Spectral clustering of graphs

Sushmita Roy

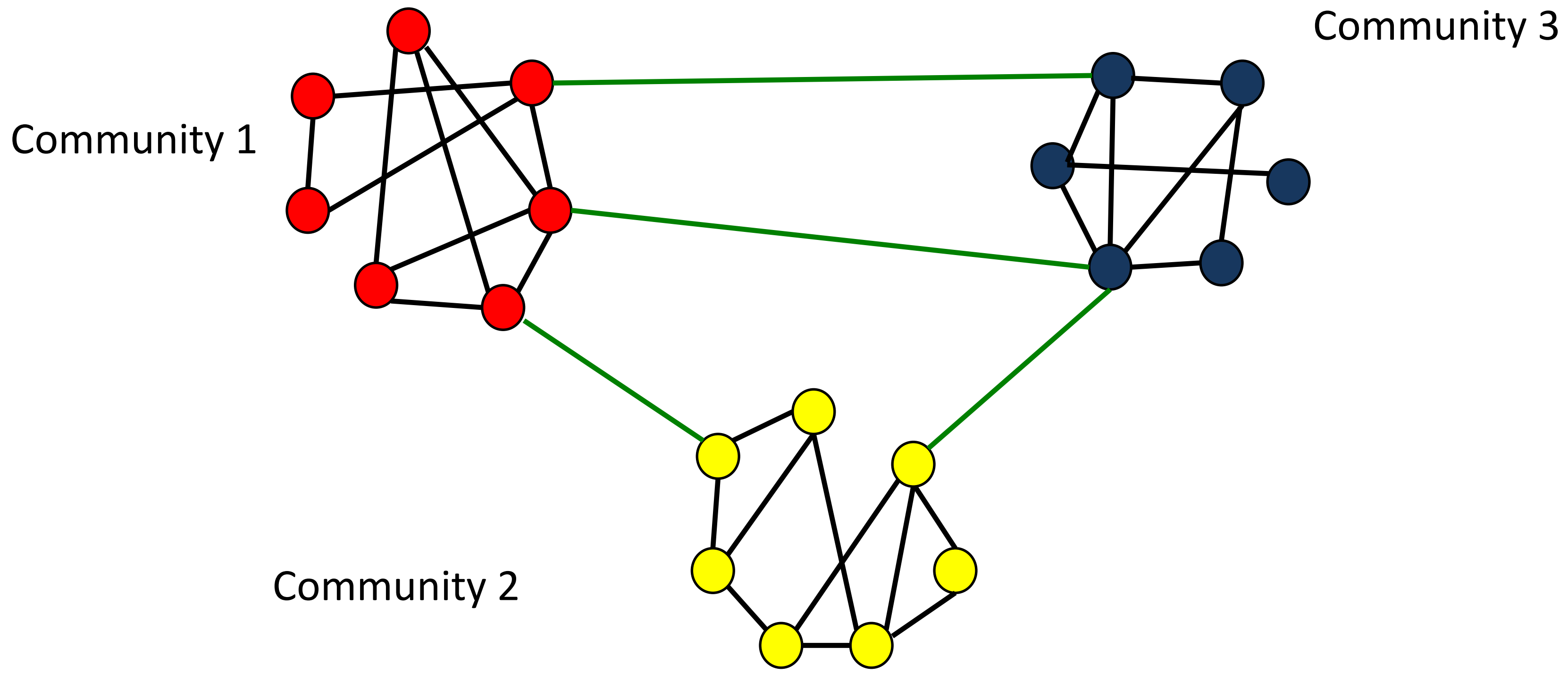
sroy@biostat.wisc.edu

Computational Network Biology
Biostatistics & Medical Informatics 826
https://compnetbiocourse.discovery.wisc.edu

Nov 6th 2018

RECAP: Graph clustering enables community structure detection

Graph clustering partitions vertices into groups based on their interaction patterns
Such partitions could be indicative of the specific properties of the vertices/form communities



RECAP: Common graph clustering algorithms

- Hierarchical or flat clustering using a notion of similarity between nodes
- Girvan-Newman algorithm
- Hierarchical Agglomerative clustering
- Spectral clustering
- Markov clustering algorithm
- Affinity propagation

Goals for this lecture

- A few graph-theoretic concepts
  - Graph Laplacian
  - Connected components from the Laplacian
- Spectral clustering
  - Spectral clustering and graph cut
  - Spectral clustering demo with Zachary Karate Club
- Application of spectral clustering

Notation

- Graph $G=\{V, E\}$ where V is the vertex set and E is the edge set
- D: Degree matrix of a graph
- W: Adjacency matrix of a graph
- L: Graph Laplacian
- x' denotes the transpose of x, where x is a vector
- A^{-1} denotes inverse of matrix A

A few linear algebra concepts

- Eigen vector of A
  - v, in n-dimensional (*nX1)* vector, is the eigen vector of A with eigen value λ, *(*a scalar*)* if

$$Av = \lambda v$$

- Positive semi-definite
  - A is said to be positive semi-definite if, for any n-dimensional vector x, the following holds

$$x' A x \geq 0$$

Eigen vector example

- Consider the matrix and vectors

$$A = \begin{pmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & -3 \end{pmatrix} \qquad u = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \qquad v = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

A*u

$$\begin{pmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & -3 \end{pmatrix} * \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \\ -3 \end{pmatrix}$$

A*v

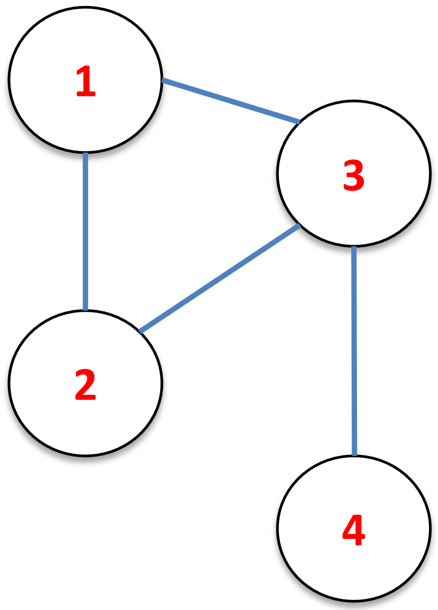
$$\begin{pmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & -3 \end{pmatrix} * \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \\ 0 \end{pmatrix}$$

Unnormalized Graph Laplacian

- For a given graph $G=\{V, E\}$
- The unnormalized graph Laplacian is a $|V|$X$|V|$ matrix

$$L = D - W$$

Unnormalized Graph Laplacian example



Example graph

Adjacency matrix(W)

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 1 & 1 & 0 \\ 2 & 1 & 0 & 1 & 0 \\ 3 & 1 & 1 & 0 & 1 \\ 4 & 0 & 0 & 1 & 0 \end{array}$$

Degree matrix (D)

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 2 & 0 & 0 & 0 \\ 2 & 0 & 2 & 0 & 0 \\ 3 & 0 & 0 & 3 & 0 \\ 4 & 0 & 0 & 0 & 1 \end{array}$$

Laplacian (L=D-W)

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 2 & -1 & -1 & 0 \\ 2 & -1 & 2 & -1 & 0 \\ 3 & -1 & -1 & 3 & -1 \\ 4 & 0 & 0 & -1 & 1 \end{array}$$

Properties of the Laplacian

- For every vector f in R^n,

$$f'Lf = \frac{1}{2}\sum_{ij} w_{ij}(f_i - f_j)^2$$

- L is symmetric and positive semi-definite

$$f'Lf \geq 0, \forall f \in R^n$$

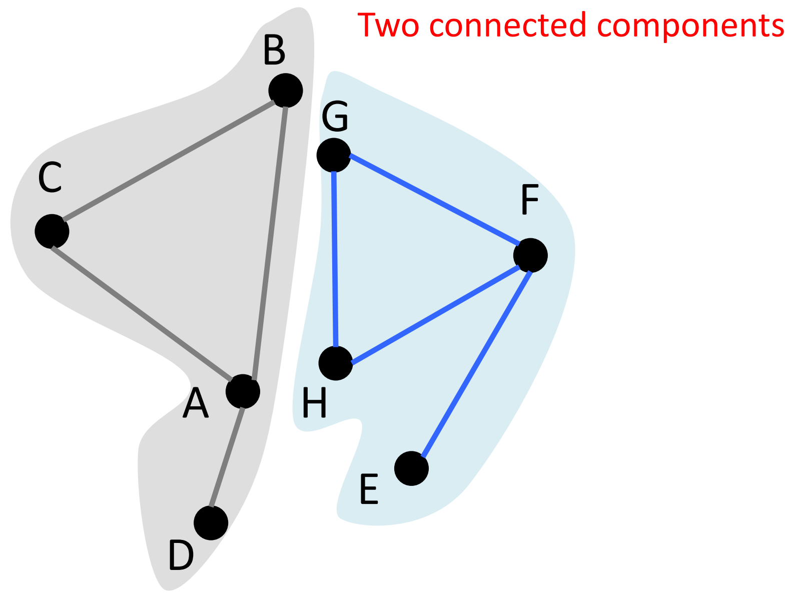
- The smallest eigen value of L is 0 and its corresponding eigen vector is all 1s

- L has n non-negative eigen values

$$0 = \lambda_1 \leq \lambda_2 \cdots \leq \lambda_n$$

Connected components

- A subgraph where there is path from one node to another
- The number of connected components is inherently tied to the Laplacian matrix



Connected component: A subgraph spanning a vertex subset where every vertex can be "reached" from another vertex

Number of connected components and the multiplicity of λ=0

- Let G be an undirected graph with non-negative weights.
- Then the multiplicity, k, of the eigenvalue 0 of L equals the number of connected components in the graph $A_1 , \dots , A_k$

Number of connected components and L's smallest eigen value

- To see why this is true, we use the property of an eigen vector, consider the case of one connected component
  - If f is an eigen vector of L, then $Lf=\lambda f$
  - For eigen value 0, $Lf=\mathbf{0}$ (vector or all zeros)
- In addition we know

$$f'Lf = \frac{1}{2}\sum_{i,j} w_{ij}(f_i - f_j)^2$$

- If f is an eigen vector corresponding to eigen value =0, this must be 0
- The only way this can be 0 is if $f_i=f_j$ because w_{ij} is non-zero
- This holds for all vertices connected by a path
- If all vertices are connected, then f is a vector of constants

Now consider a graph with k components

$$\begin{bmatrix} W_1 & & & \\ & W_2 & & \\ & & \ddots & \\ & & & W_k \end{bmatrix} \begin{bmatrix} L_1 & & & \\ & L_2 & & \\ & & \ddots & \\ & & & L_k \end{bmatrix}$$

