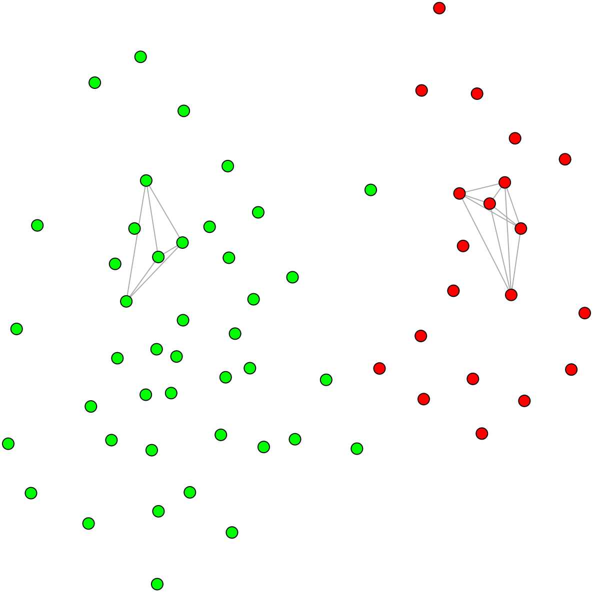
Output: k strongest *"latent bonds"* between vertices.

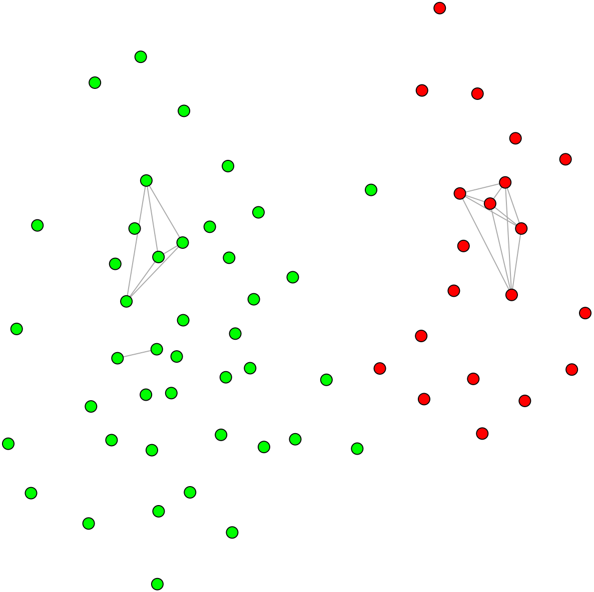


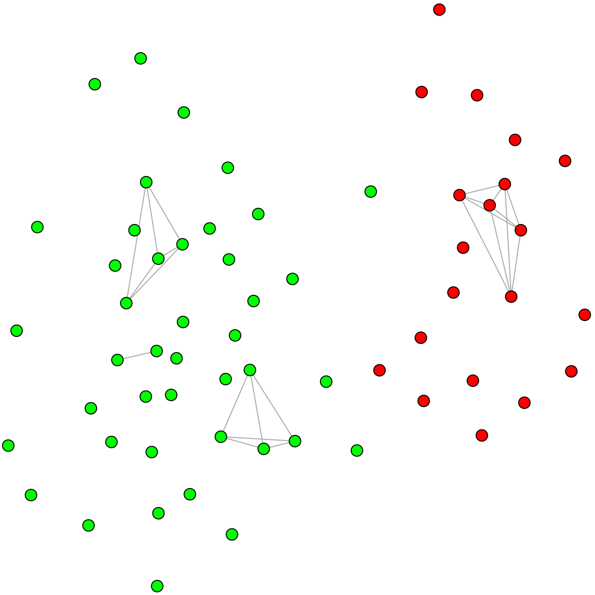
Next slide: increase k gradually $\Rightarrow$ dynamic picture.