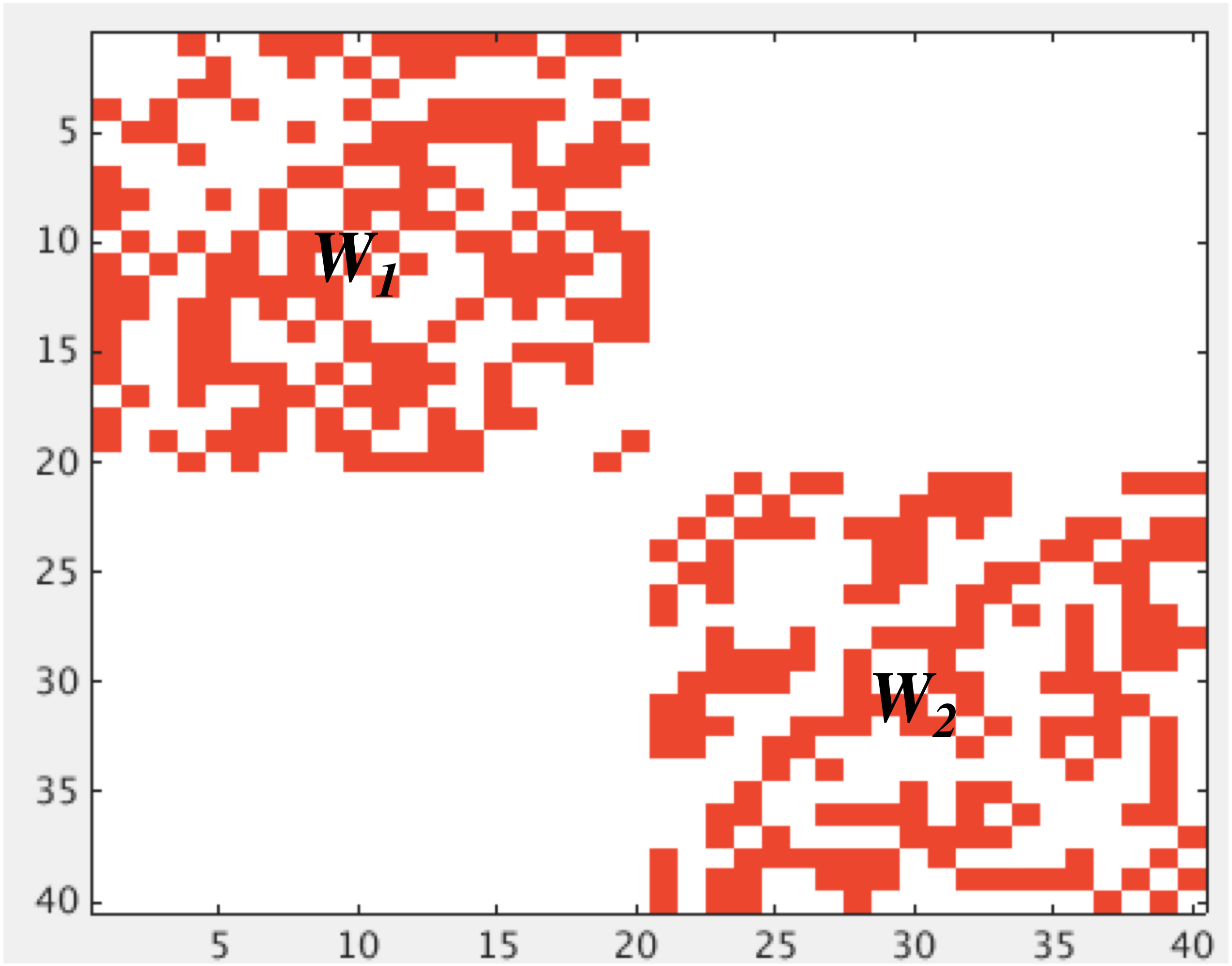
- W is block diagonal with *k* blocks
- L is also block diagonal

Consider a graph with k components

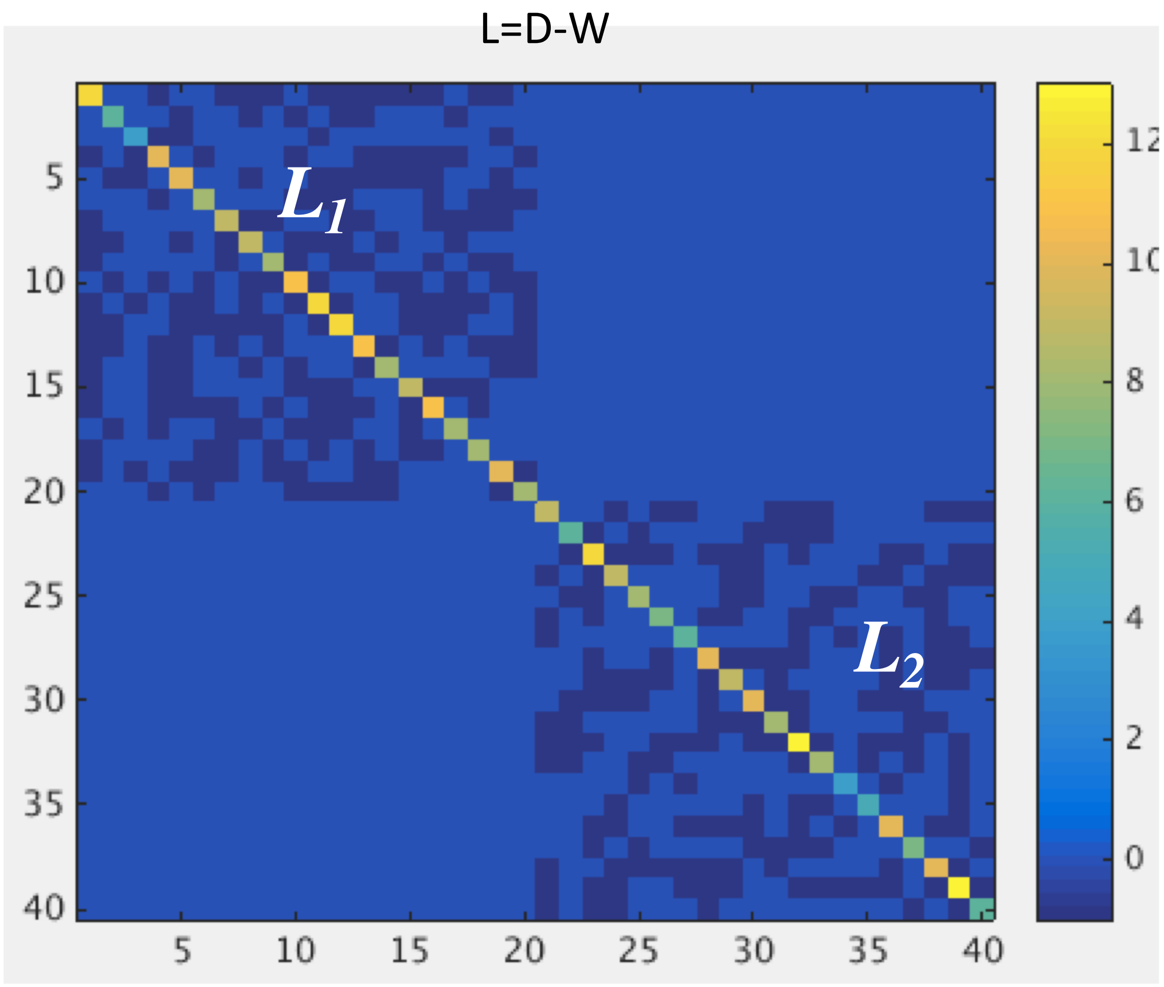
- Each L_i is a proper graph Laplacian for the subgraph of the i^{th} component.
- Each L_i has one eigen value of 0 and the corresponding eigen vector is constant one vector
- Eigen vectors of the full graph is the same as the eigen vectors of individual blocks with the remaining entries set to 0.
- Thus L must have k eigen values equal to 0.
- Each eigen vector of L is constant non-zeros for the entries corresponding to each connected component

An example with 2 connected components

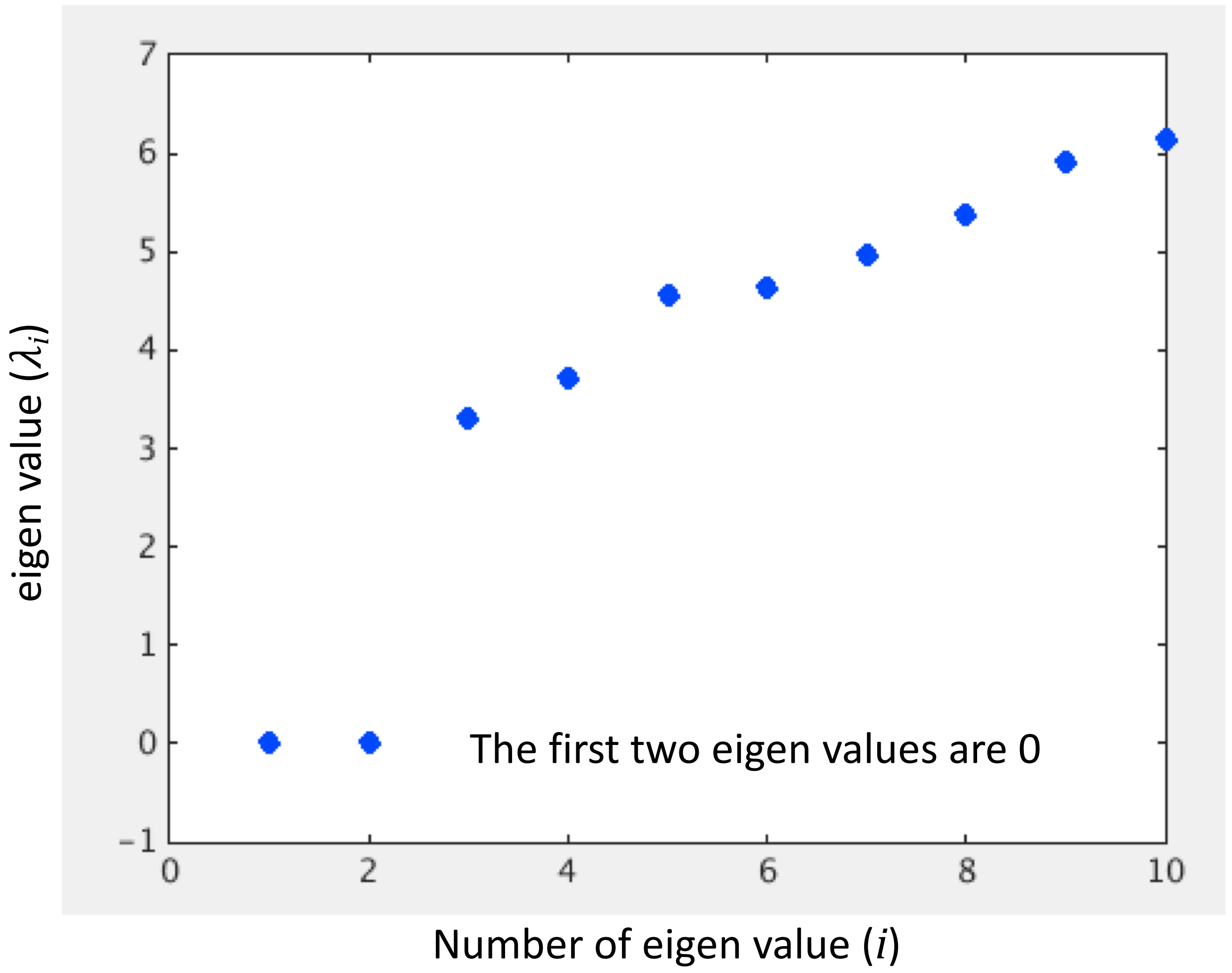
For this matrix, we expect to have two eigen vectors associated with eigen value =0



The corresponding Laplacian

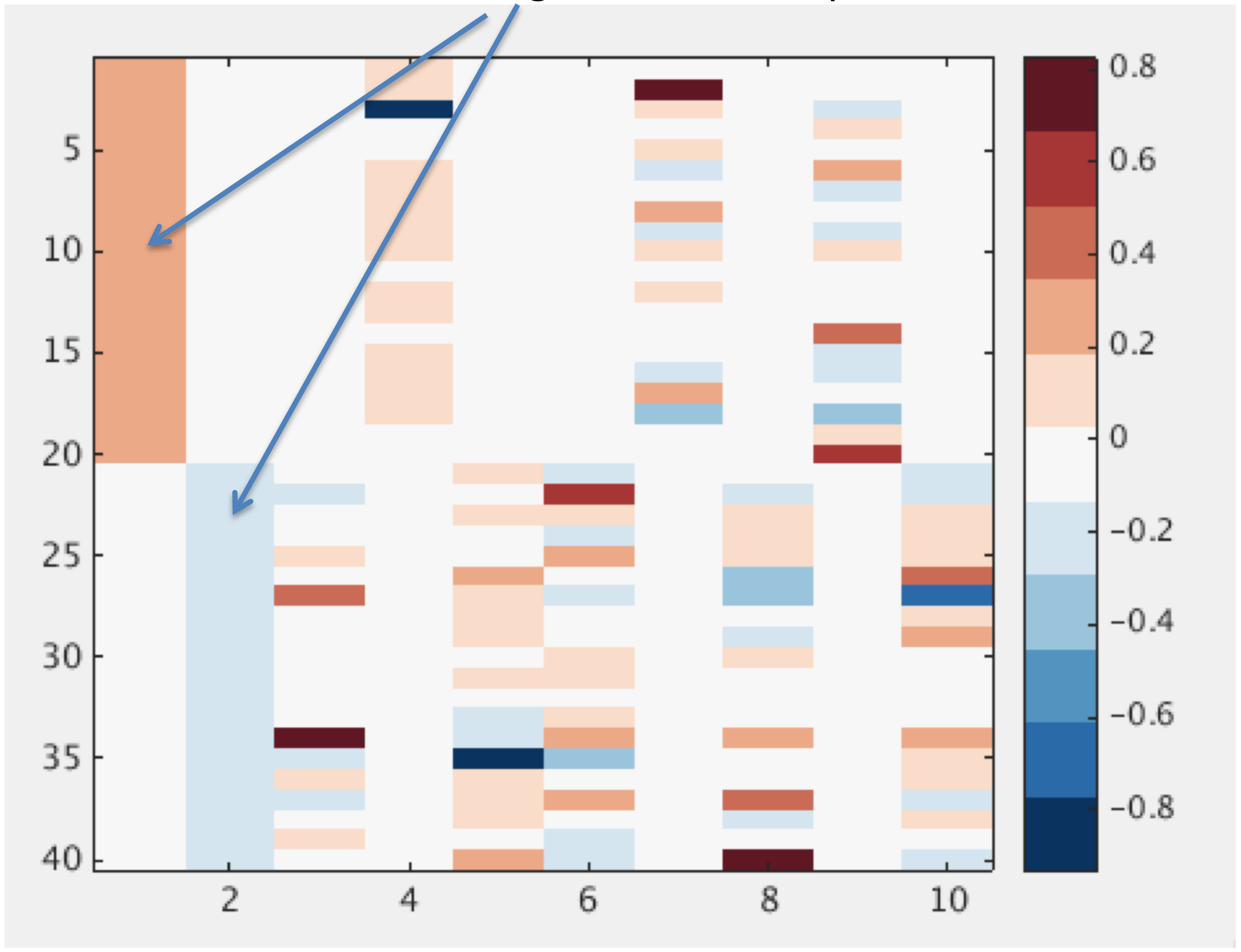


First 10 eigen values



First 10 eigen vectors

First two eigen vectors are piece-wise constant



Normalized graph Laplacians

- L_{sym}

$$L_{sym} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$$

- L_{rw}

$$L_{rw} = D^{-1} L = I - D^{-1} W$$

Graph Laplacians have a lot of applications

- Graph clustering
- Regularization in an objective function to find a solution that obeys the graph structure
- Diffusion on a graph

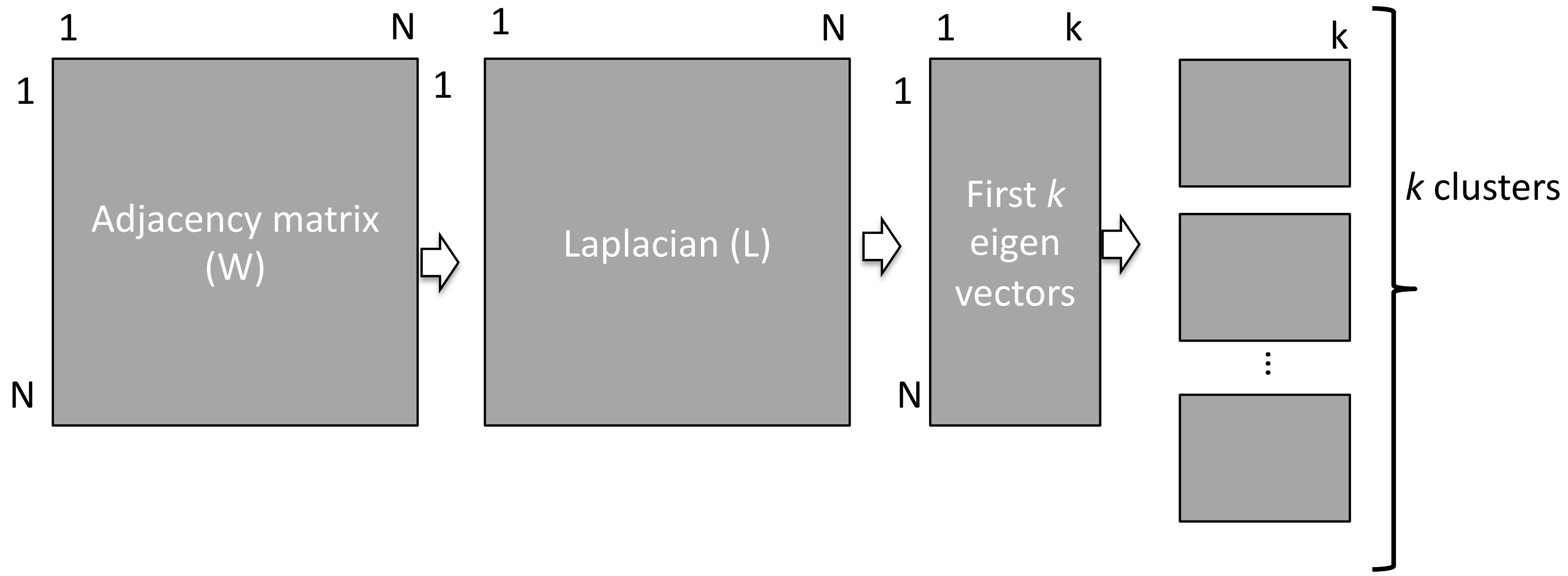
Goals for this lecture

- A few graph-theoretic concepts
  - Graph Laplacian
  - Connected components from the Laplacian
- Spectral clustering
  - Spectral clustering and graph cut
  - Spectral clustering demo with Zachary Karate Club
- Applications of spectral clustering

Spectral clustering

- Based on the graph Laplacian
- Graph Laplacian $L=D-W$
  - D is the diagonal degree of matrix
  - W is the adjacency matrix
- Obtain the k eigen vectors associated with k smallest eigen values of L
- Represent each node as the k-dimensional vector
- Cluster nodes based on k-means clustering

Spectral clustering key steps

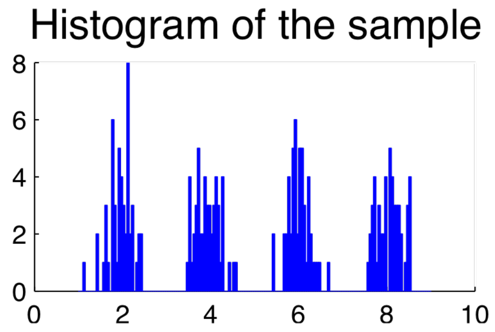


Spectral clustering can be applied on non-graph data

- Let $\{x_1 .. x_n\}$be a set of data points
- We can create a similarity graph or a distance graph using the pairwise similarity (s_{ij}) or distance (d_{ij}) with one of the following strategies
- ε- neighborhood graph
  - Connect all vertices v_i and v_j such that $d_{ij}<\varepsilon$
- k-nearest neighbor graph
  - Connect v_i to its k nearest neighbors
  - Make the graph symmetric
- Fully connected graph
  - Use s_{ij} as similarity for all pairs

Toy example of spectral clustering

- Let's consider spectral clustering for a toy dataset:
  - $\{x_1 .. x_{200}\}$: 200 points drawn from a mixture of four Gaussians



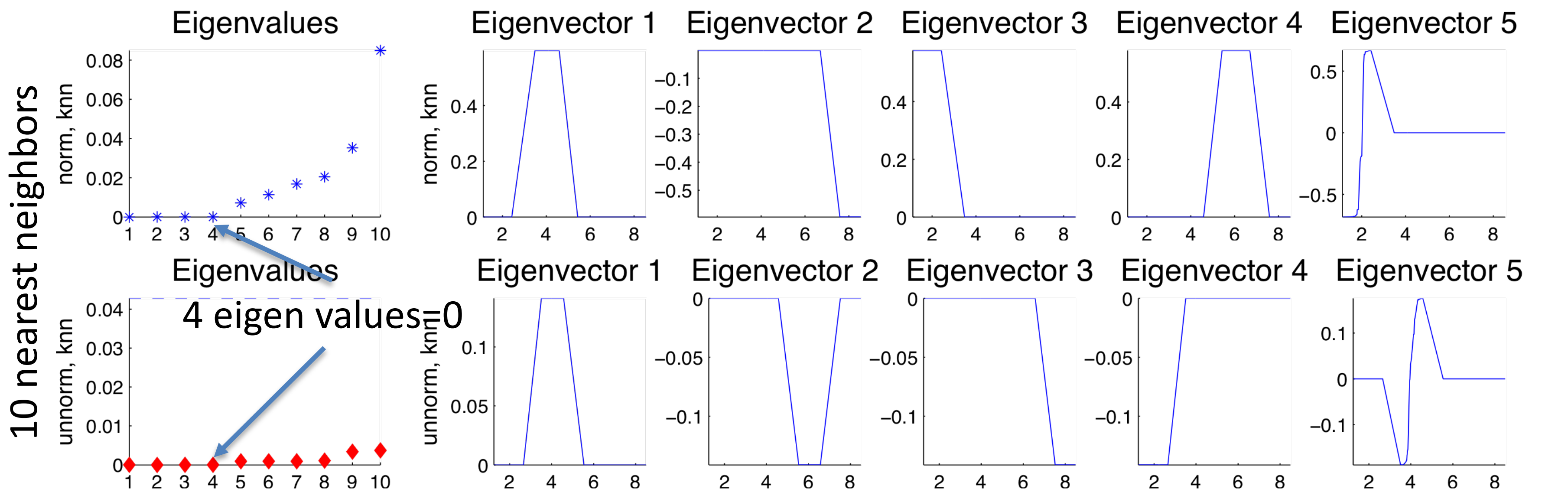
  - Use Gaussian similarity to create a graph