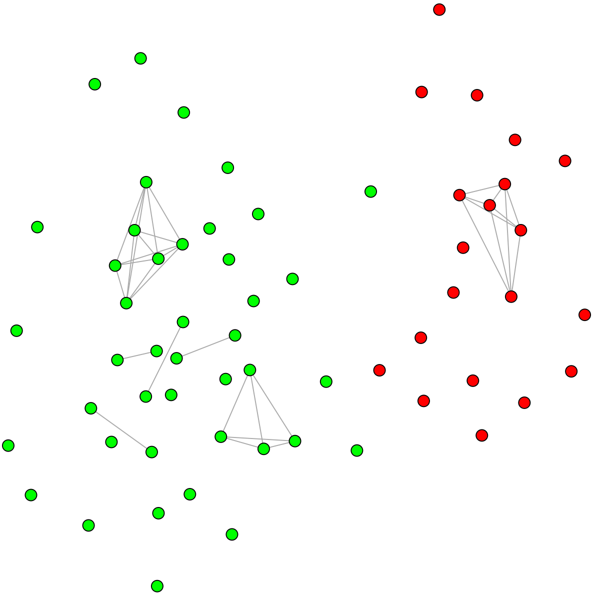


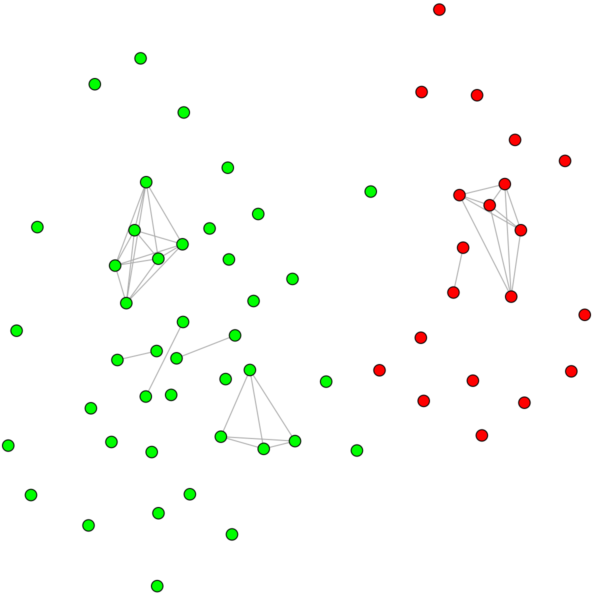


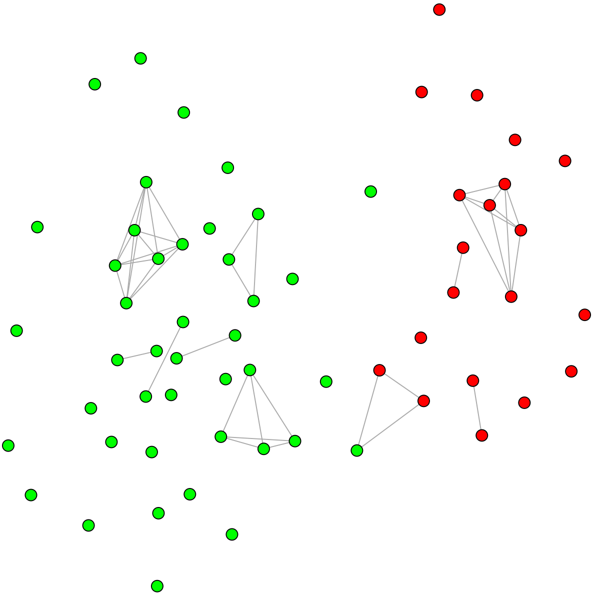


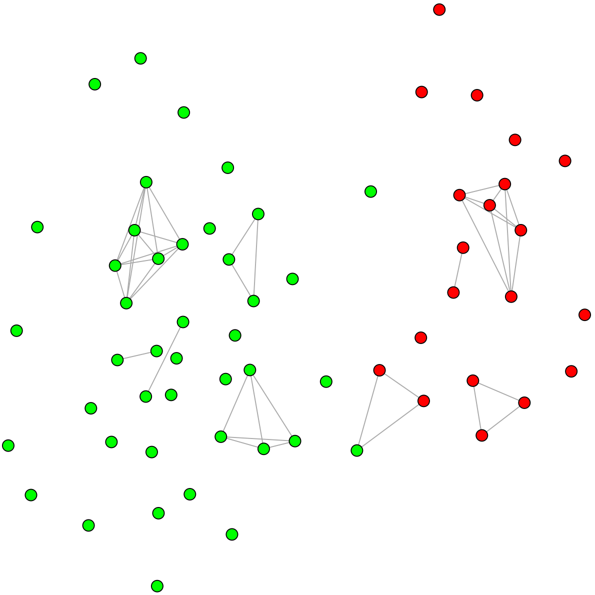


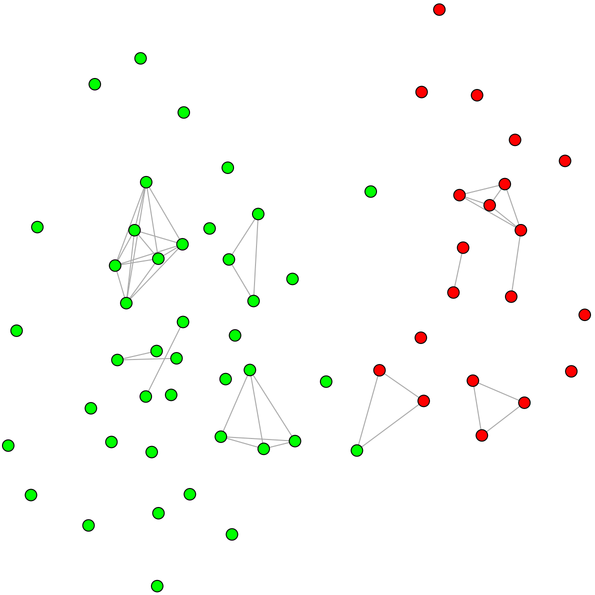


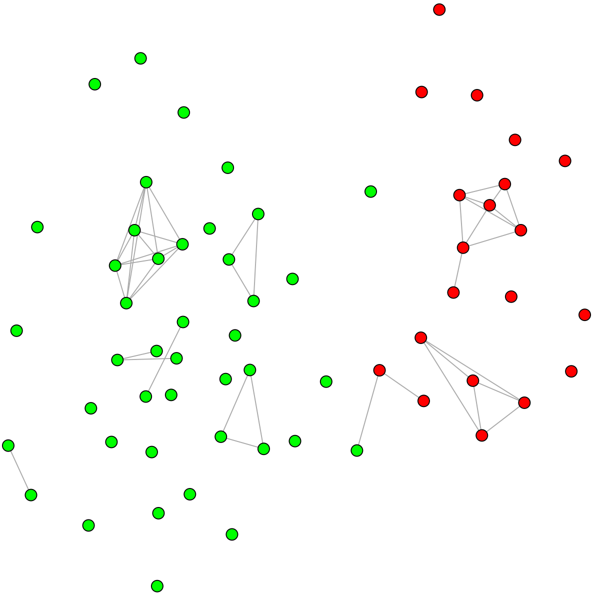


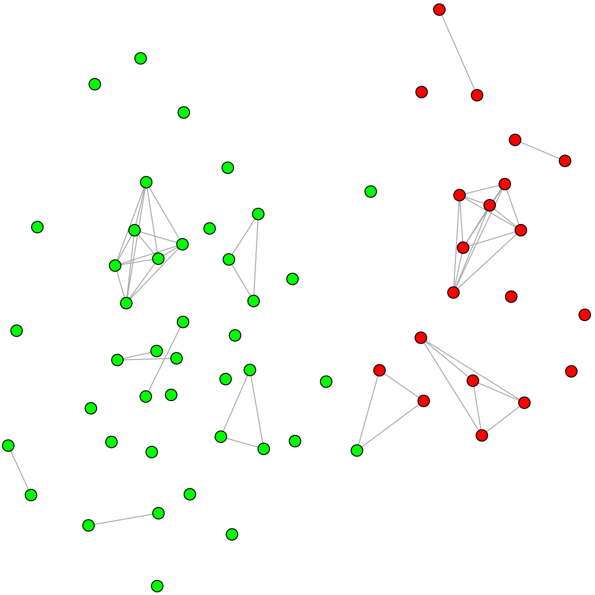


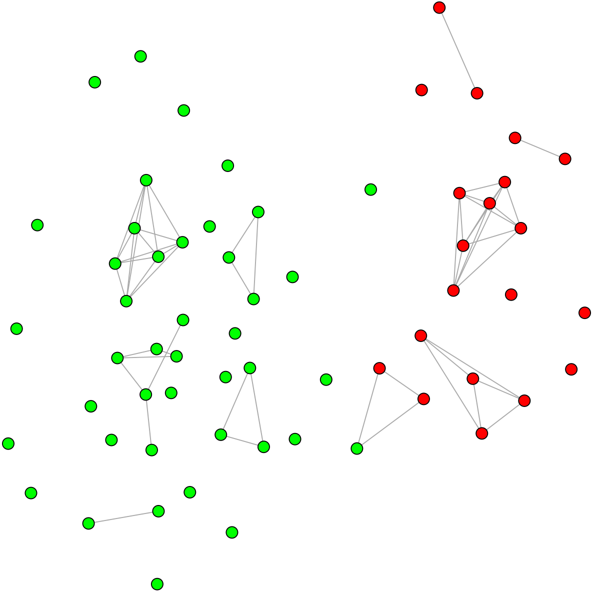


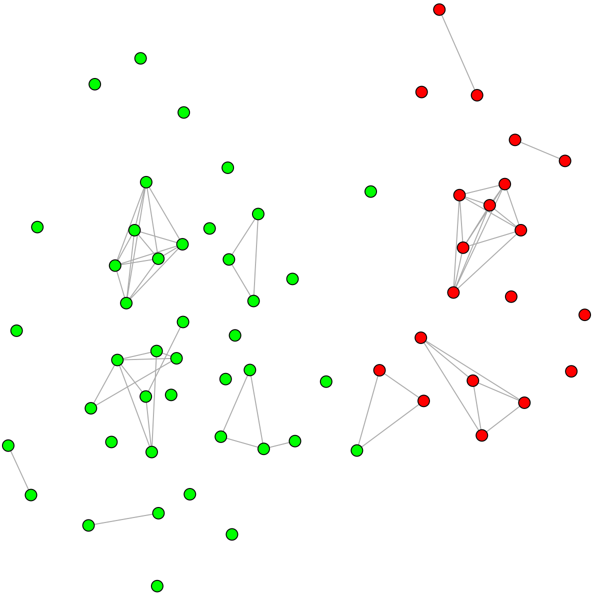


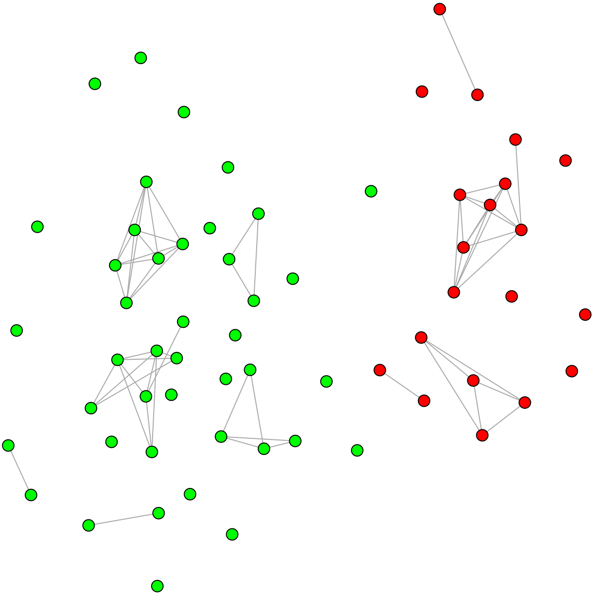


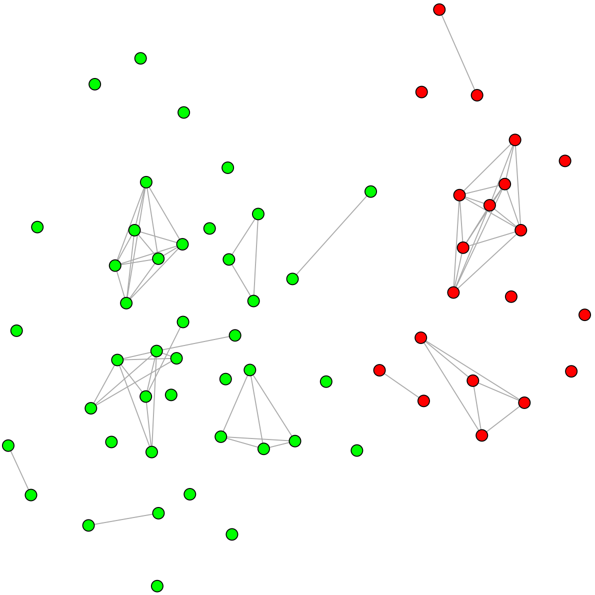


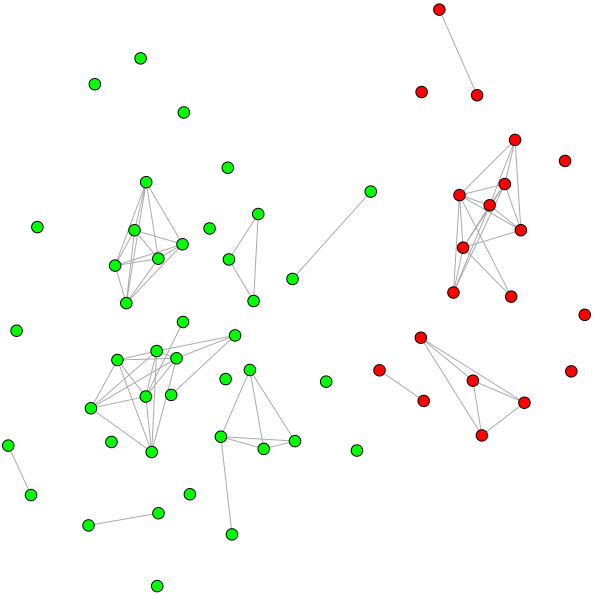


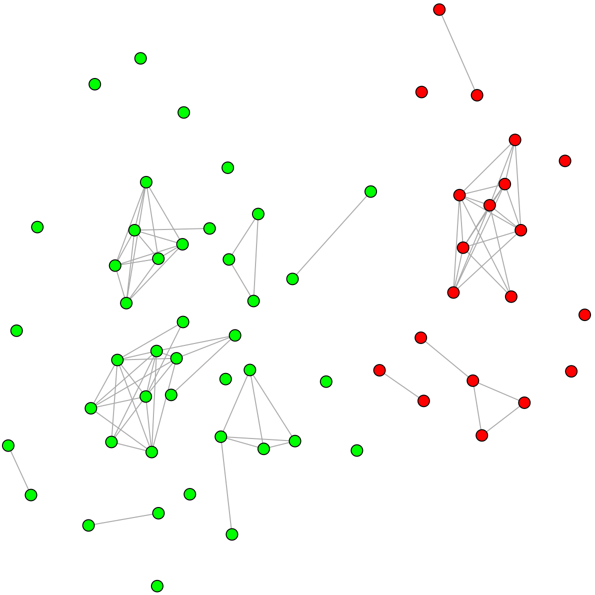


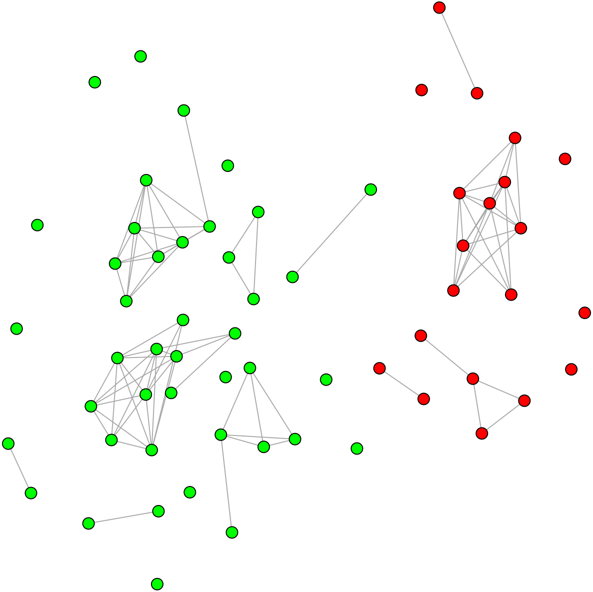


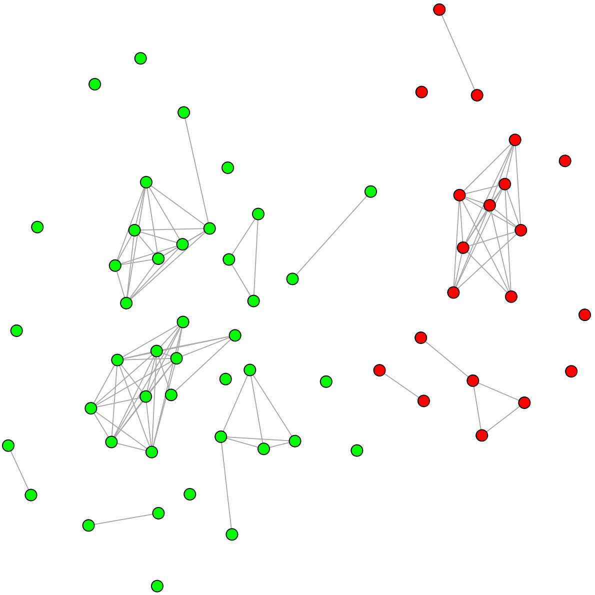


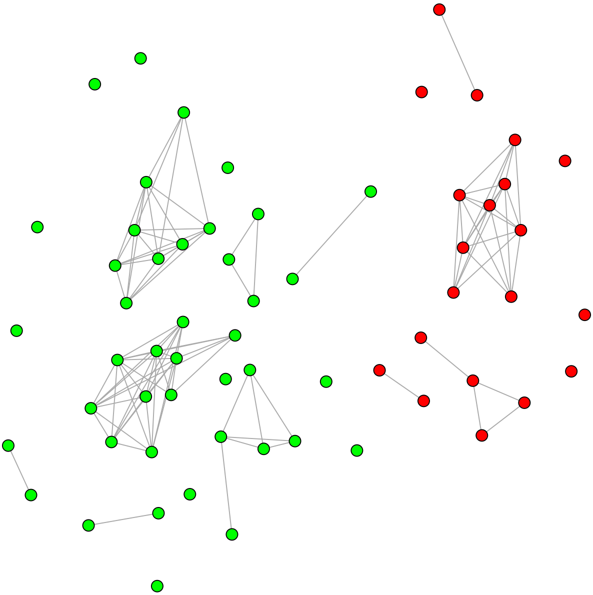


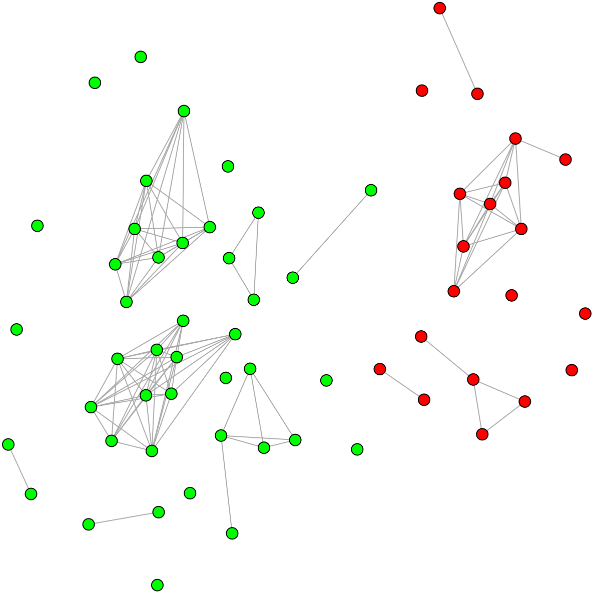