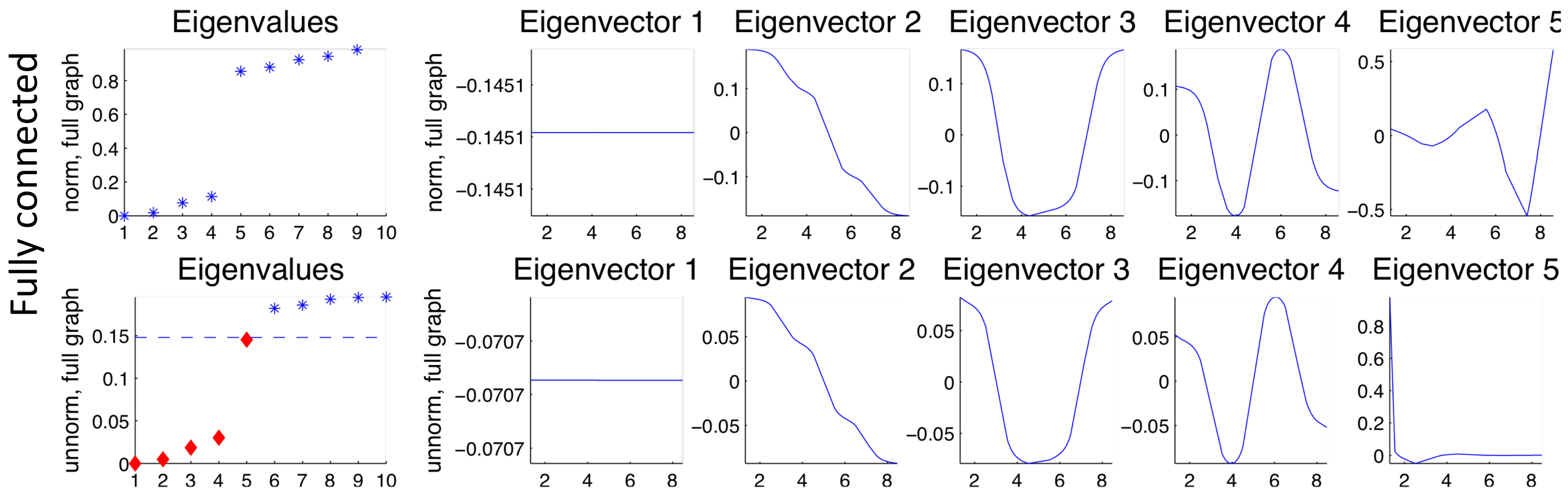
$$s_{ij} = \exp(-|x_i - x_j|^2)/2$$

  - Consider two variants : 10 nearest neighbors and fully connected graph

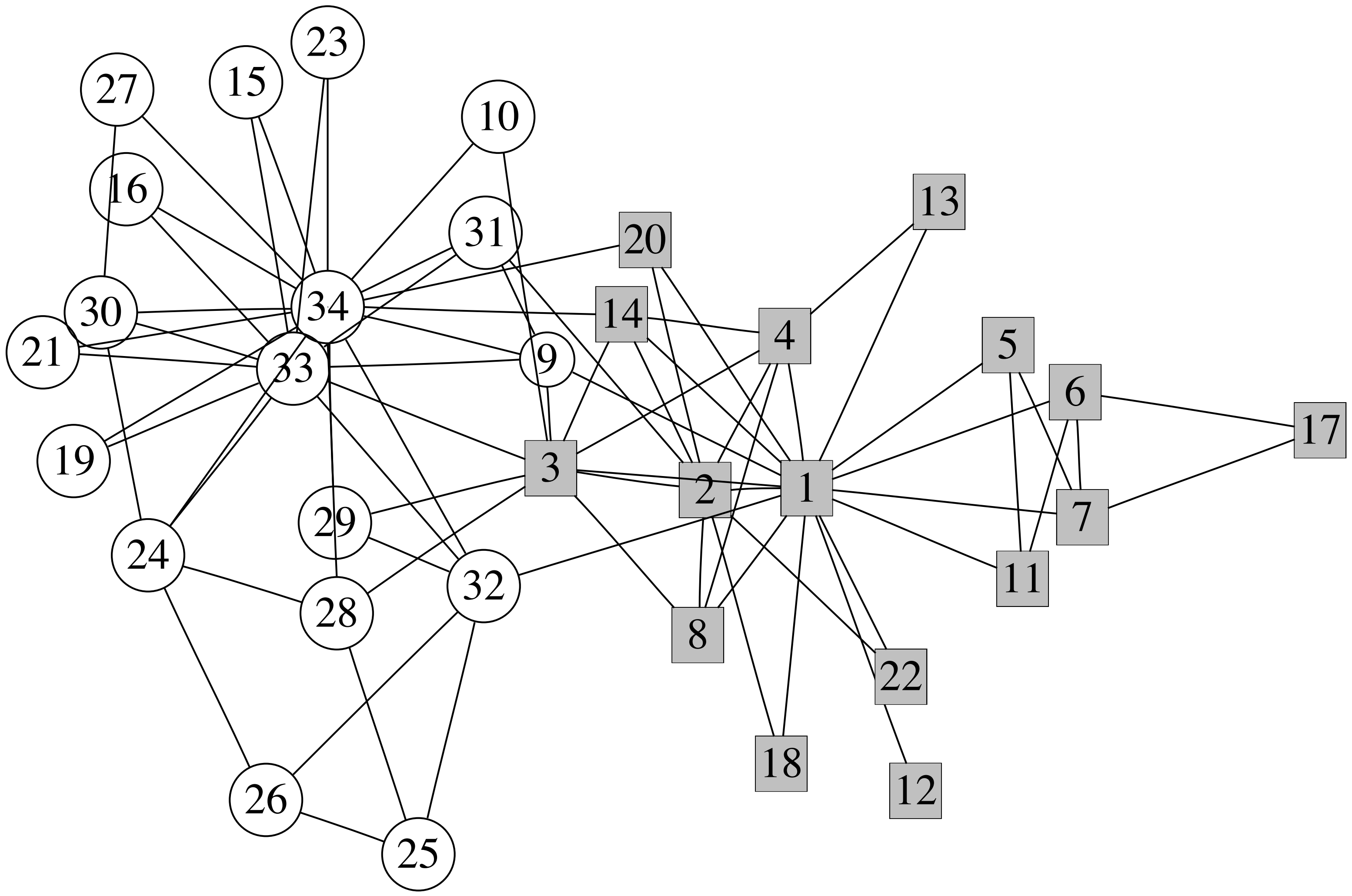
Luxburg tutorial on spectral clustering

Toy example of spectral clustering

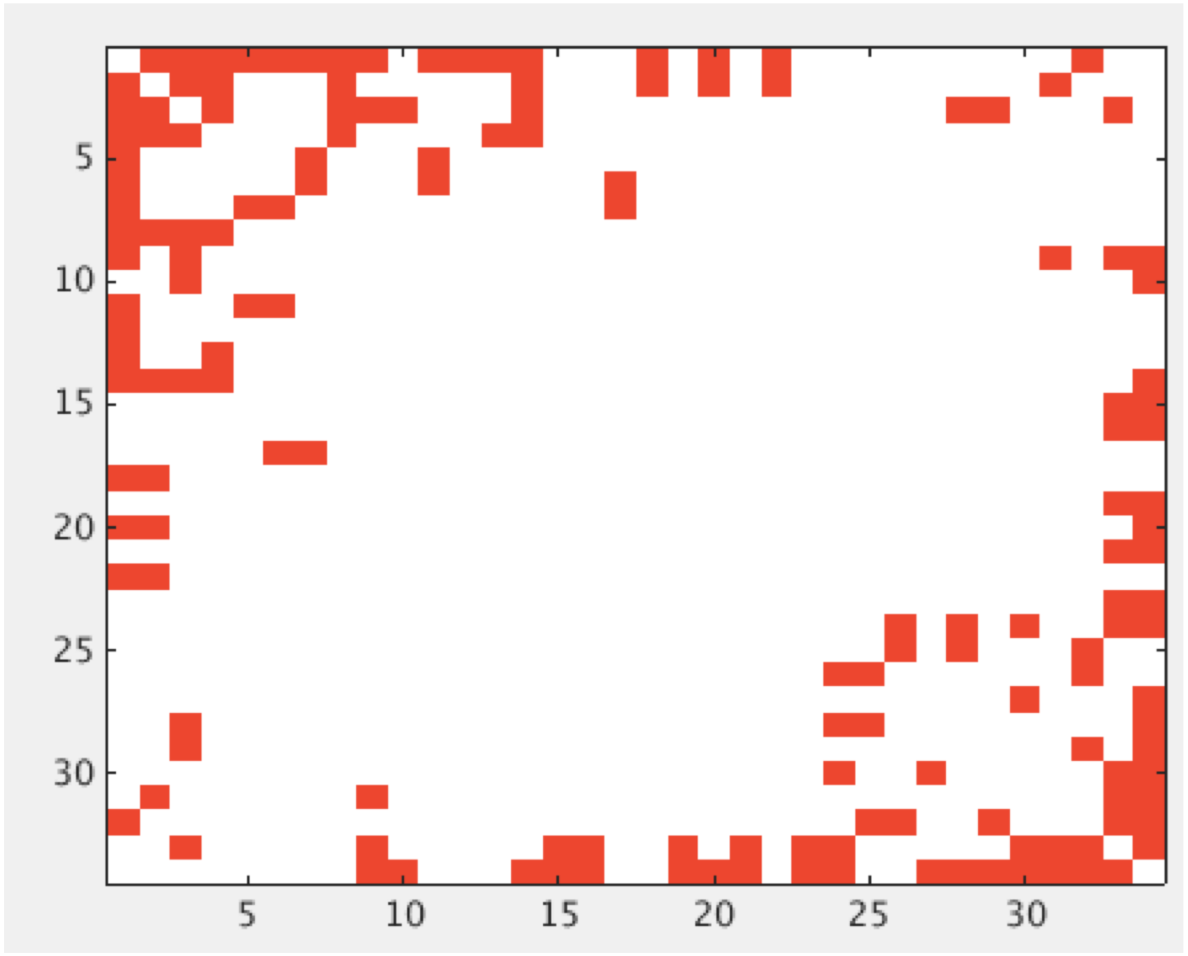




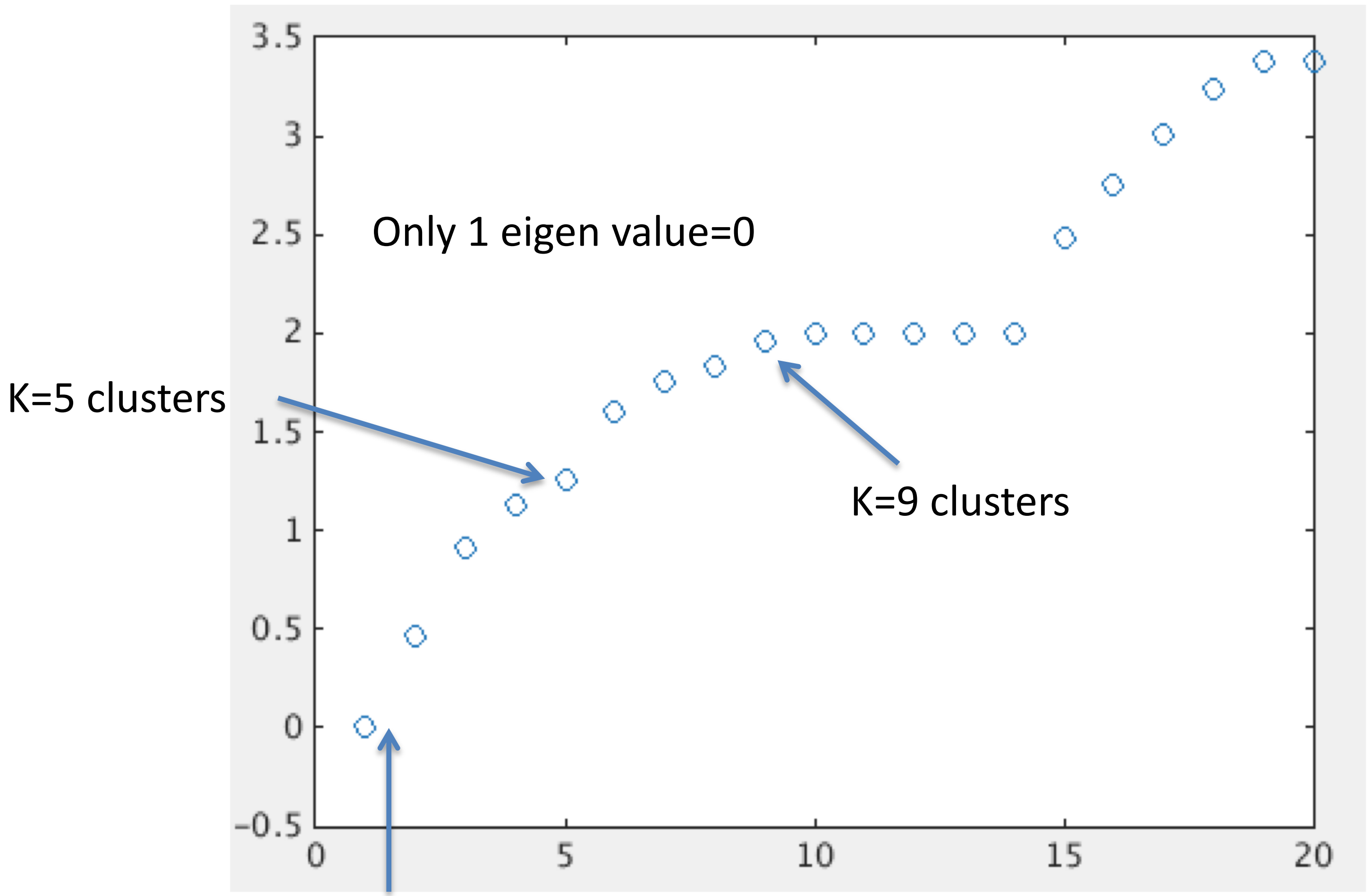
Recall the Zachary karate club study



Adjacency matrix of Zachary Karate club study

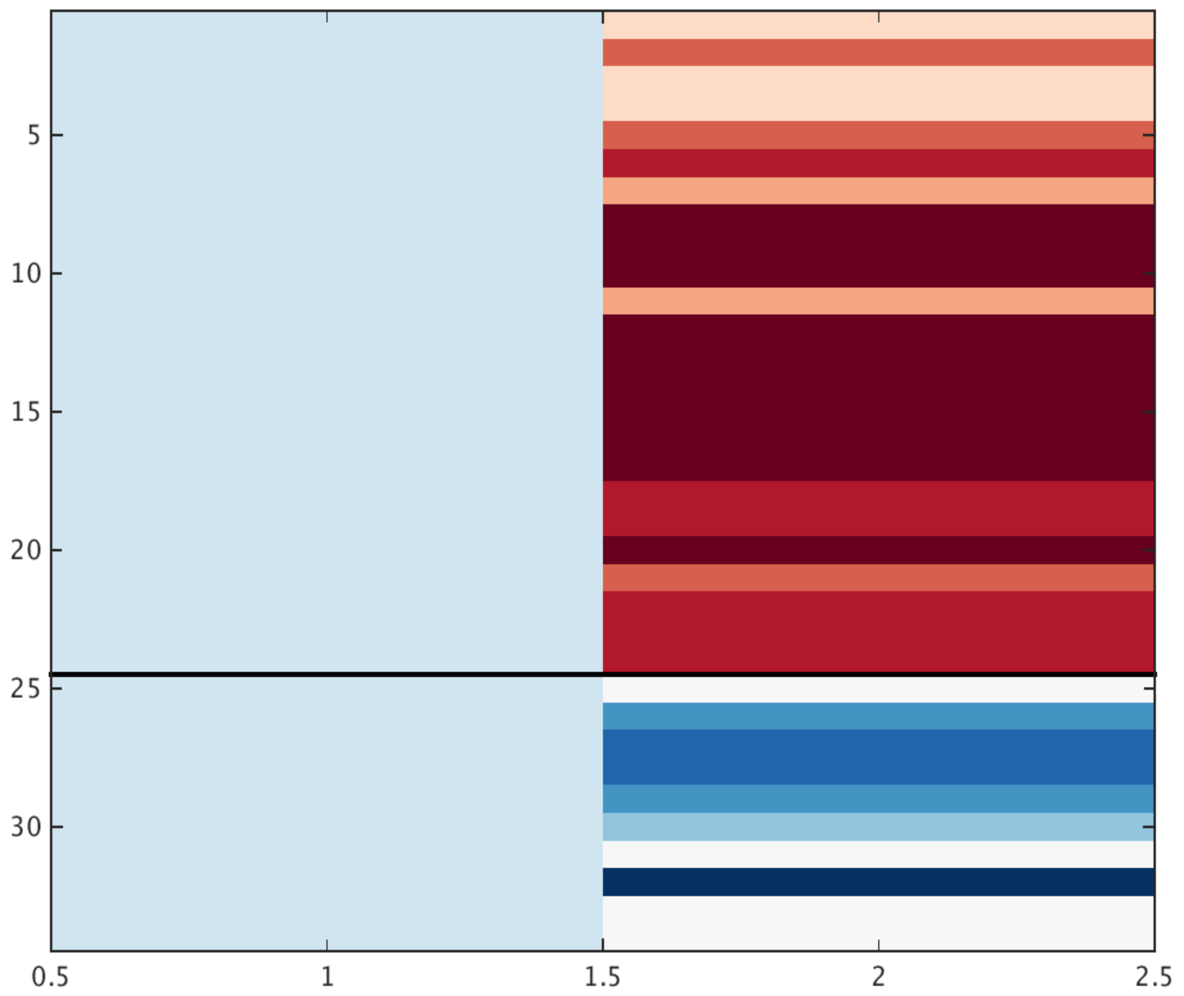


First 20 eigen values of the ZKC graph



Only one connected component

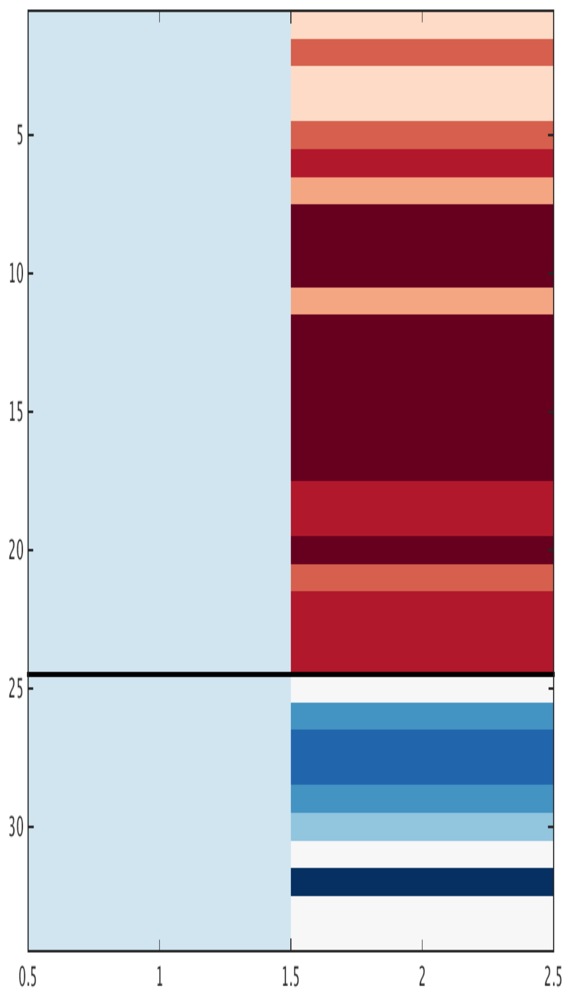
First two eigen vectors of the ZKC graph

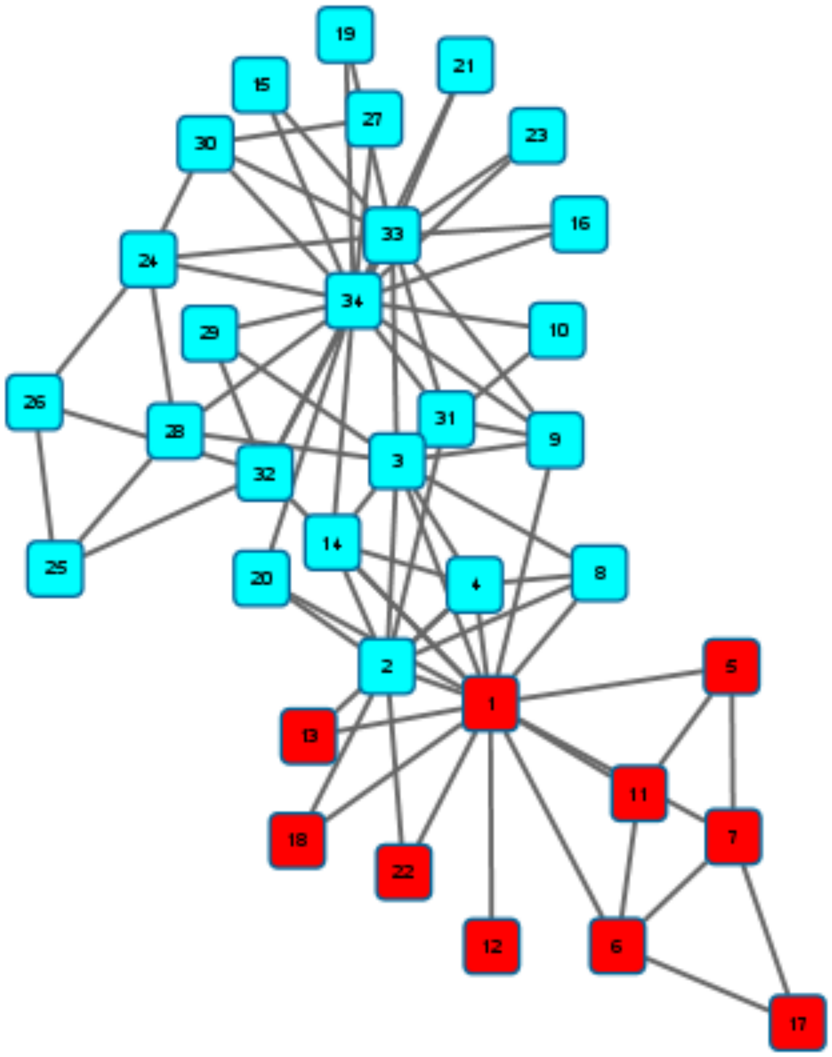


Cluster 2

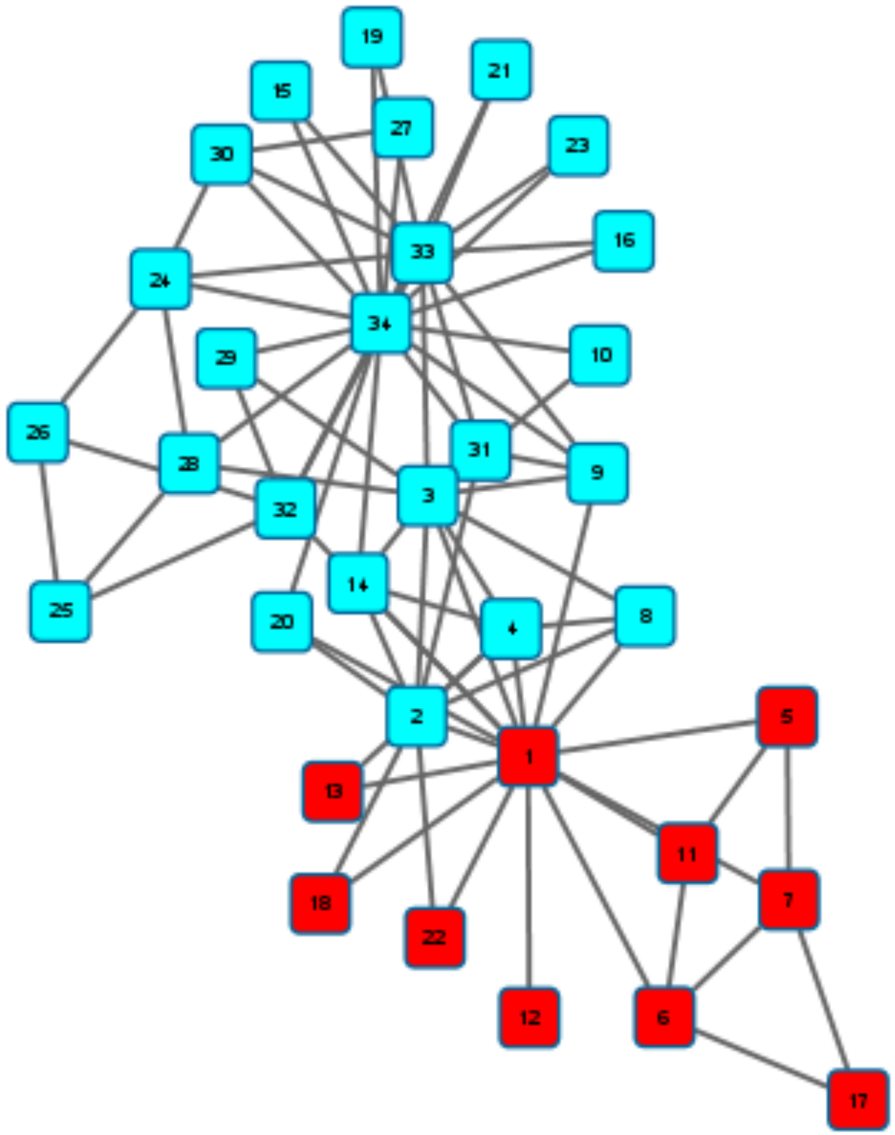
Cluster 1

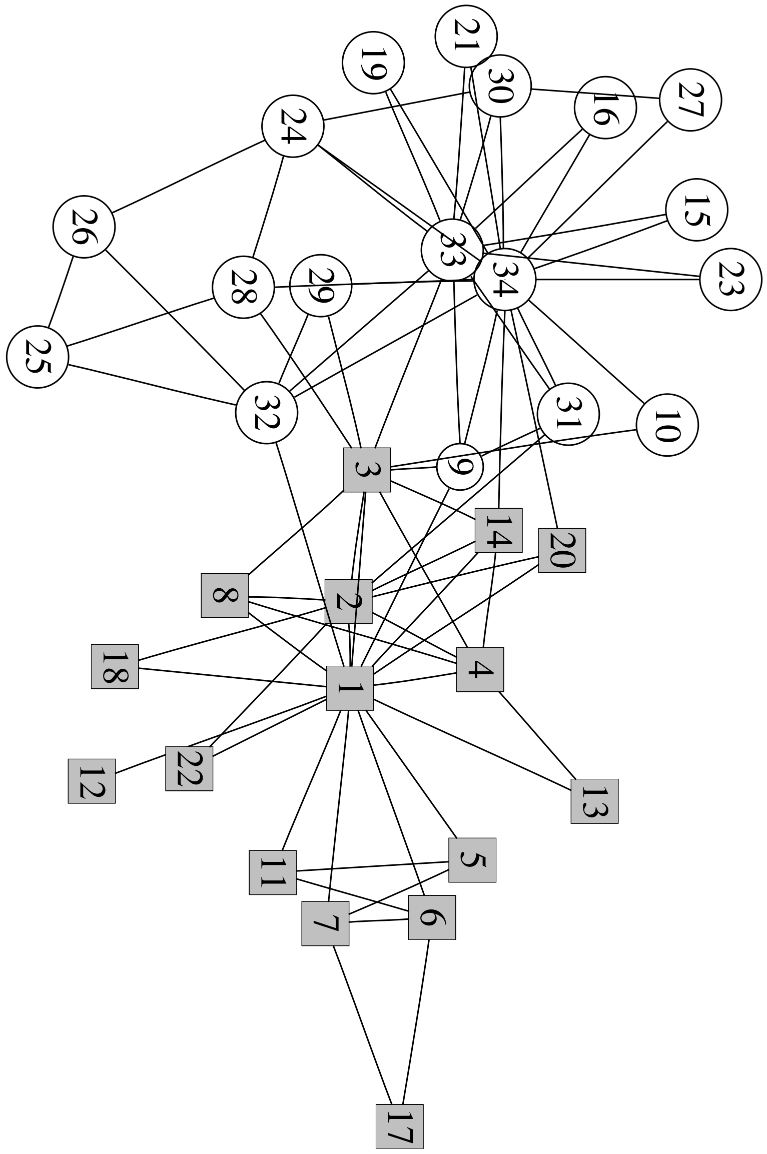
Clusters depicted on the graph



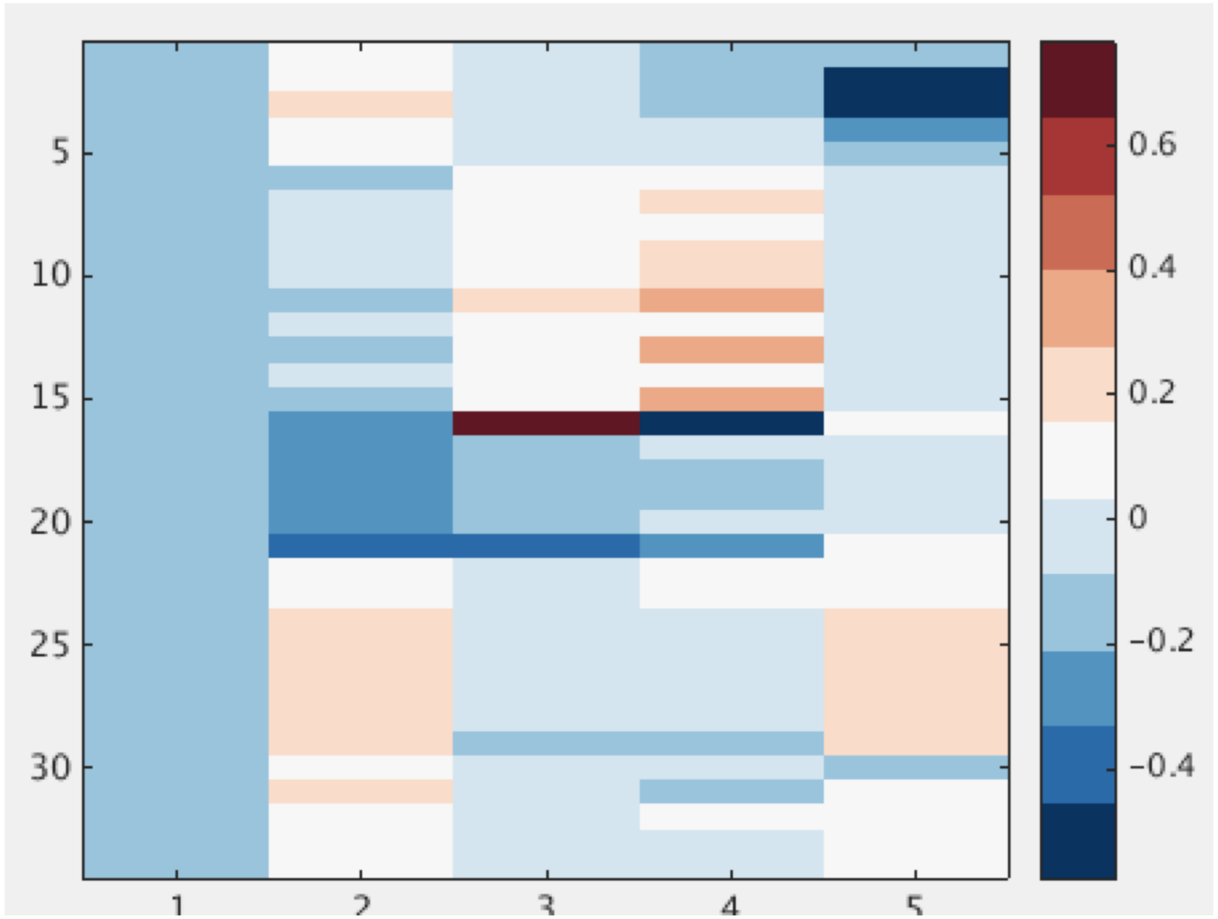


Spectral vs Girvan-Newman

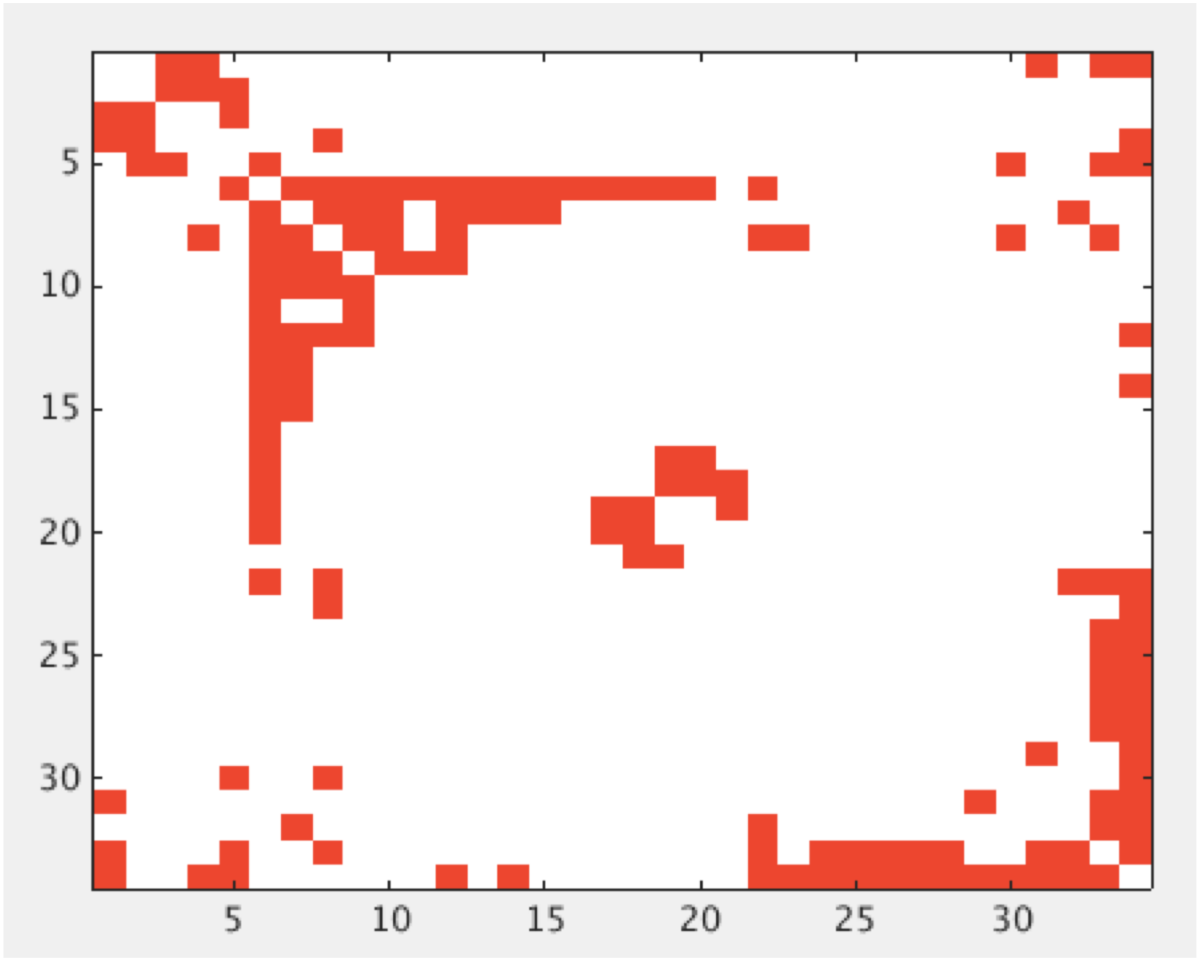




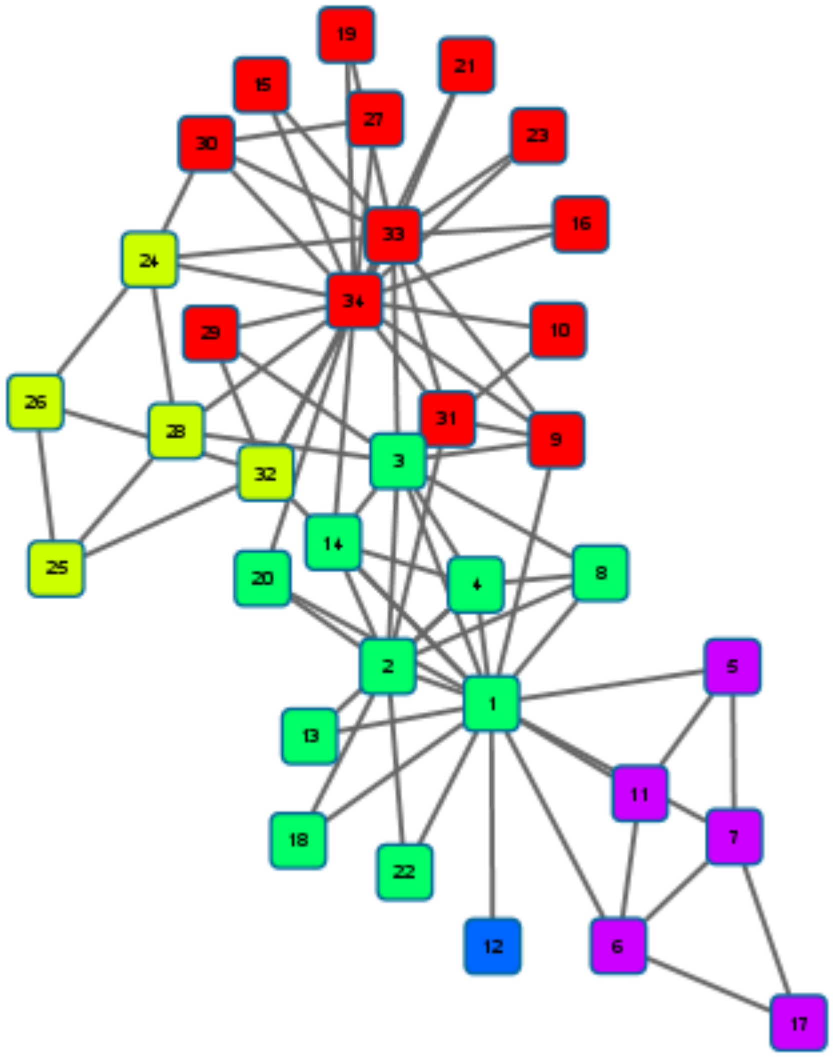
First five eigen vectors of Zachary Karate club data



Reordered matrix post clustering

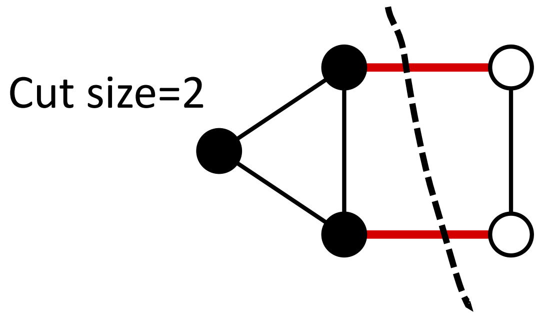


ZKC graph clustered into k=5 clusters



Graph cuts

- A cut on graph is defined as a partitioning on the nodes of the graph into two sets
- The size of a cut is the number of edges spanning the cut



Different types of graph cuts

- A cut can be
  - Min Cut: No other cut is smaller (fewer edges)
  - Max Cut: No other cut is larger