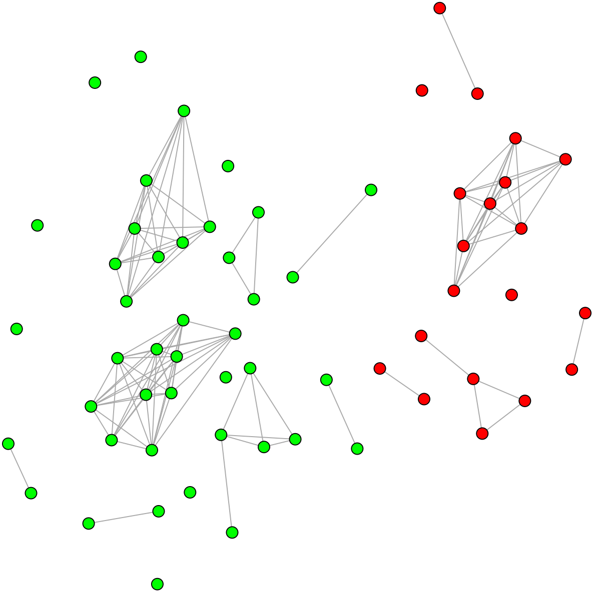


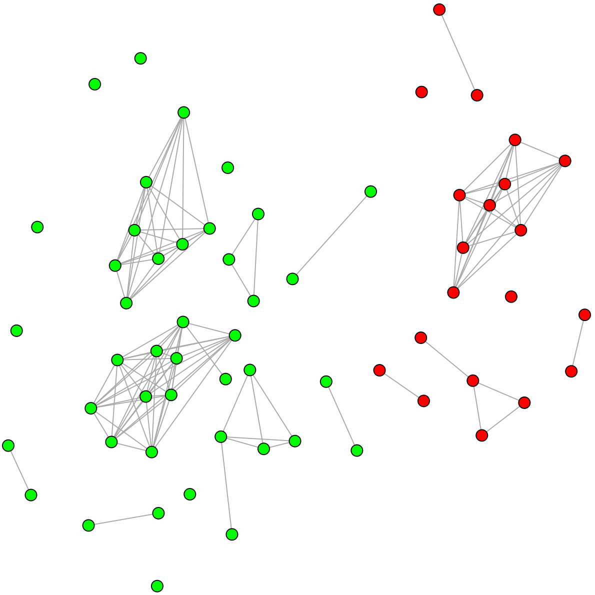


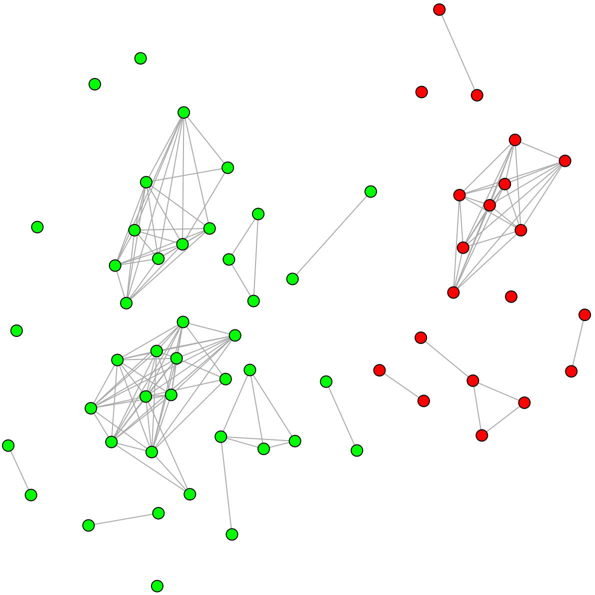


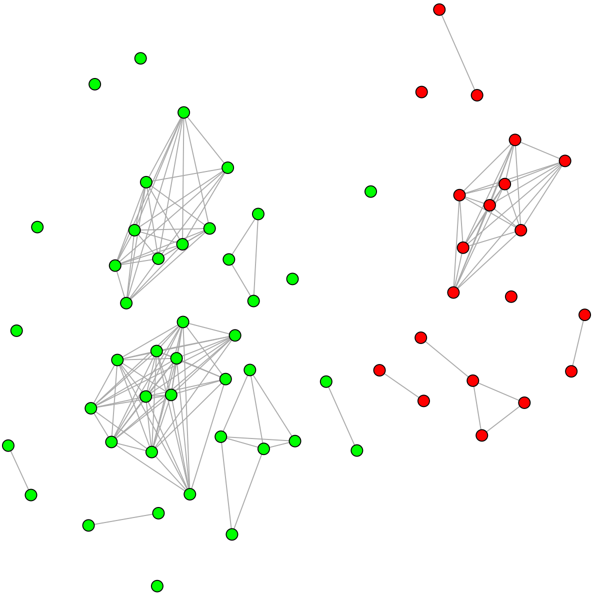


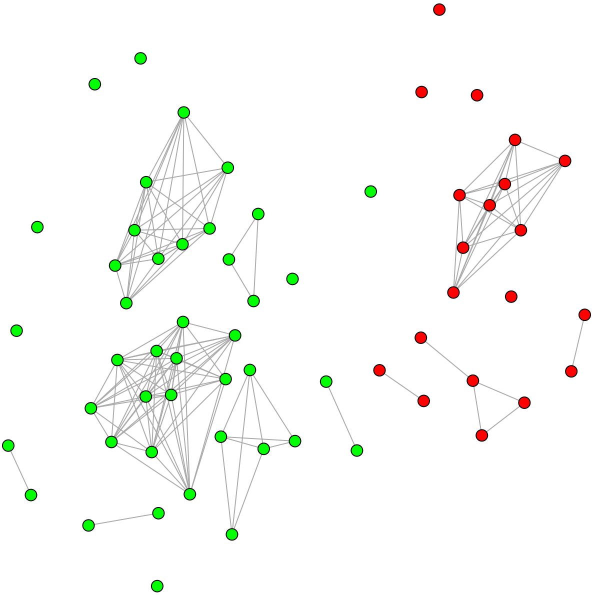


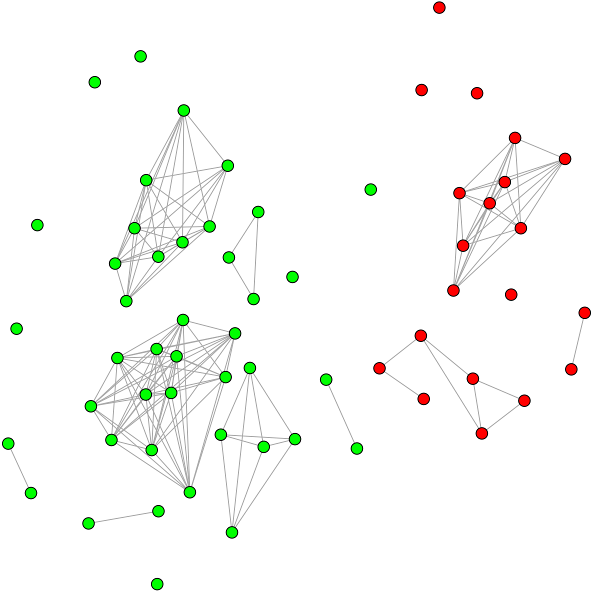


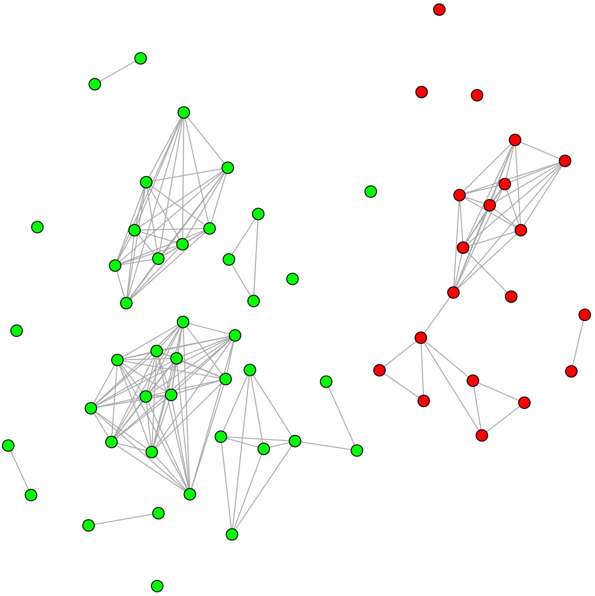


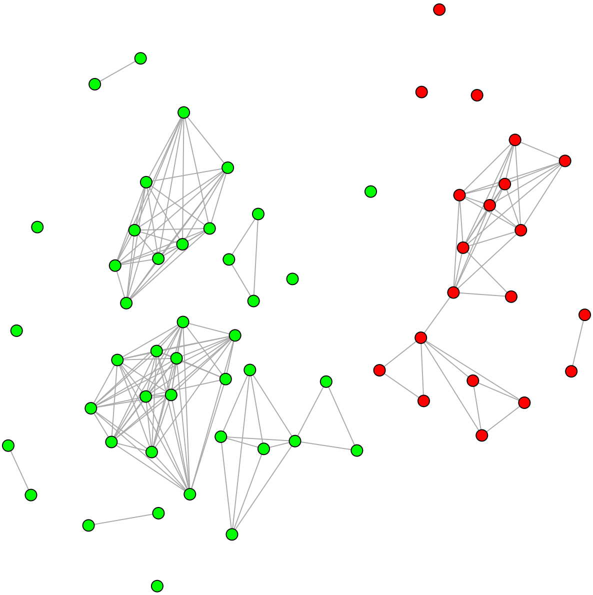


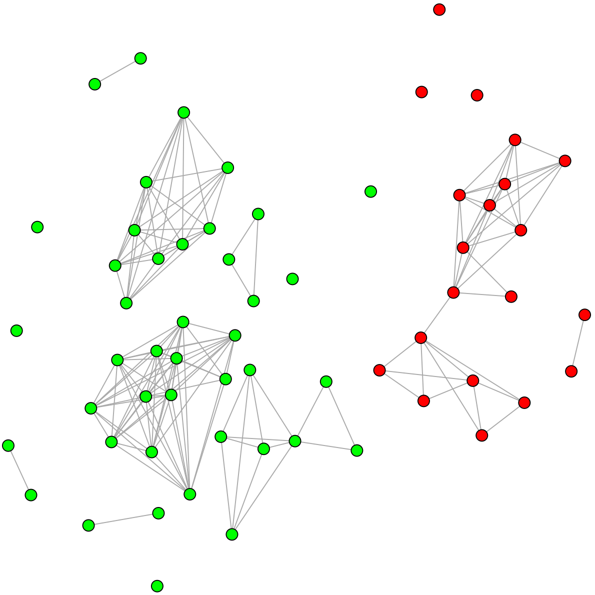


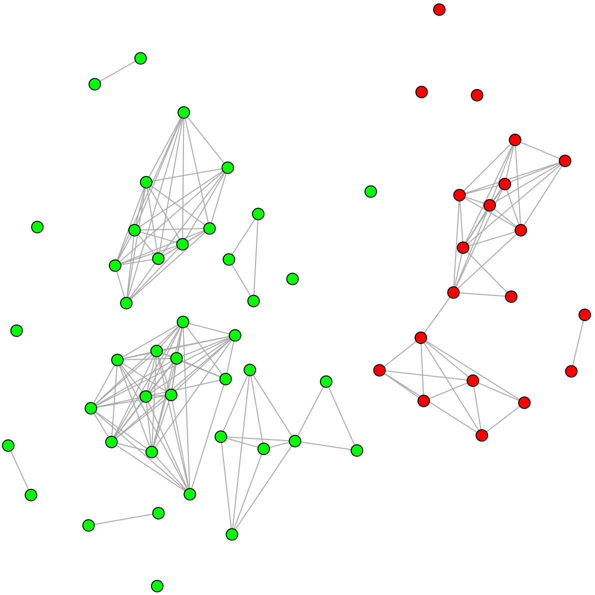


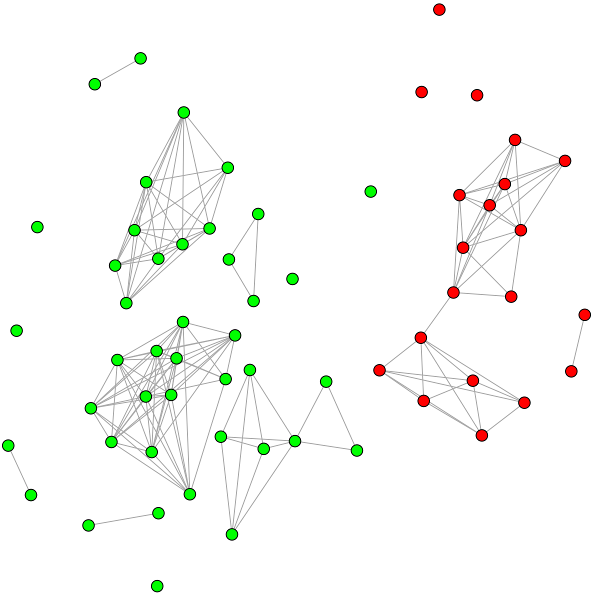


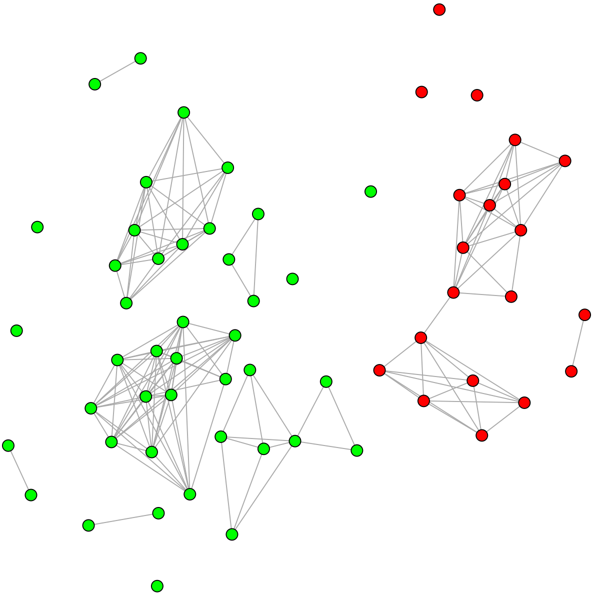


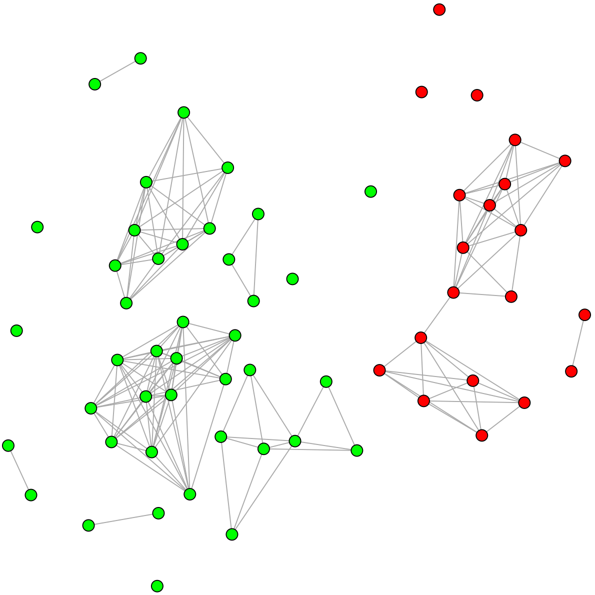


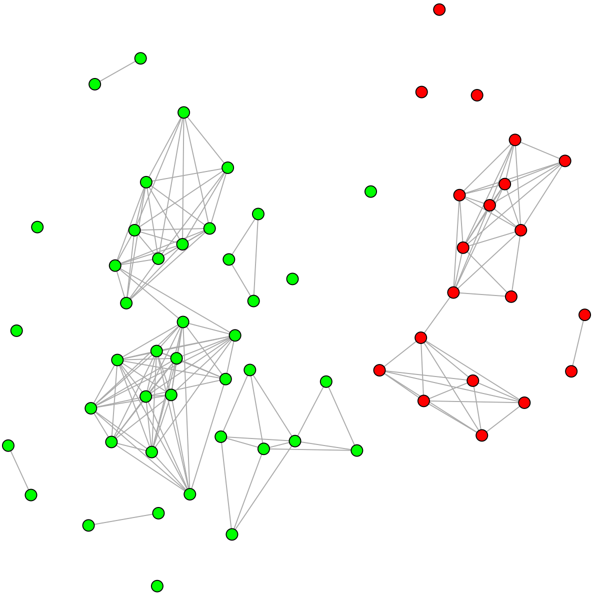


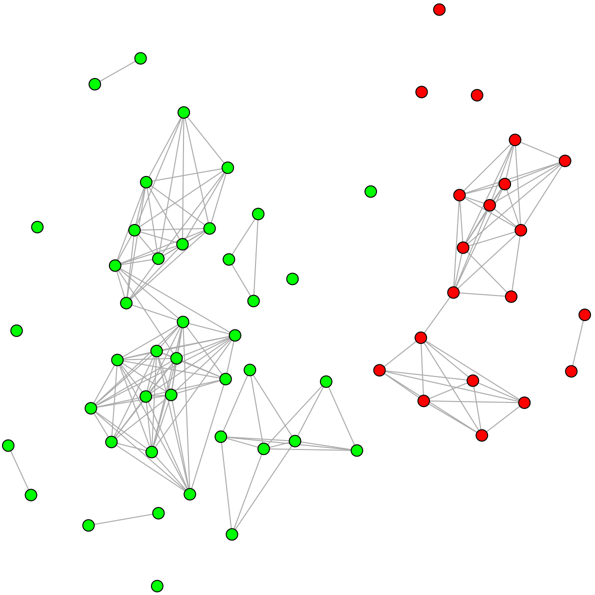


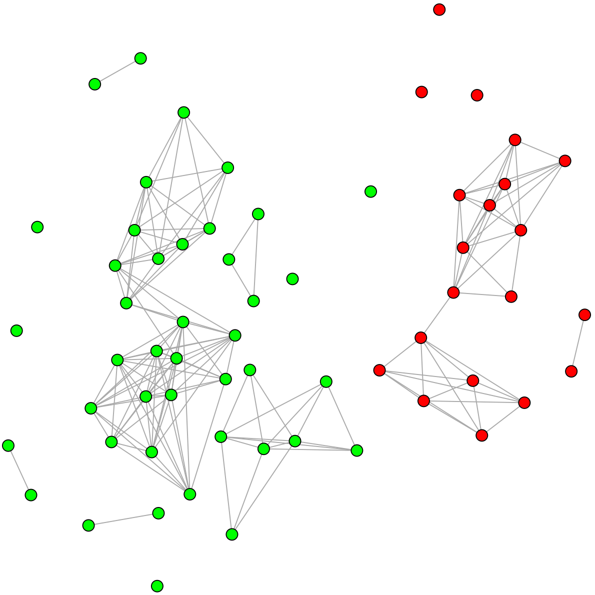


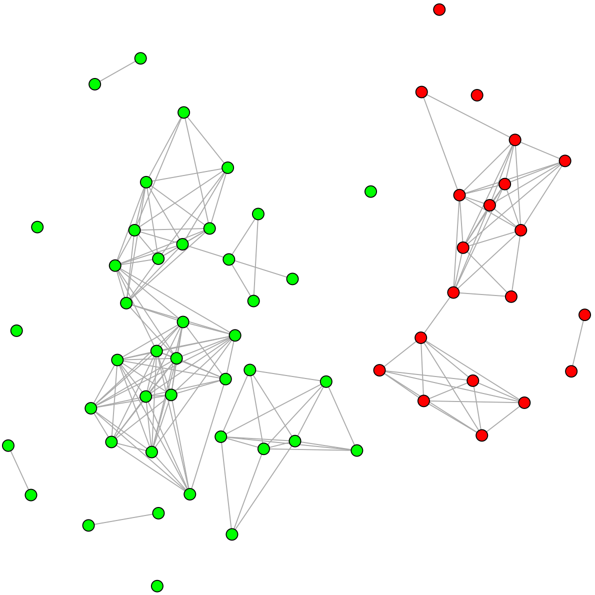


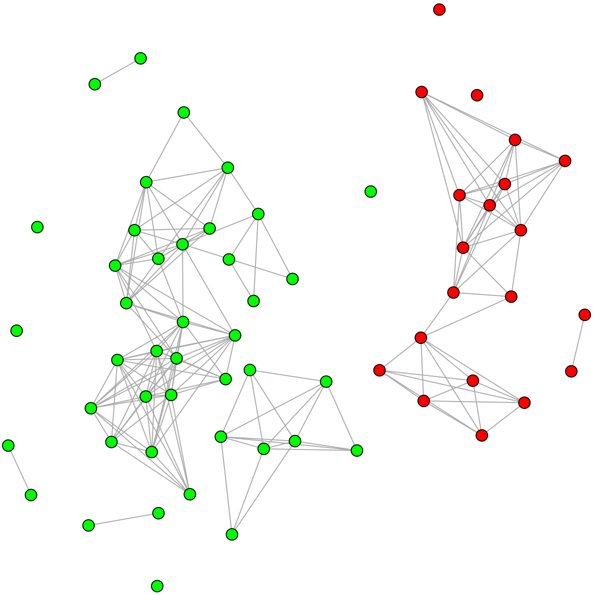


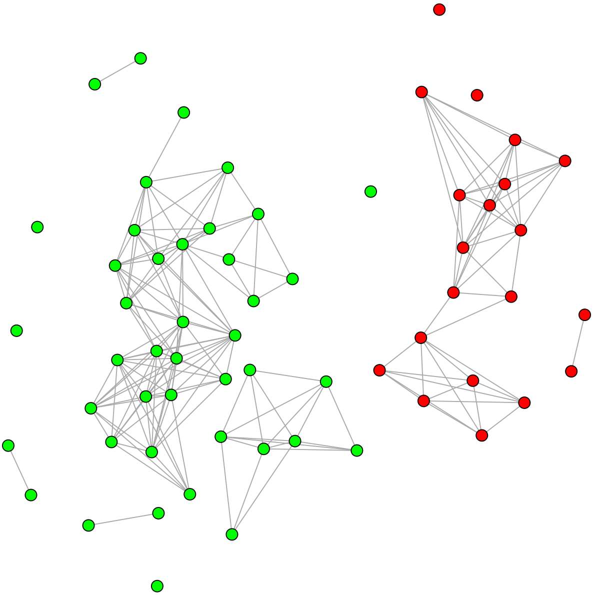


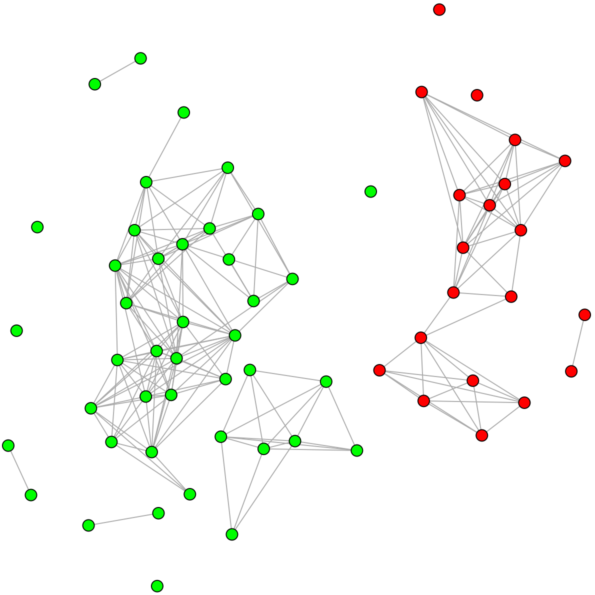


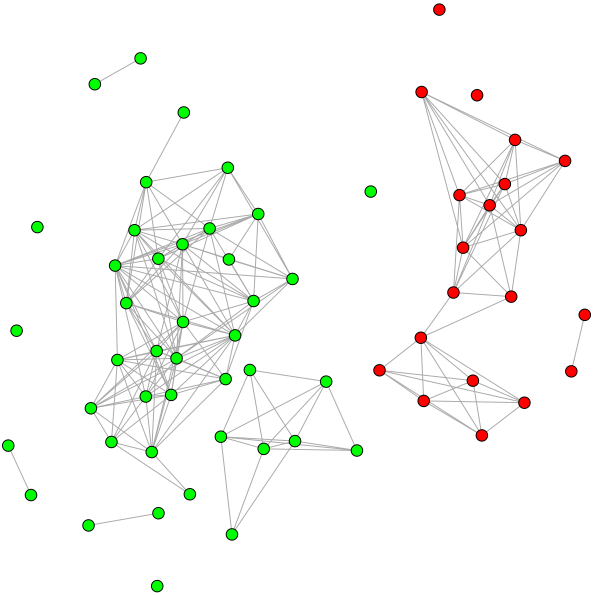


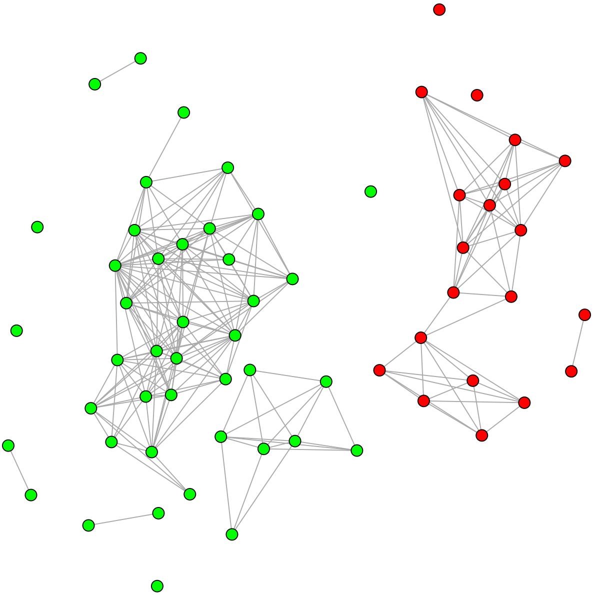


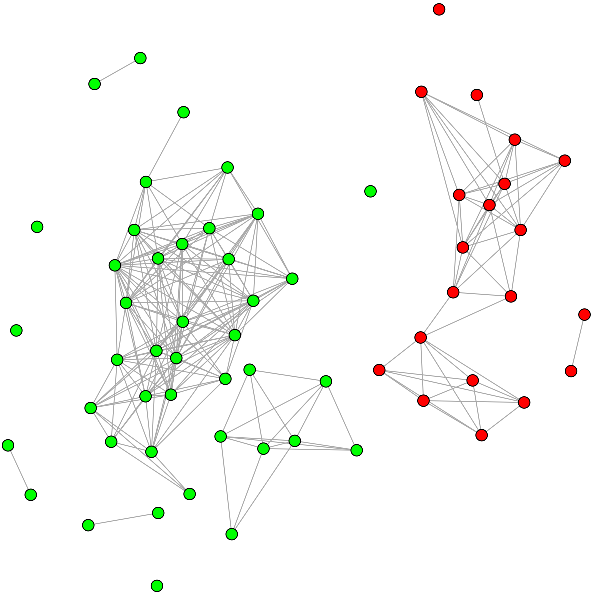


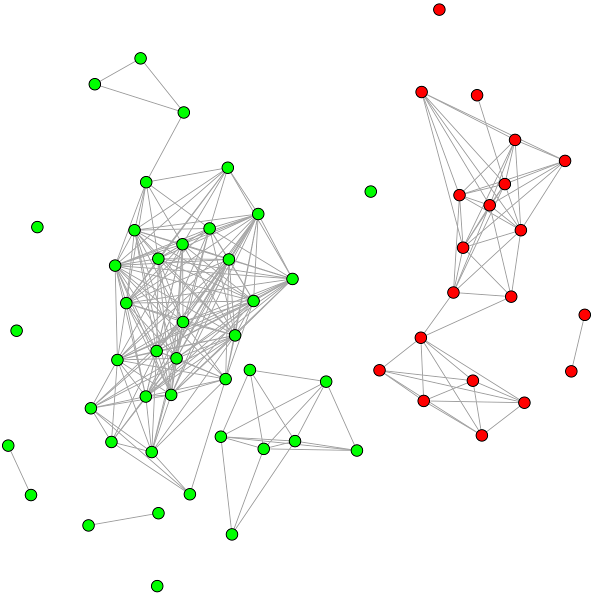


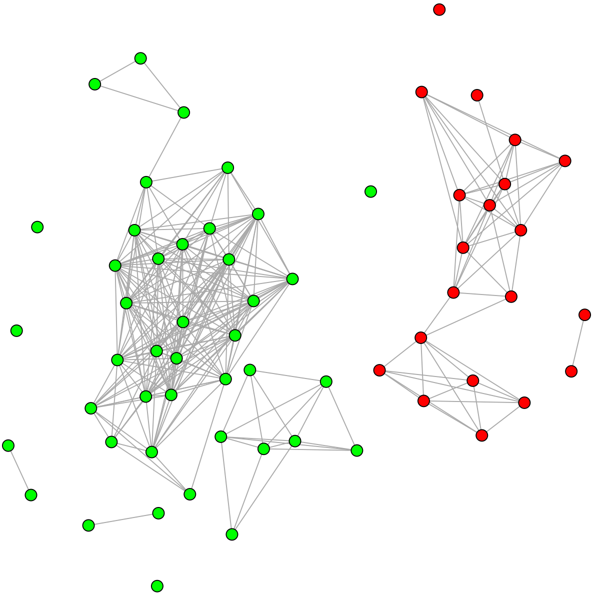


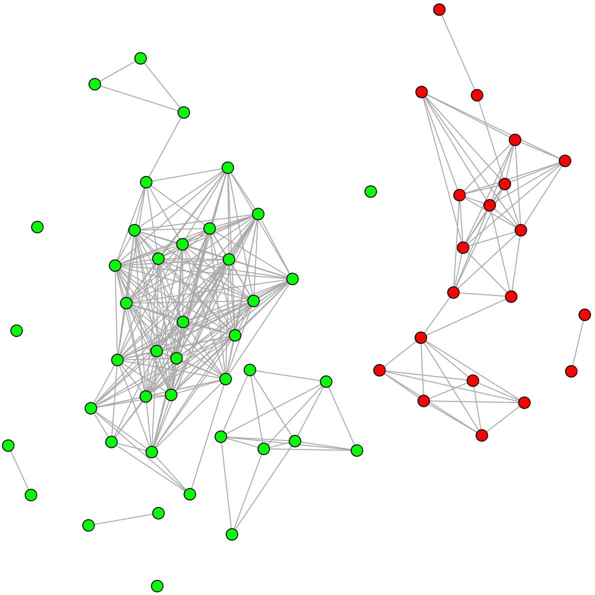


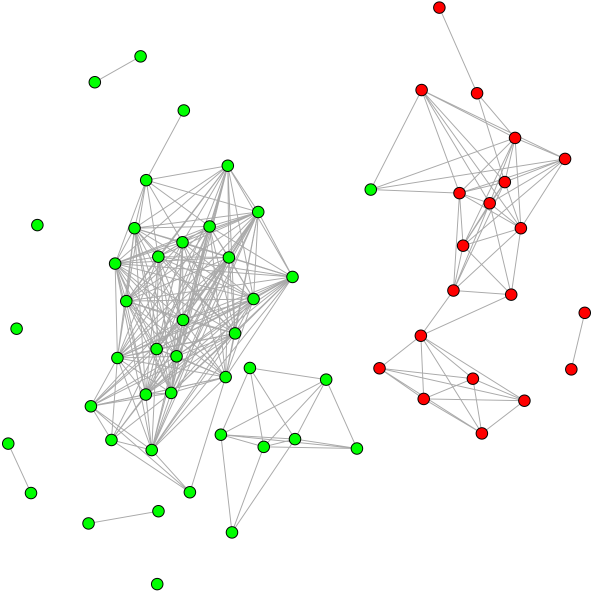


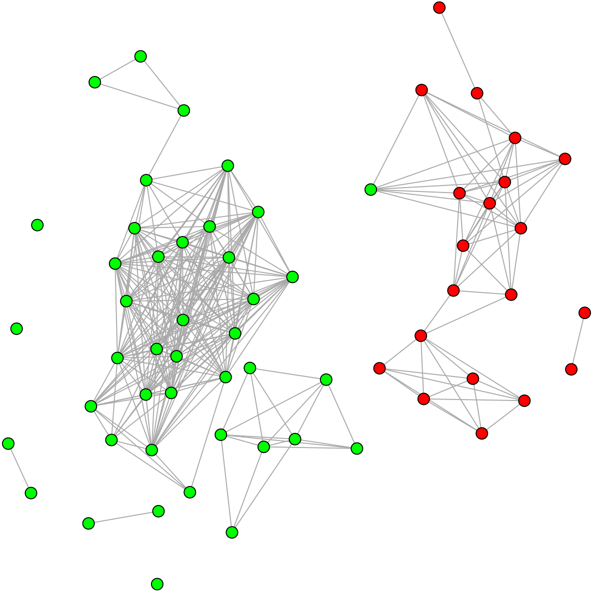


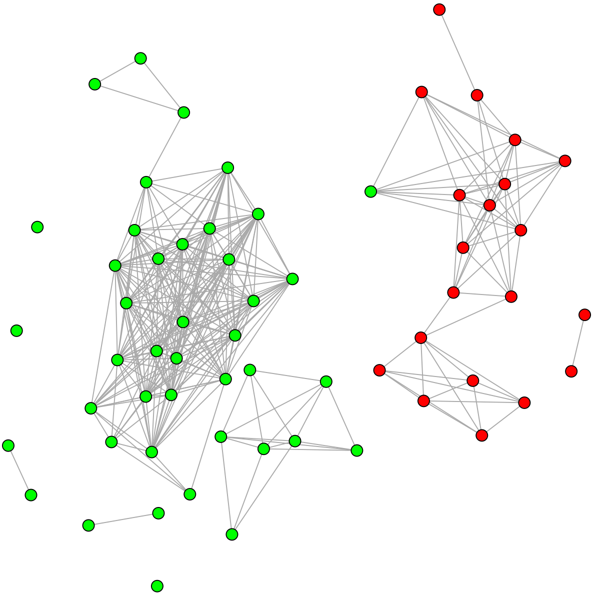


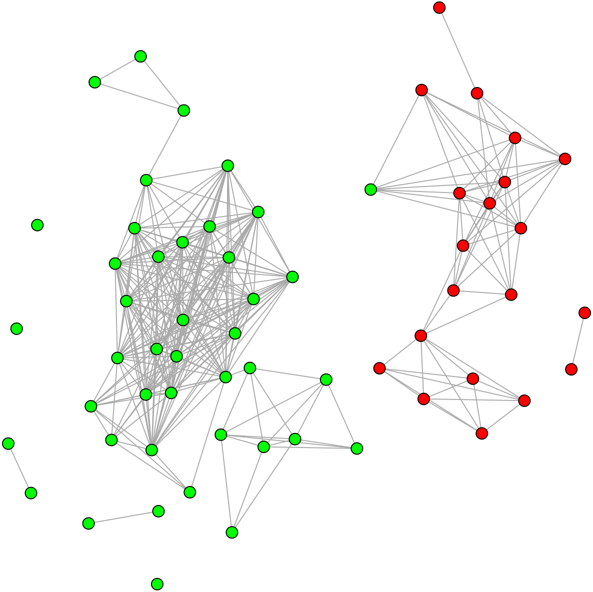


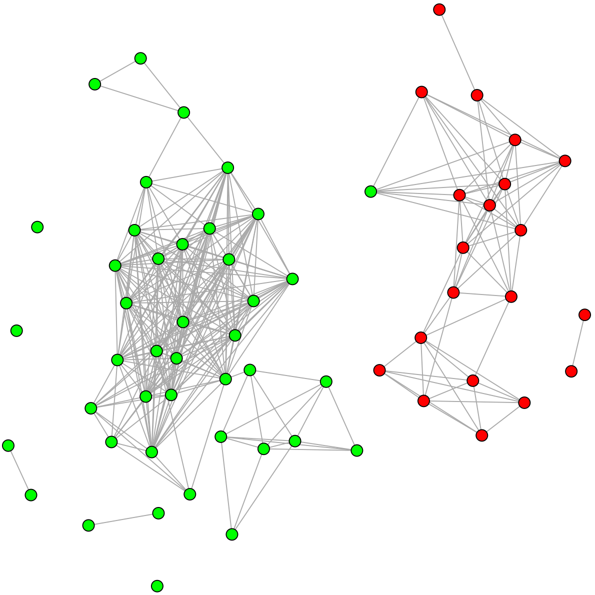


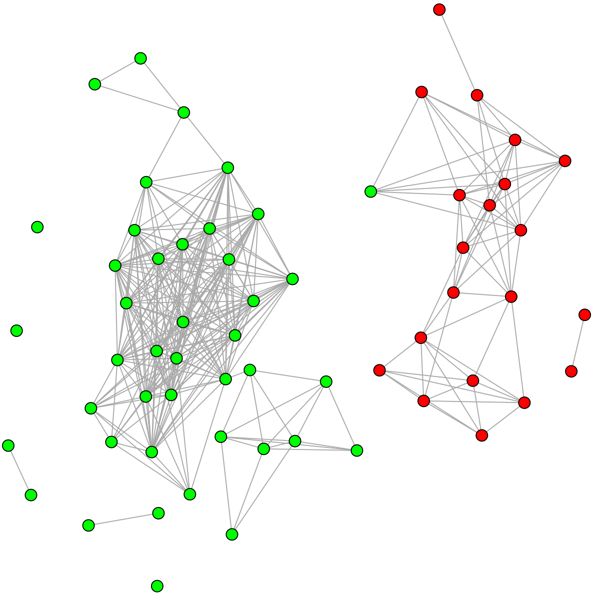


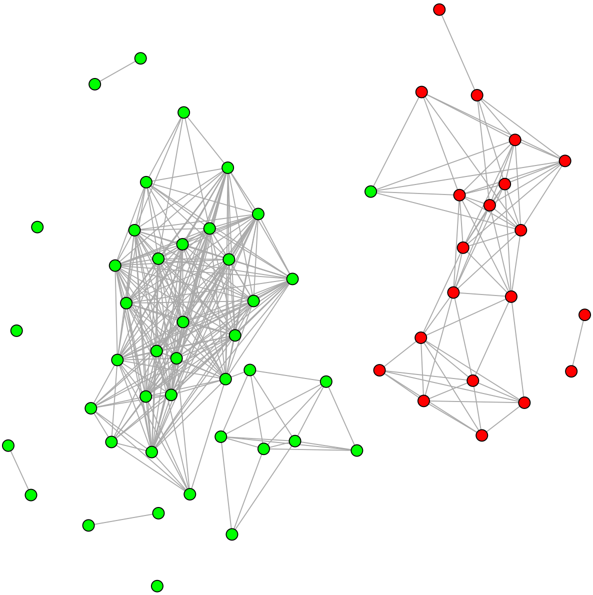