Graph clustering from a graph cut point of view

- Clustering on the graph can be re-stated as follows:
  - Find a partition of the graph such that the edges between different groups have a very low weight (which means that points in different clusters are dissimilar from each other) and the edges within a group have high weight (which means that points within the same cluster are similar to each other).
- The most direct way to find such a partition is by solving the Min-Cut problem
  - This is an NP-hard problem
- The spectral clustering algorithm can be thought of as solving a continuous relaxation of this problem

Luxburg Tutorial on Spectral graph clustering

Goals for this lecture

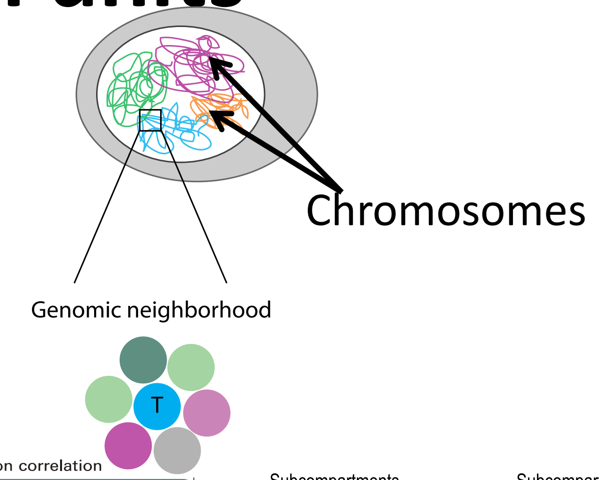
- A few graph-theoretic concepts
  - Graph Laplacian
  - Connected components from the Laplacian
- Spectral clustering
  - Spectral clustering and graph cut
  - Spectral clustering demo with Zachary Karate Club
- Application of spectral clustering

Application of spectral clustering

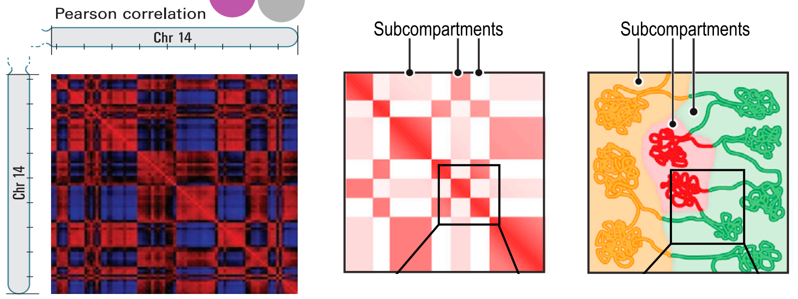
- Finding higher-order Topologically Associated Domains from Hi-C data
- Disease module identification
- Similarity network fusion for aggregating data types on a genomic scale

Genome is organized into multiple organizational units

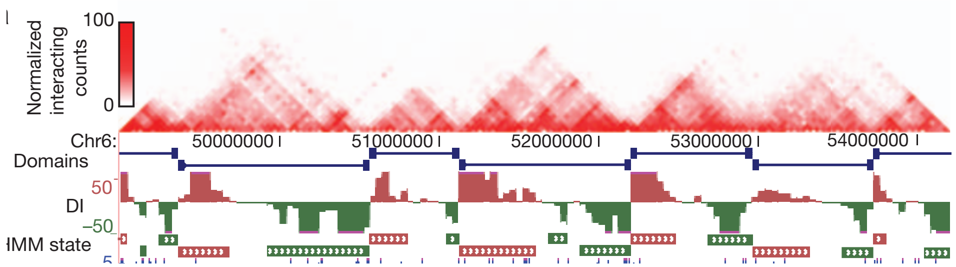
- Chromosomal territories through inter-TAD interactions