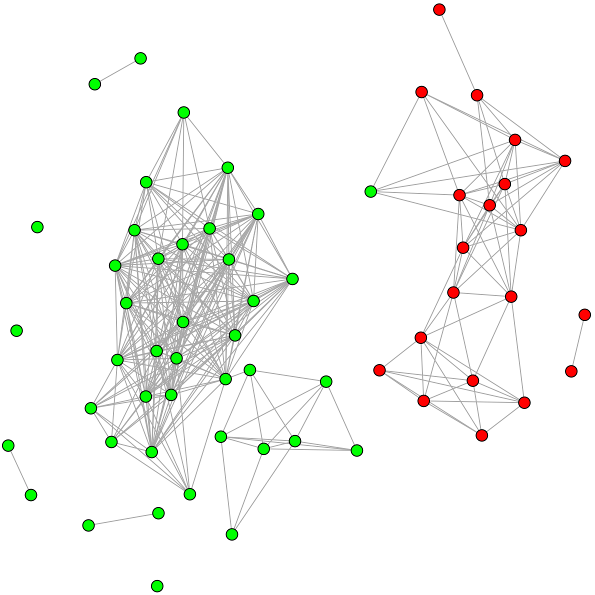


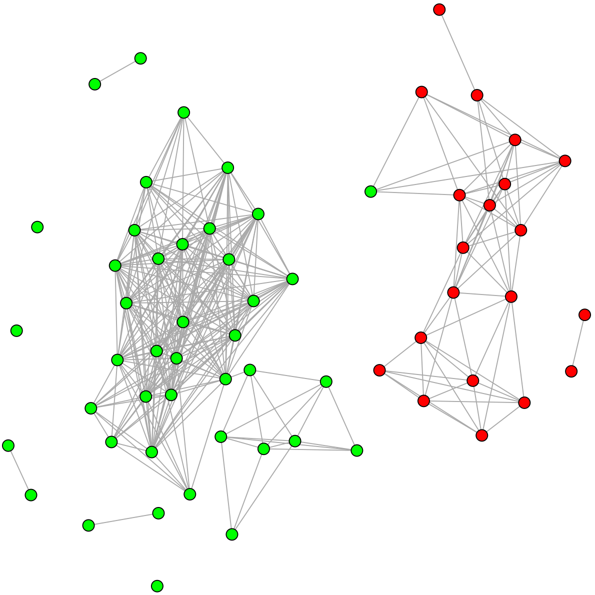


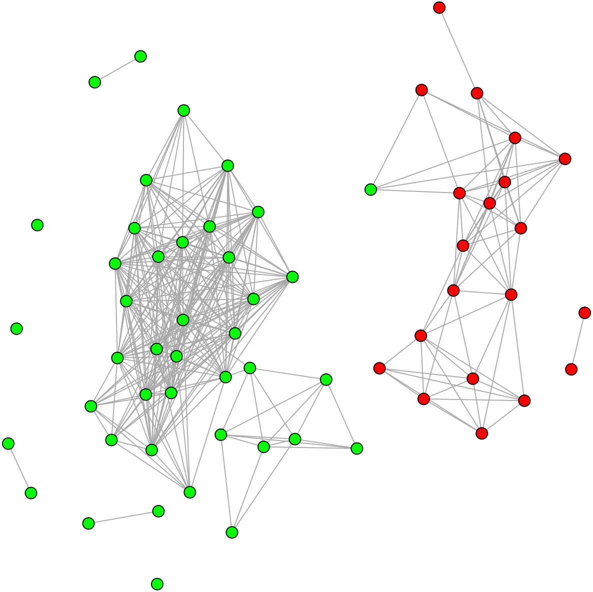


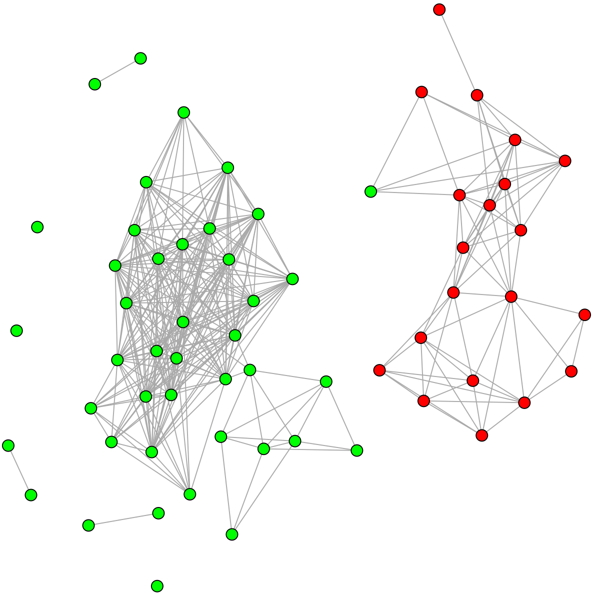


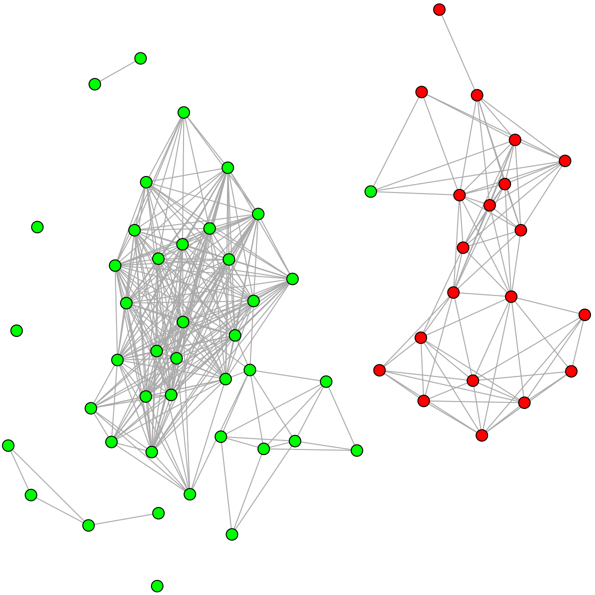


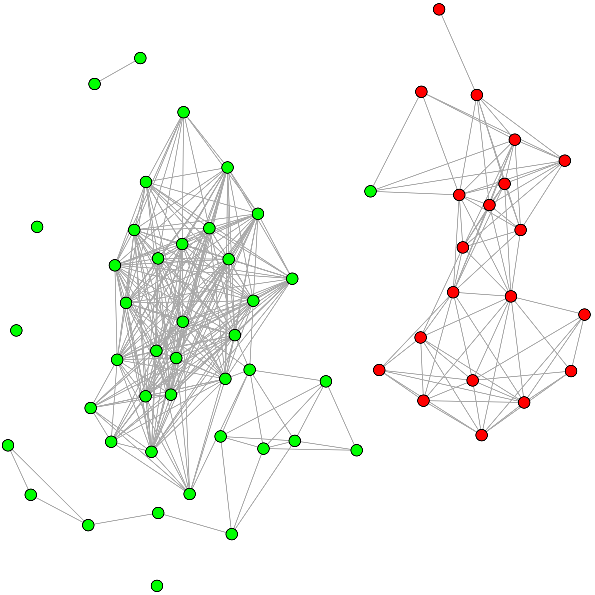


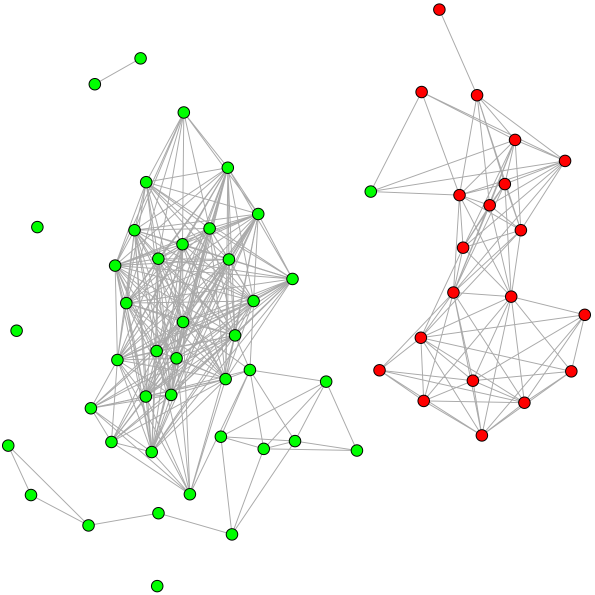


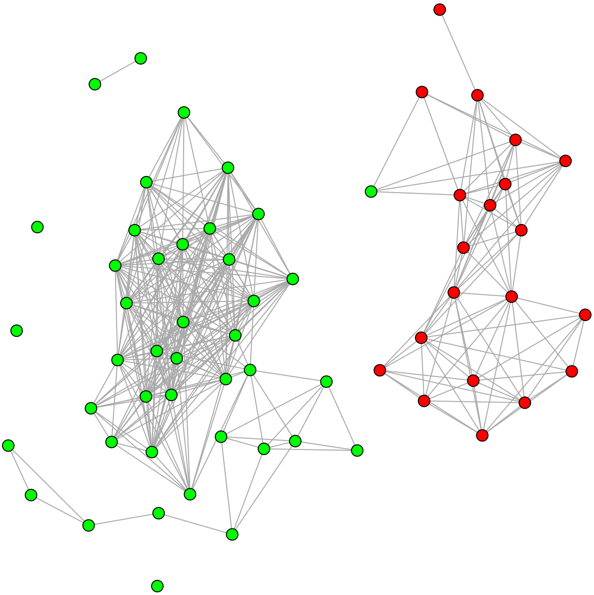


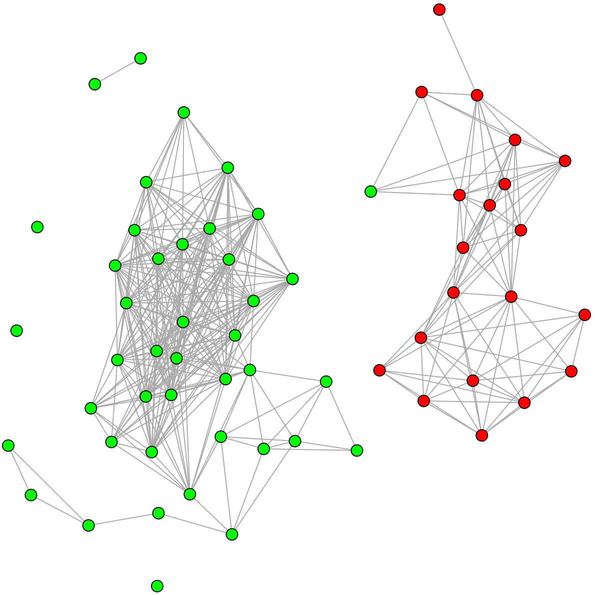


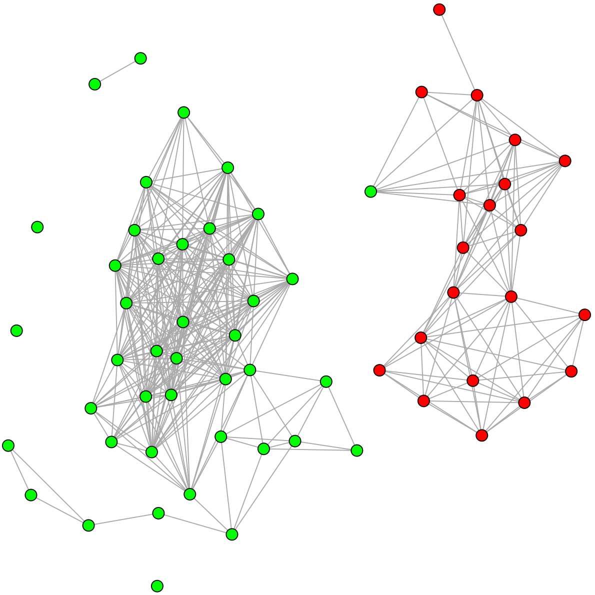


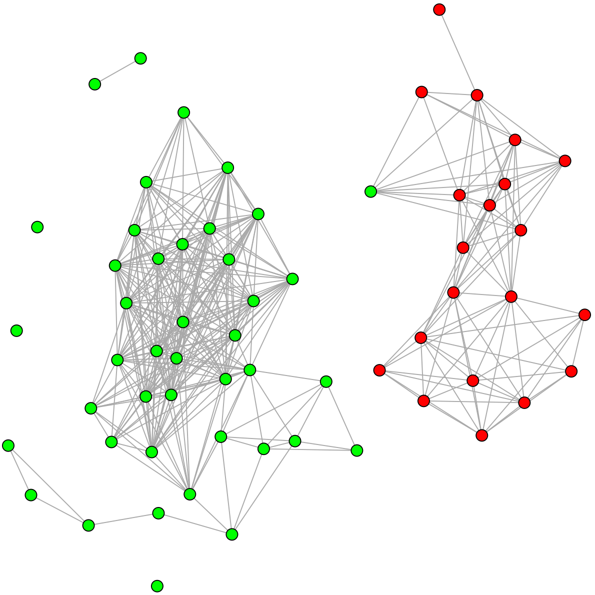


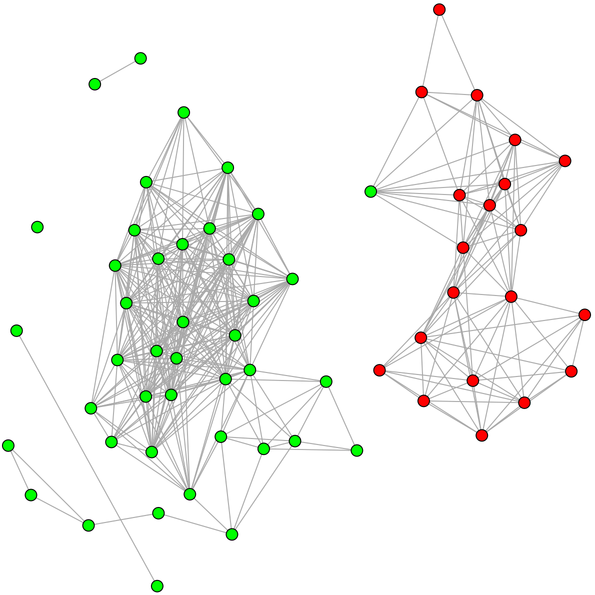


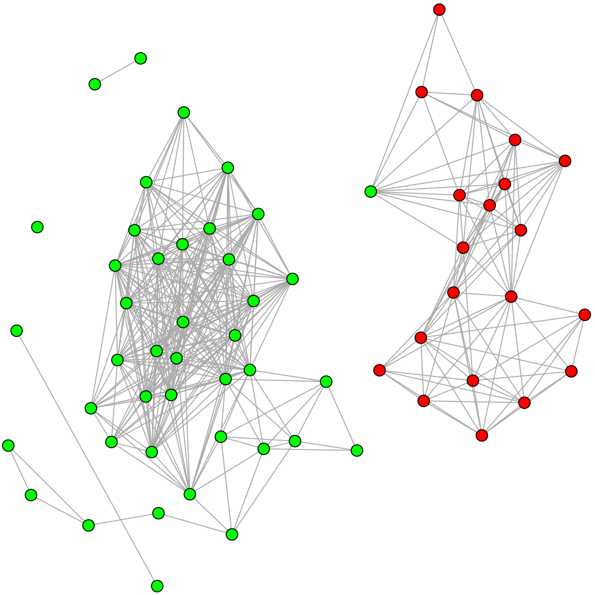


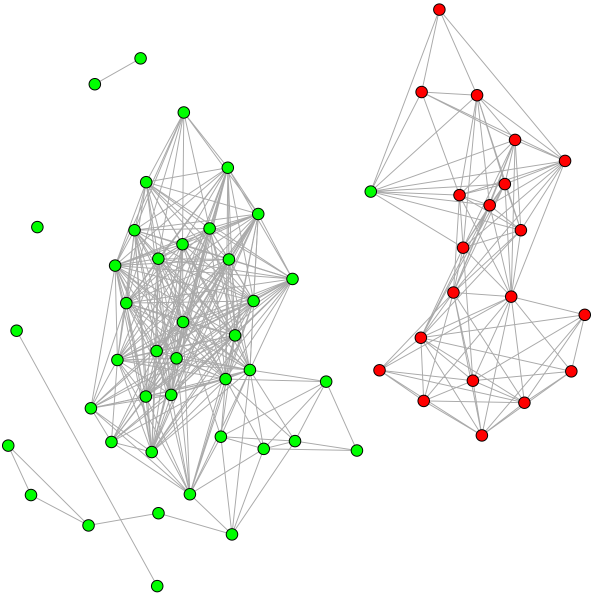


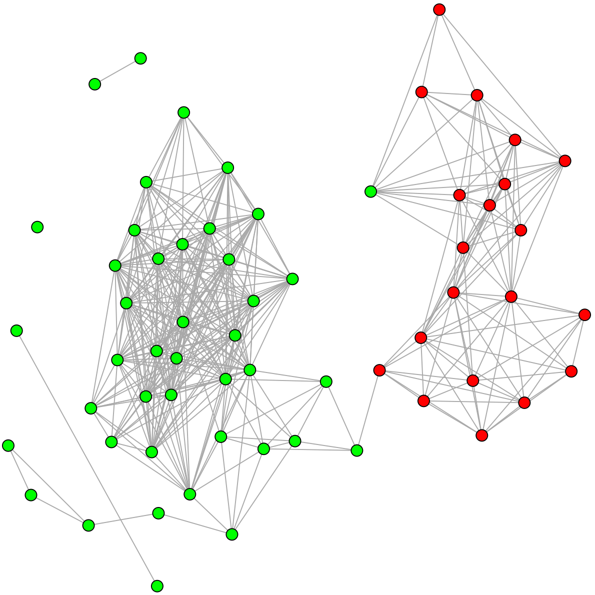


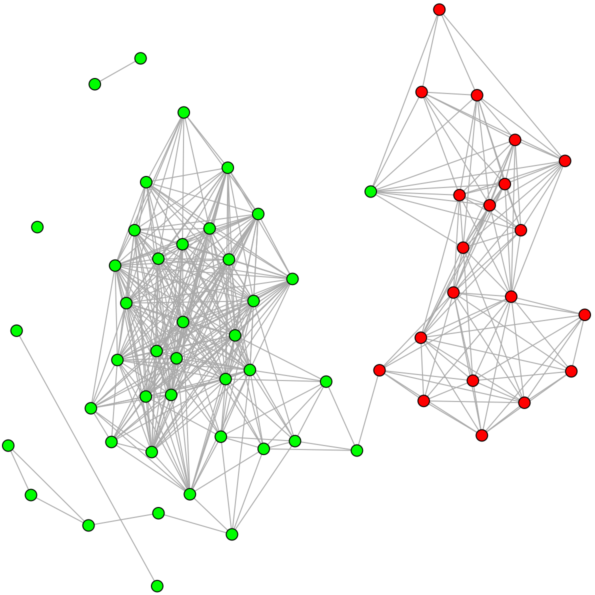


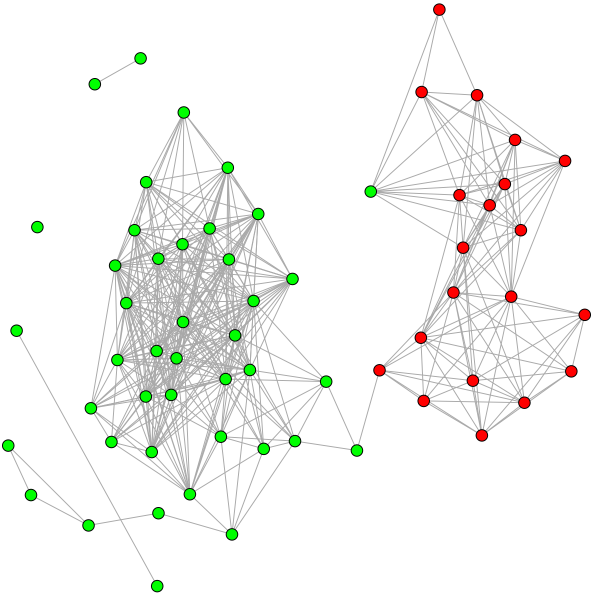


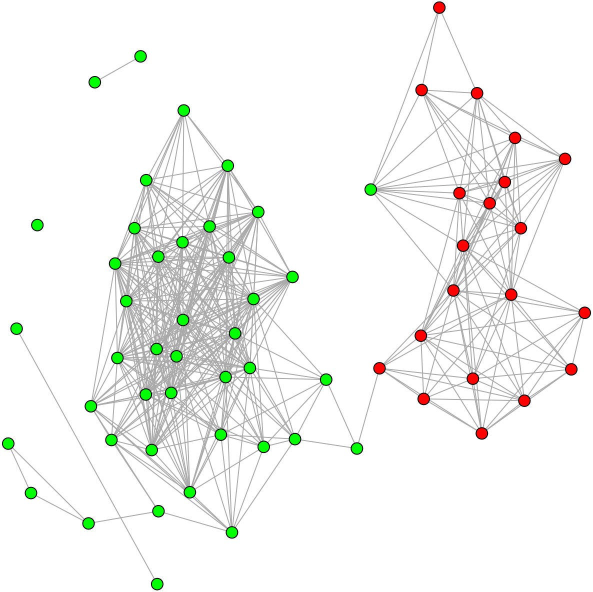


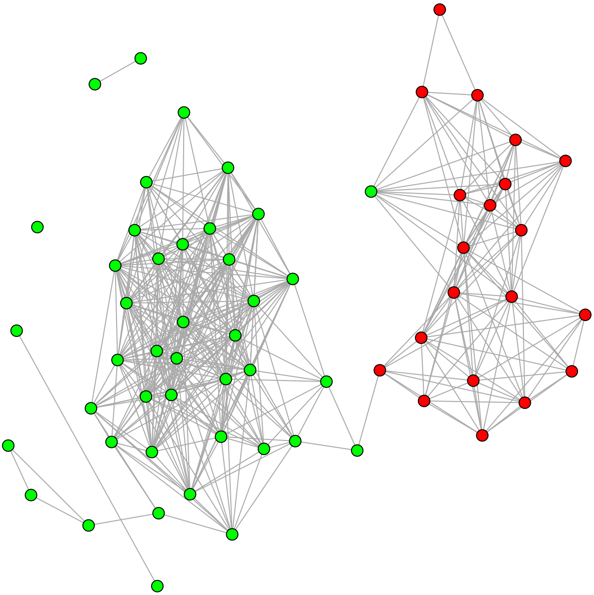


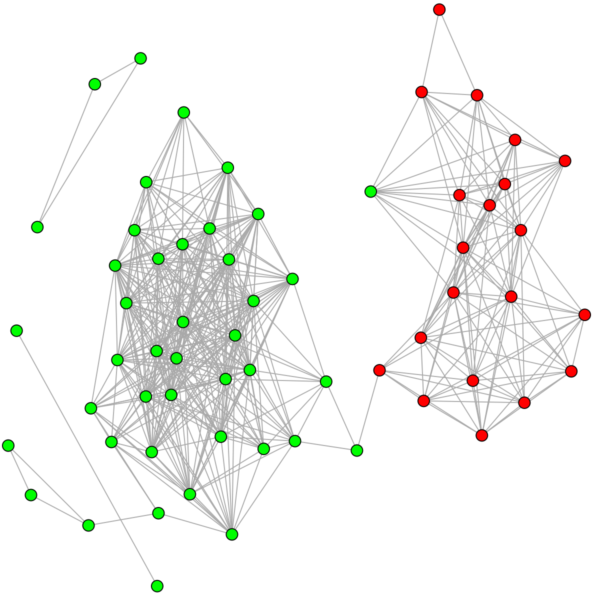


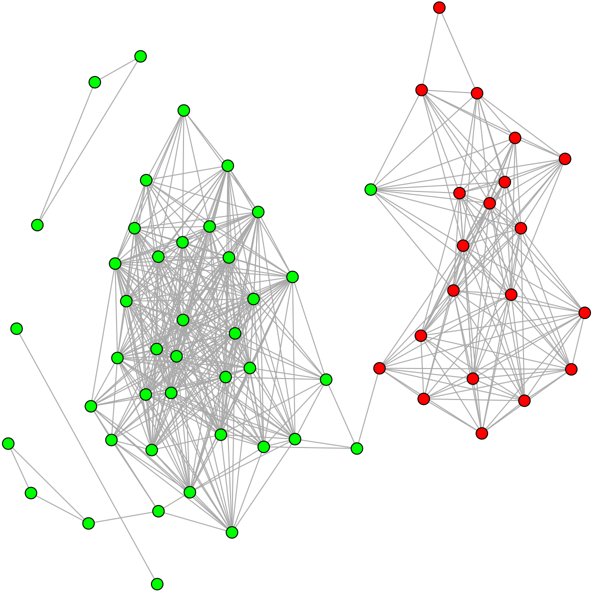


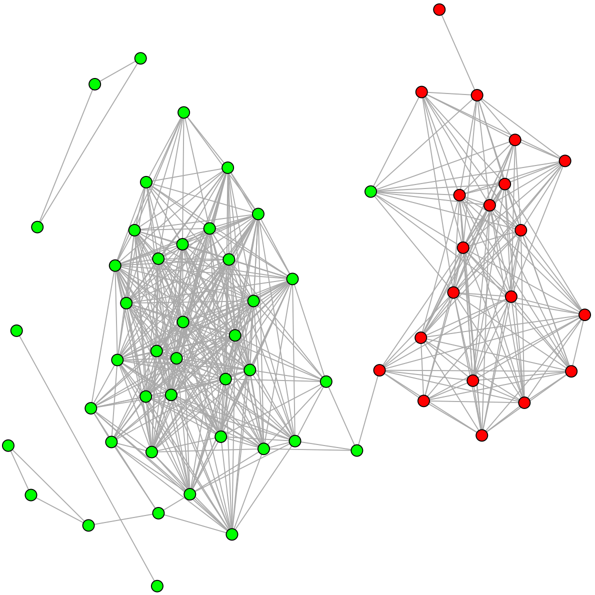


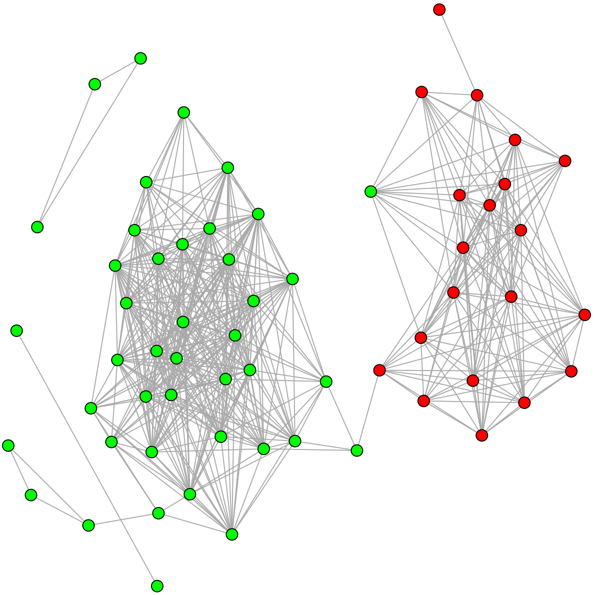


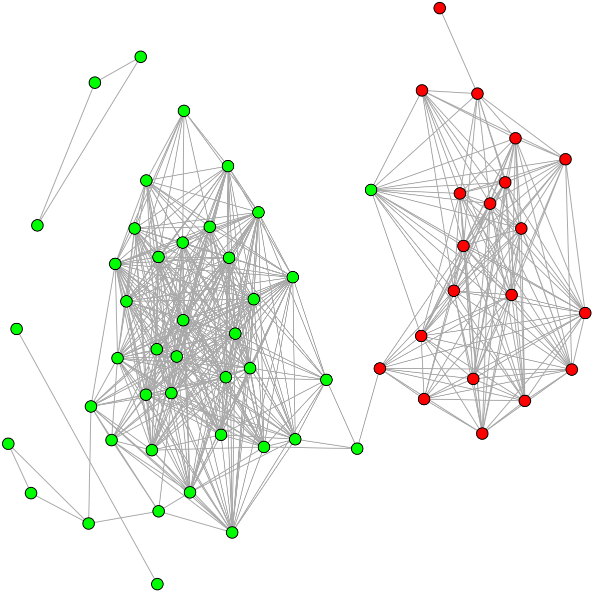


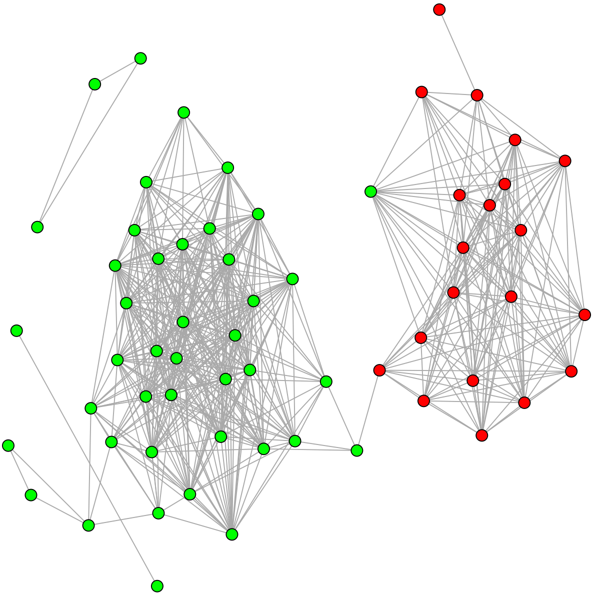


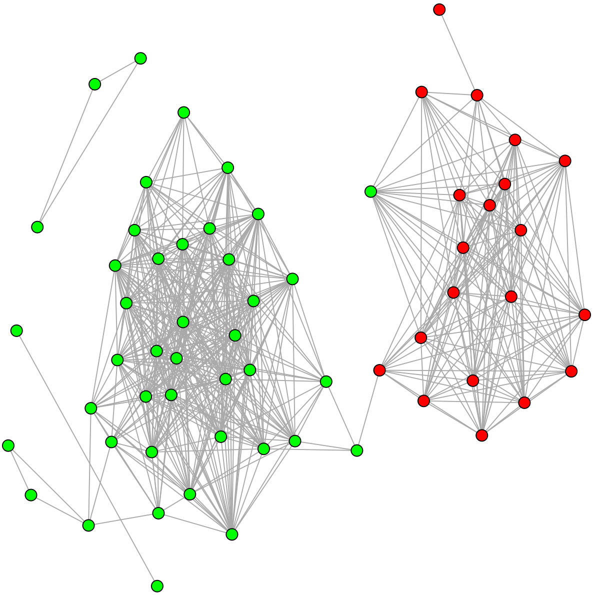


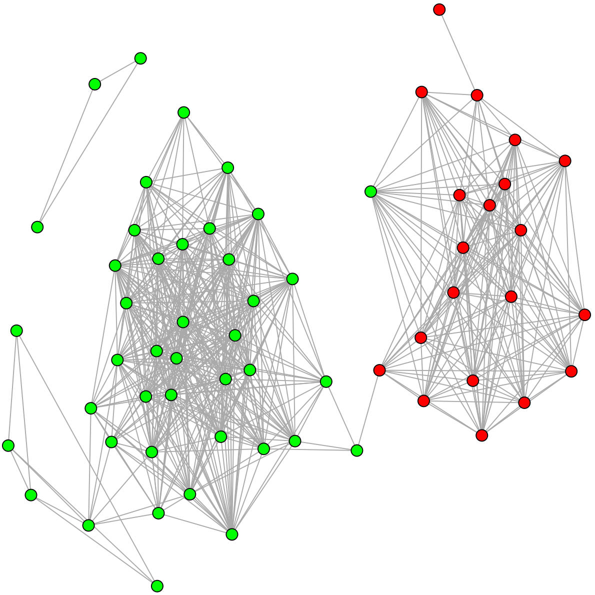


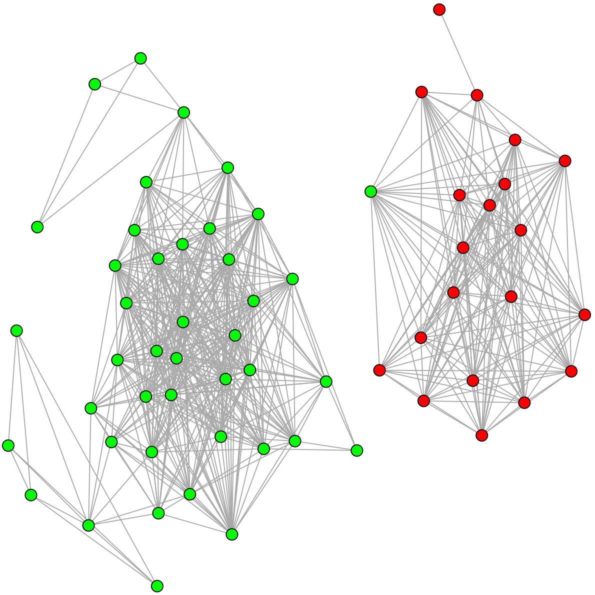


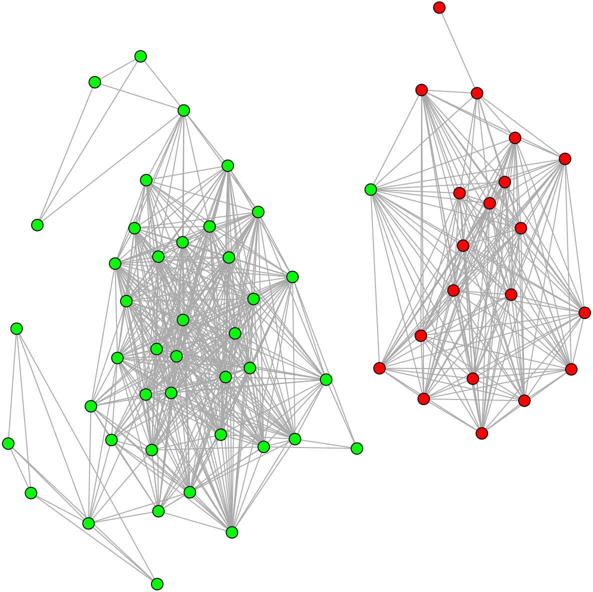


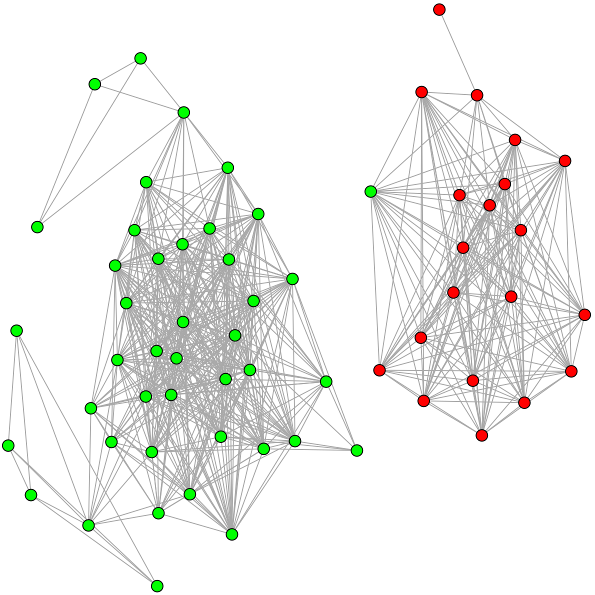


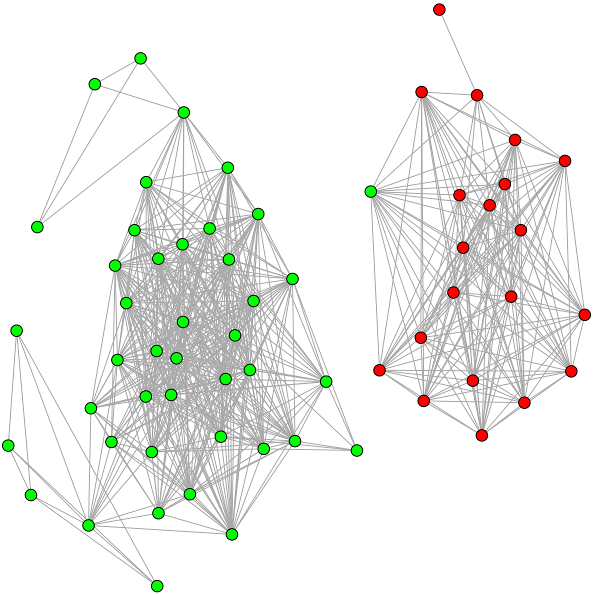


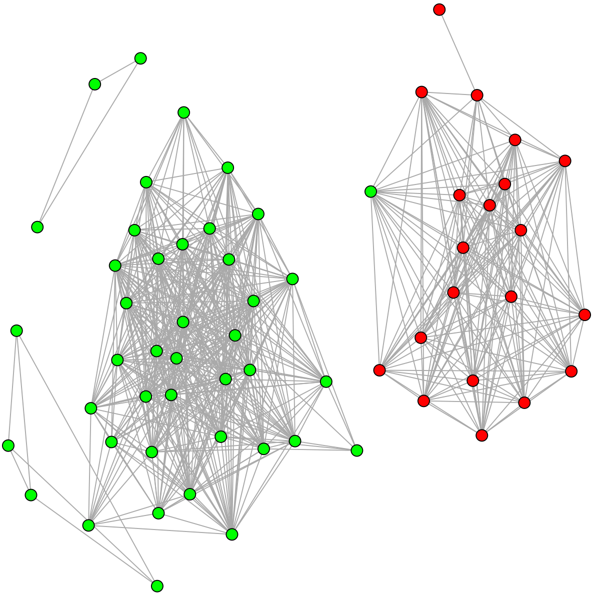


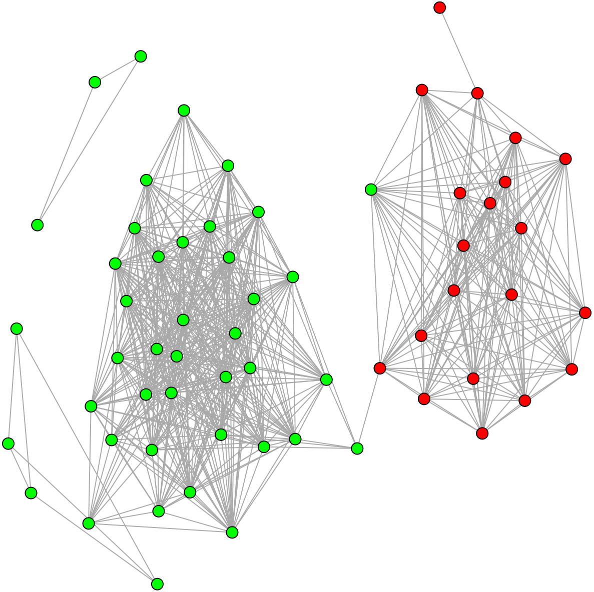


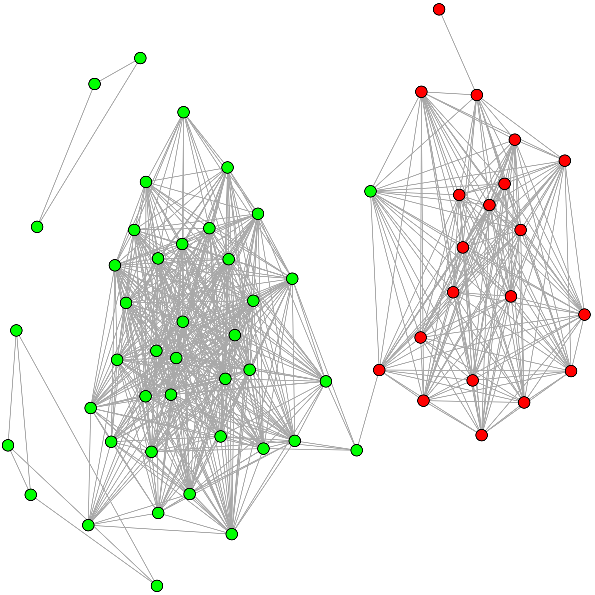