- Compartments and sub-compartments

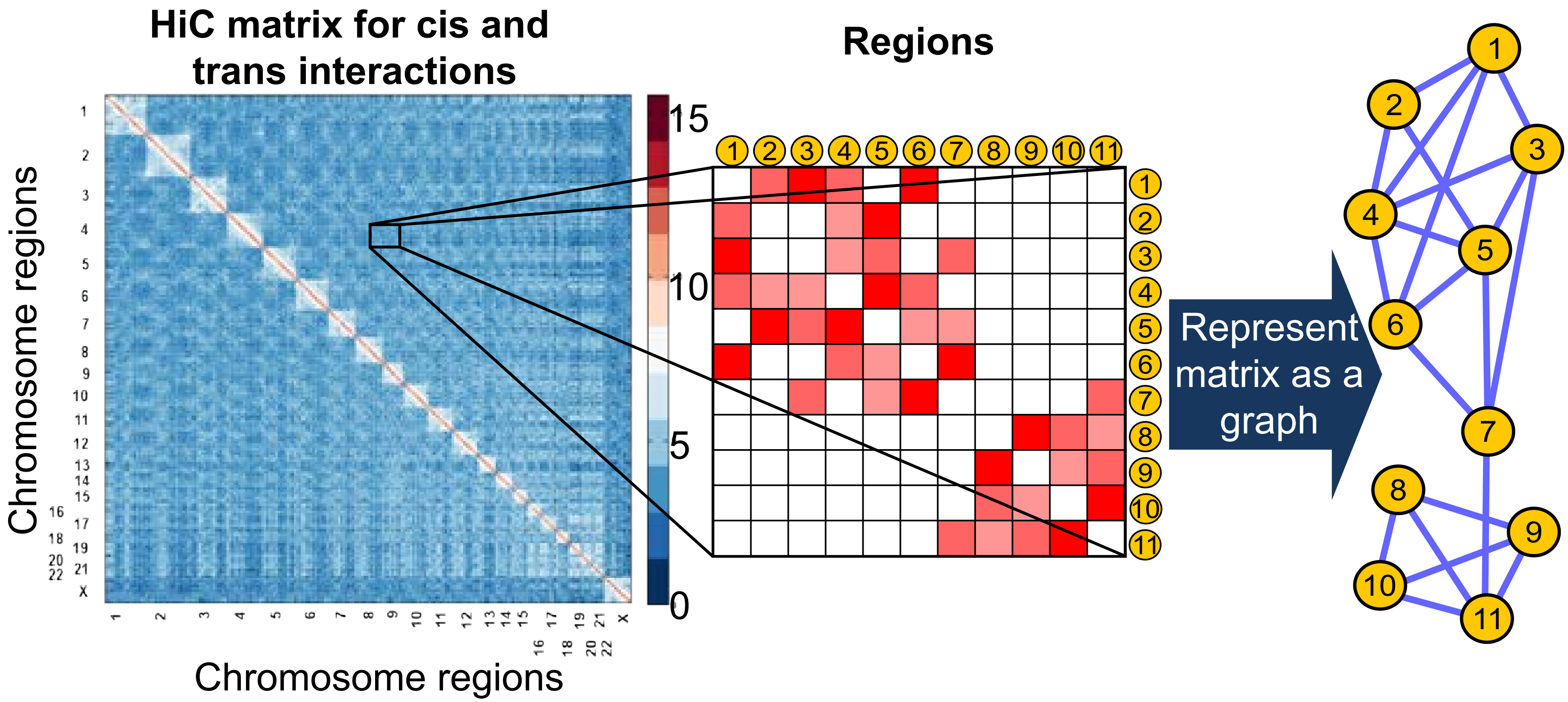


- Topologically associated domains (TADs) and sub-TADs

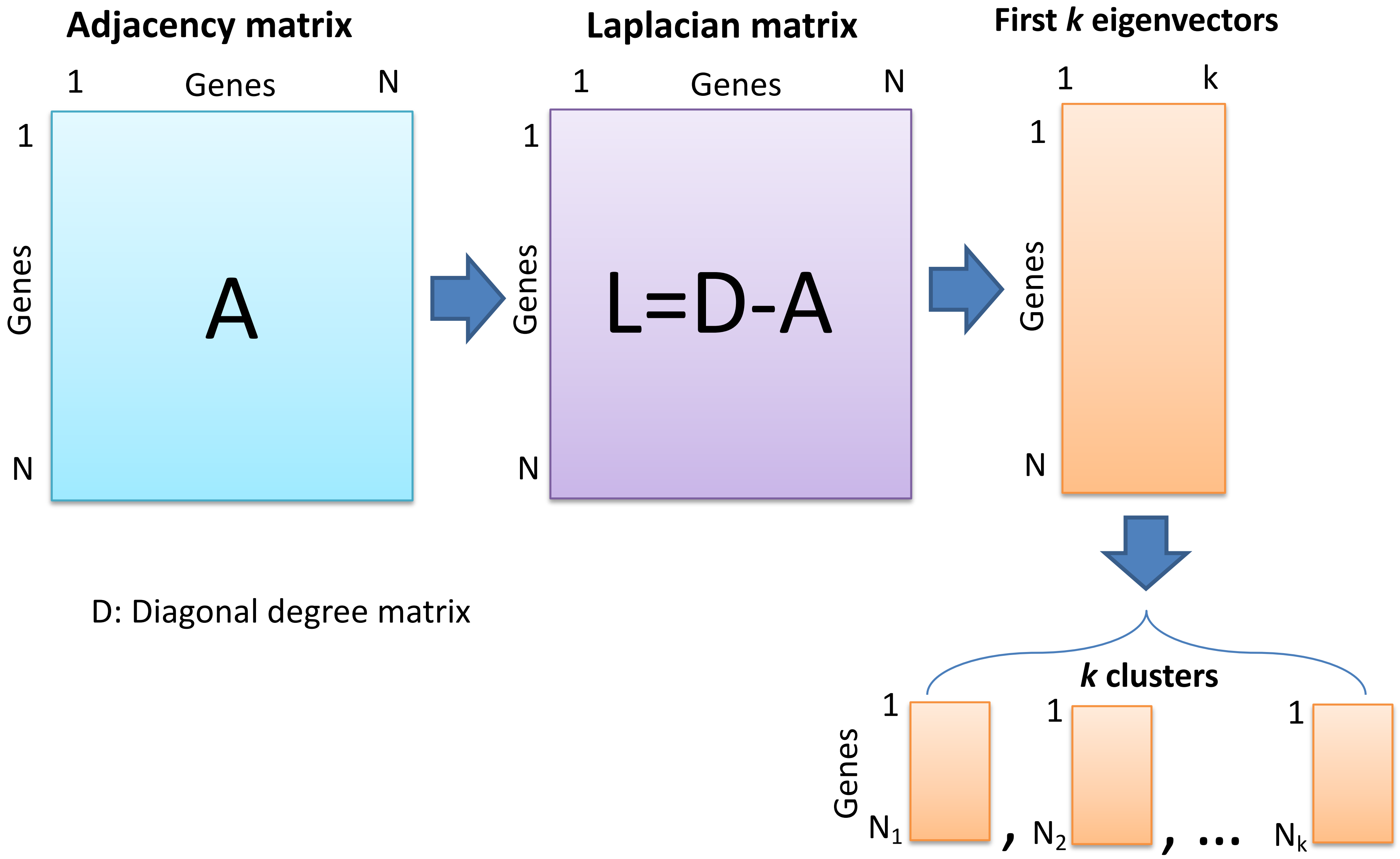


Lierberman et al 2009, Rao et 2014, Dixon et al 2012

A graph is a natural representation of a Hi-C dataset