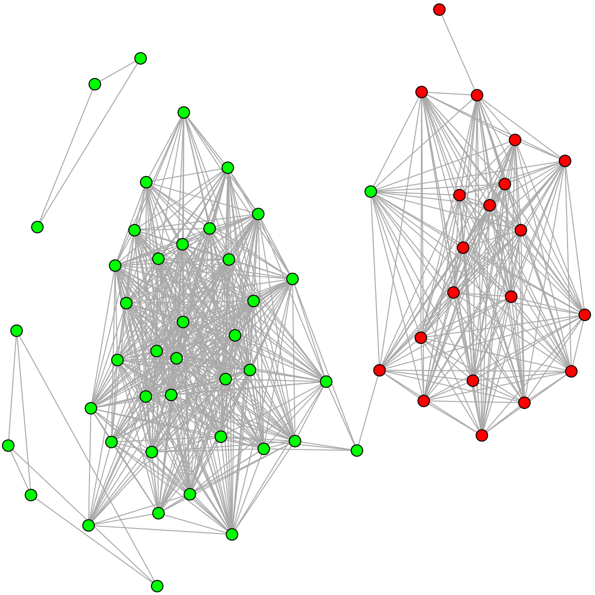


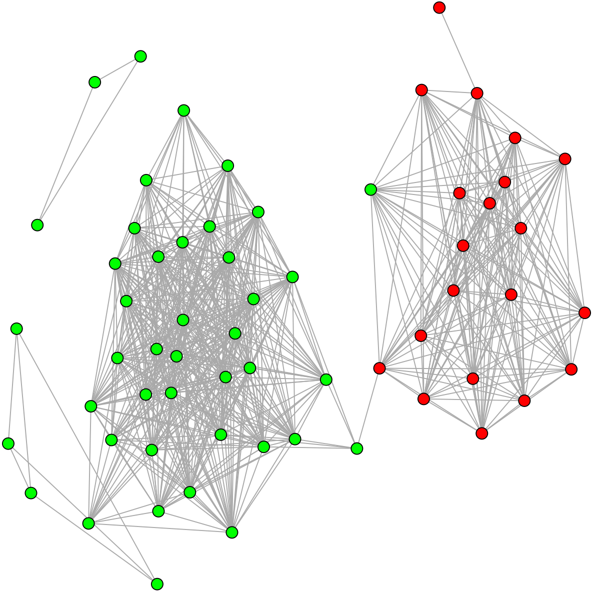


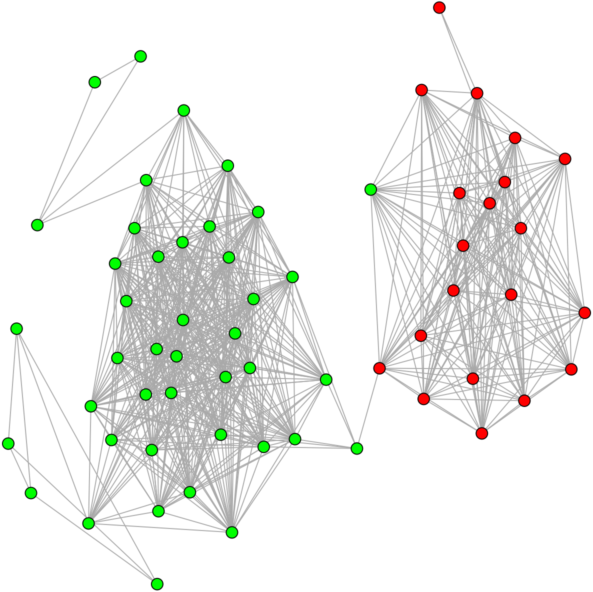


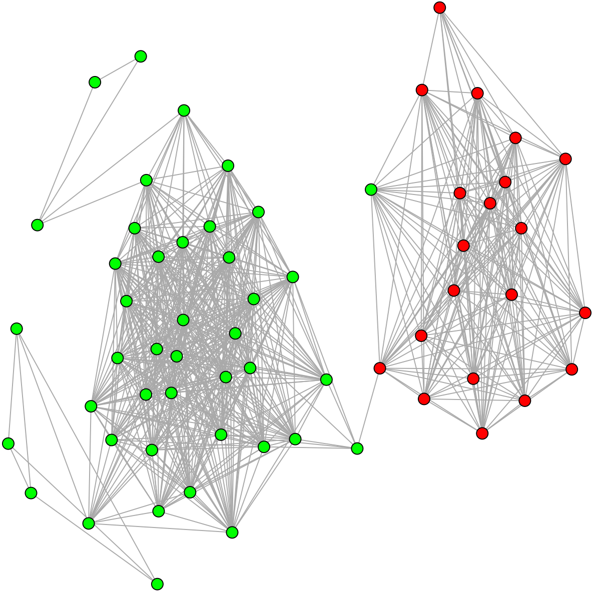


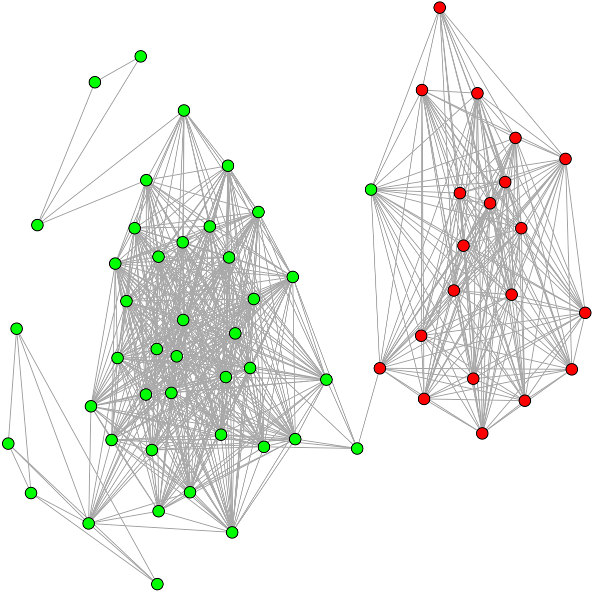


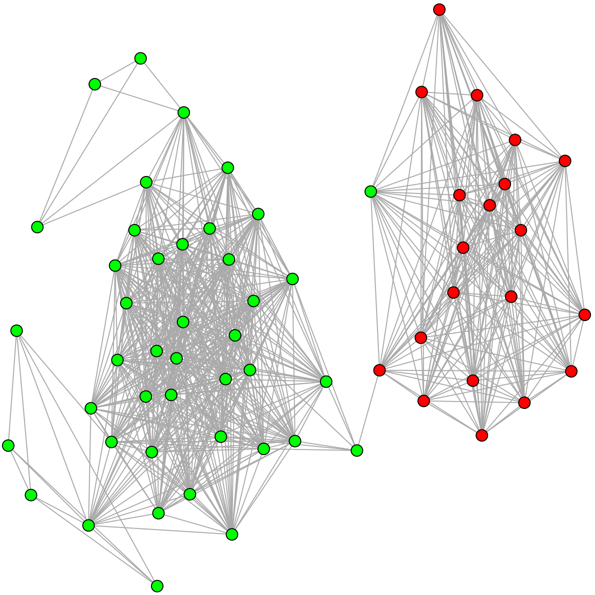


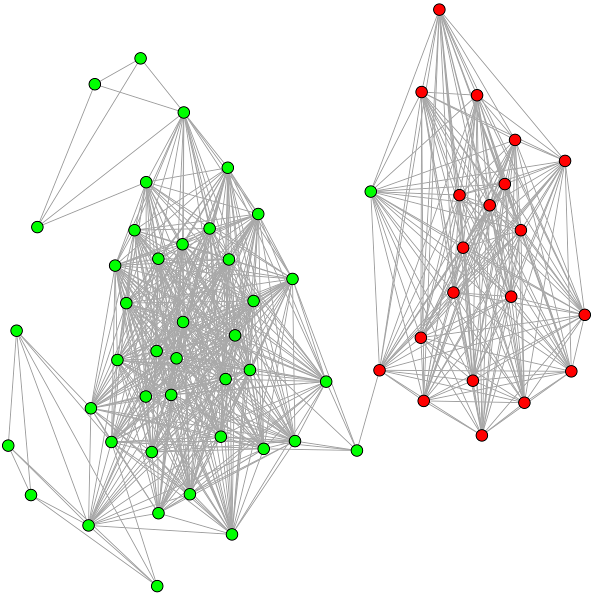


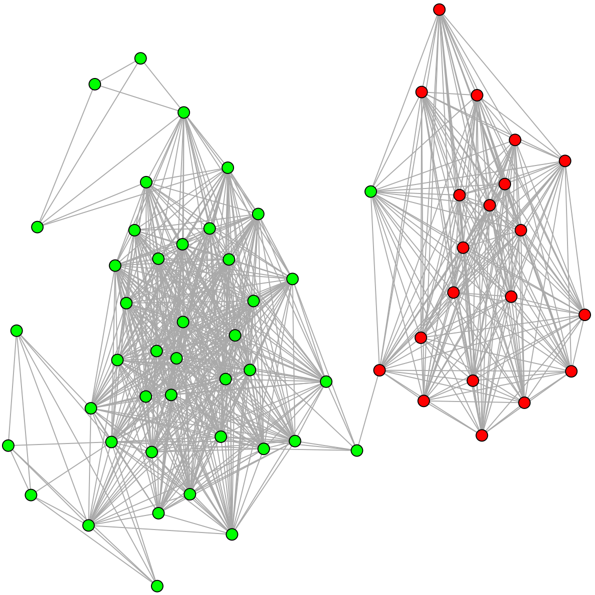


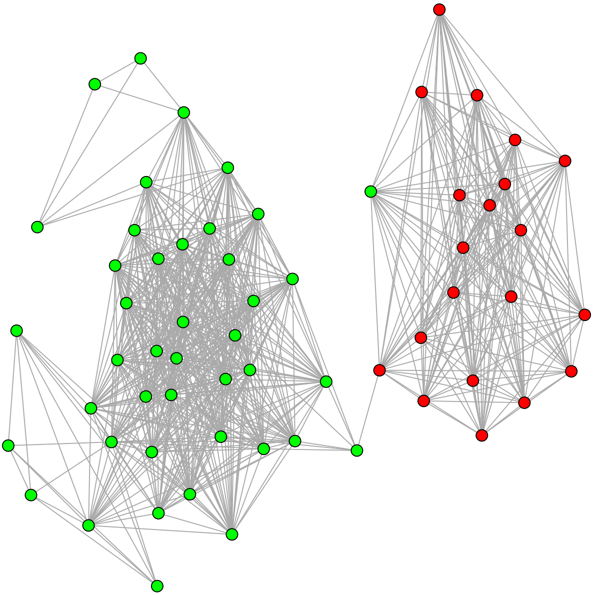


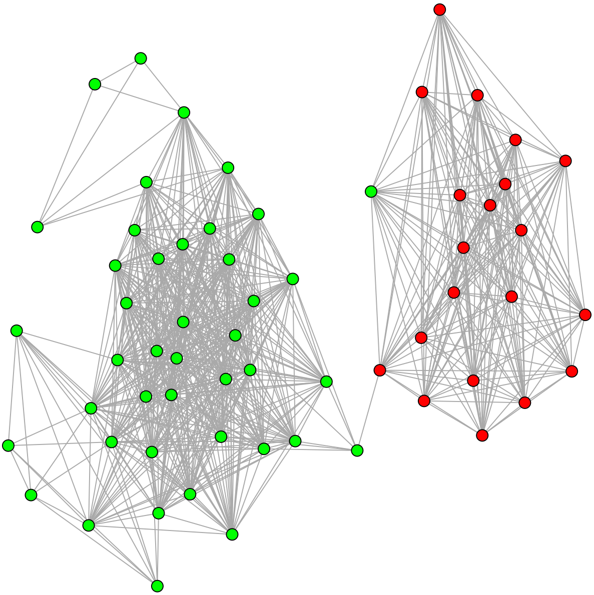


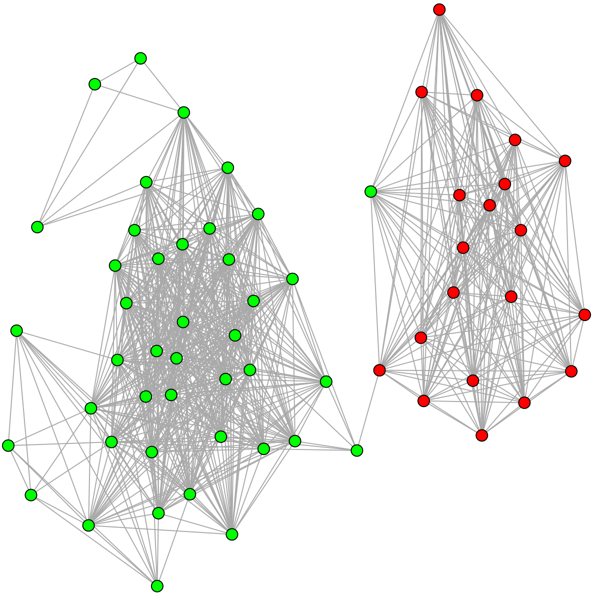






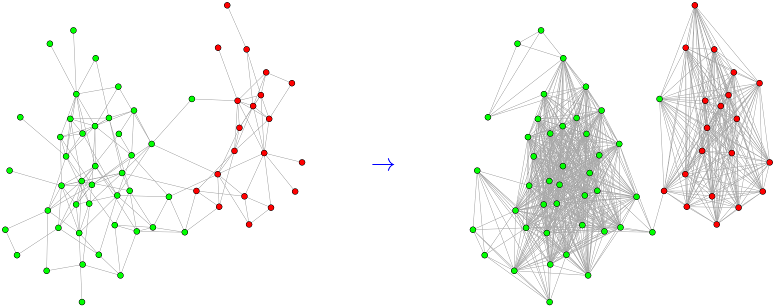






Performance of semidefinite relaxation

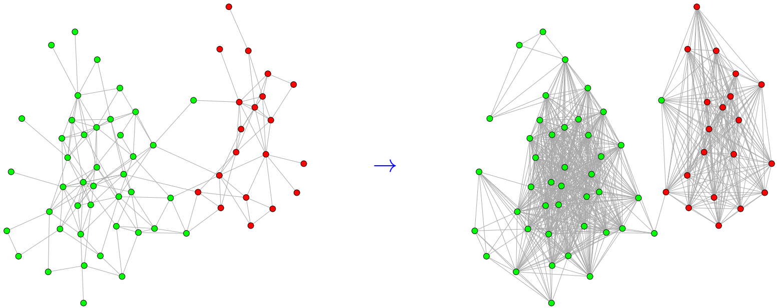
SDP enhances the latent structure of the network:



SDP *densifies* communities, *sparsifies* cuts across communities.

Performance of semidefinite relaxation

SDP enhances the latent structure of the network:



SDP *densifies* communities, *sparsifies* cuts across communities.

SDP did not know the number of communities in advance.
It decided that 2 communities should fit best.

Compressed sensing vs. networks

Compressed sensing

Signal: vector, matrix
Structure: sparsity, low rank
Measurements: random linear, few
Outliers: permitted in robust PCA
Exact recovery; exact thresholds
Recent blowup (2004+)

Structure recovery in networks

Signal: network model (p_{ij})
Structure: low rank, ??? (open)
Measurements: 0/1 random, few
Outliers: permitted (high/low degree vertices)
Exact recovery; exact thresholds
Recent blowup (2012+)