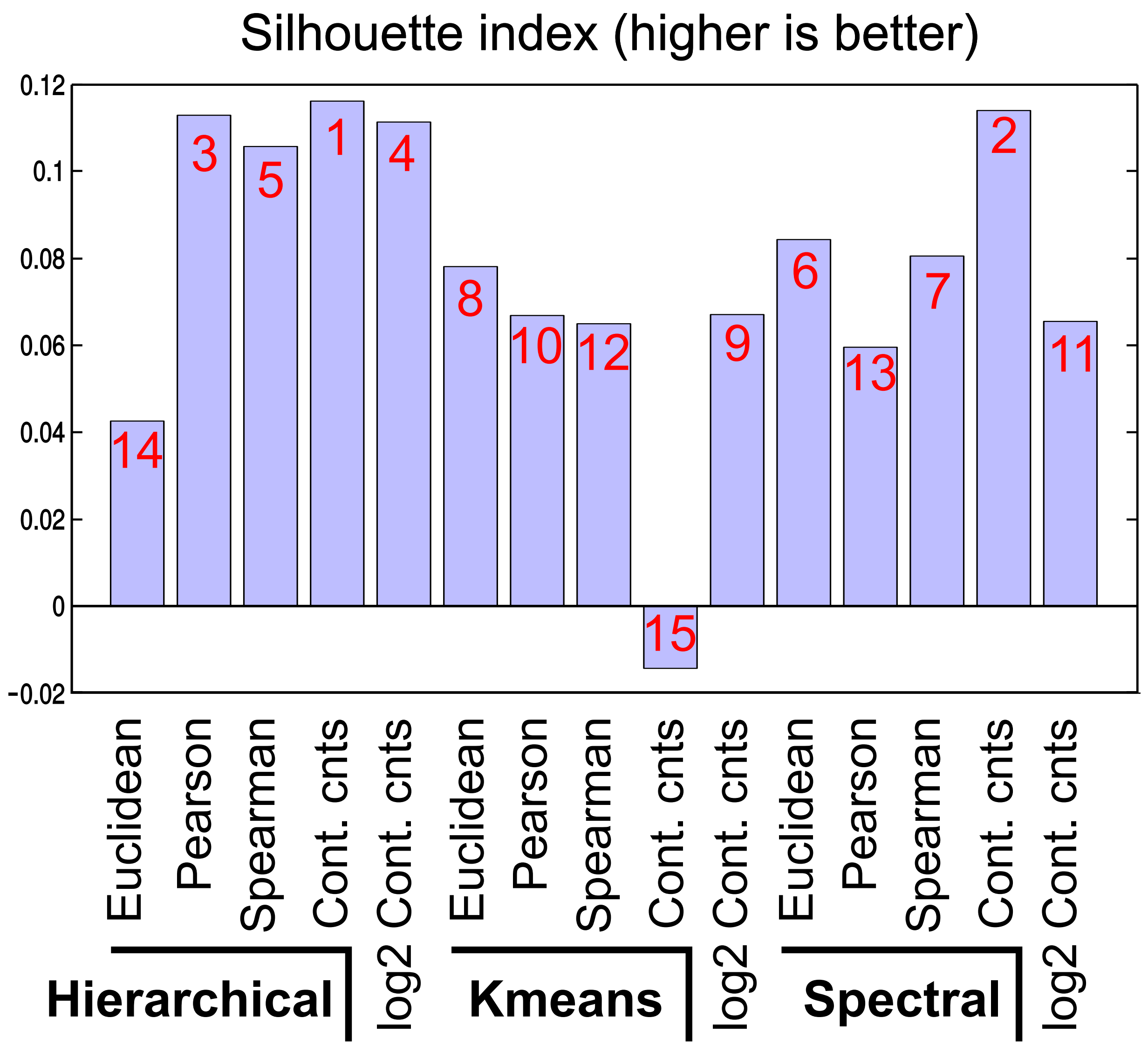


An overview of spectral clustering

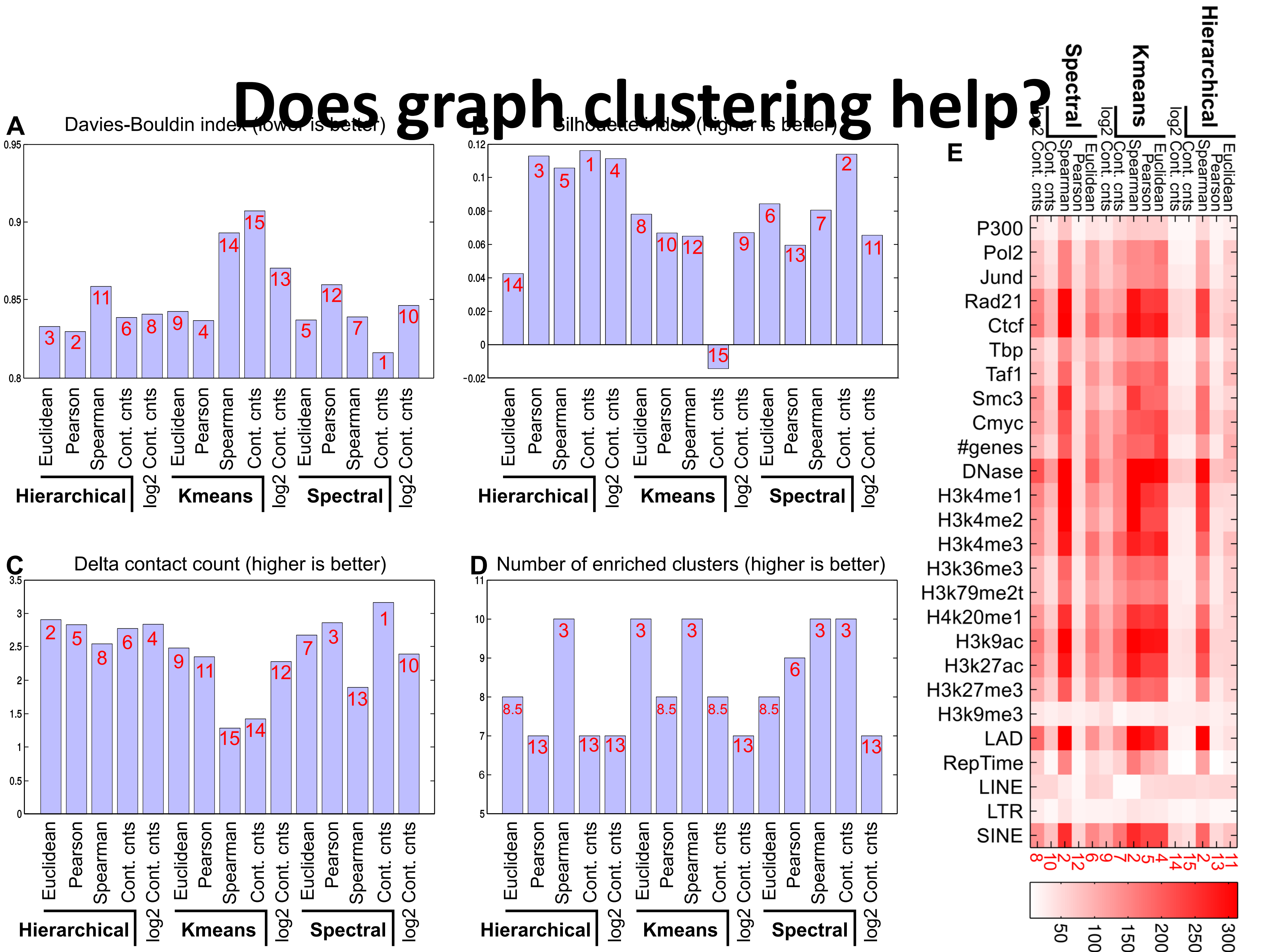


D: Diagonal degree matrix

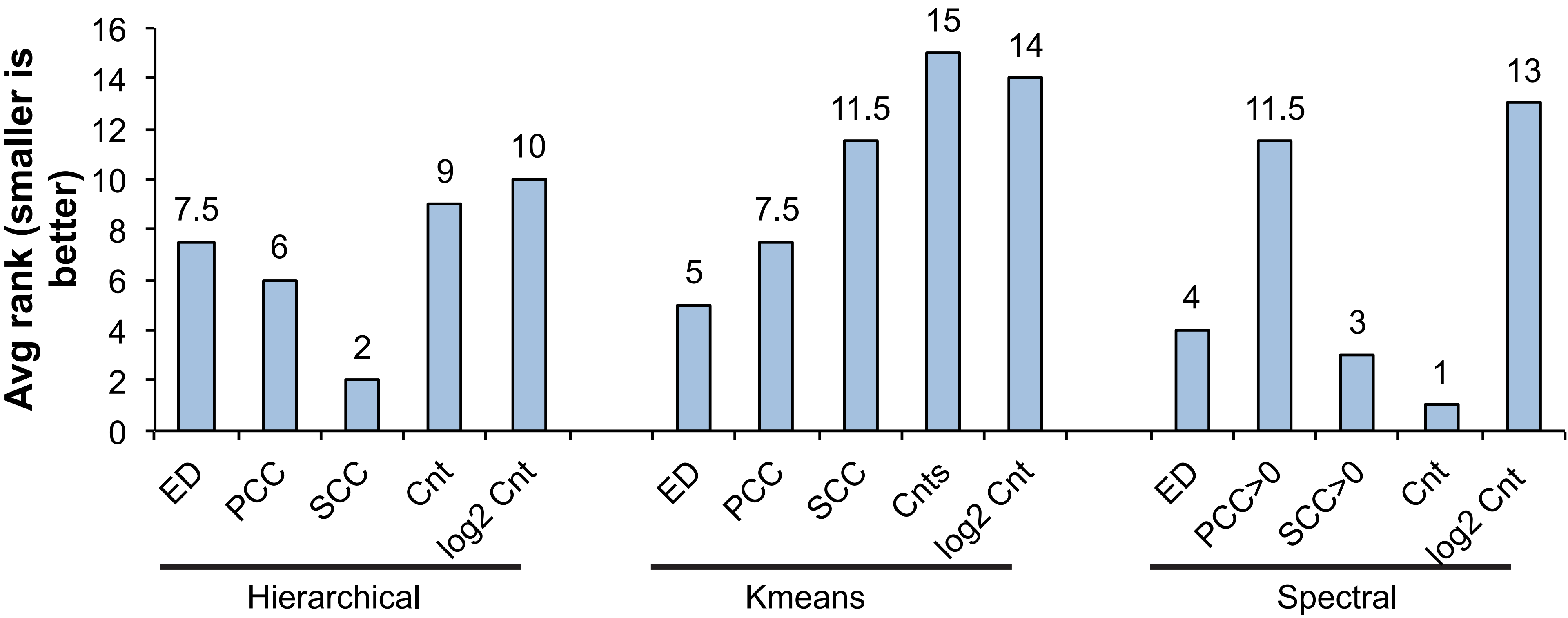
Does graph clustering help?



Does graph clustering help?

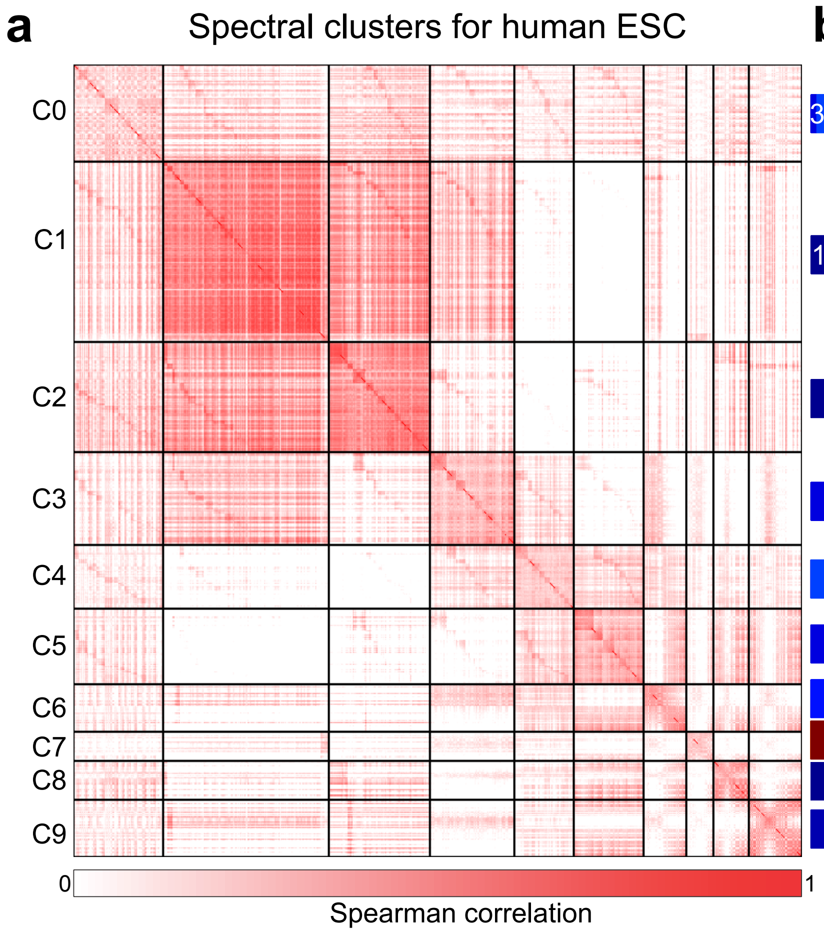


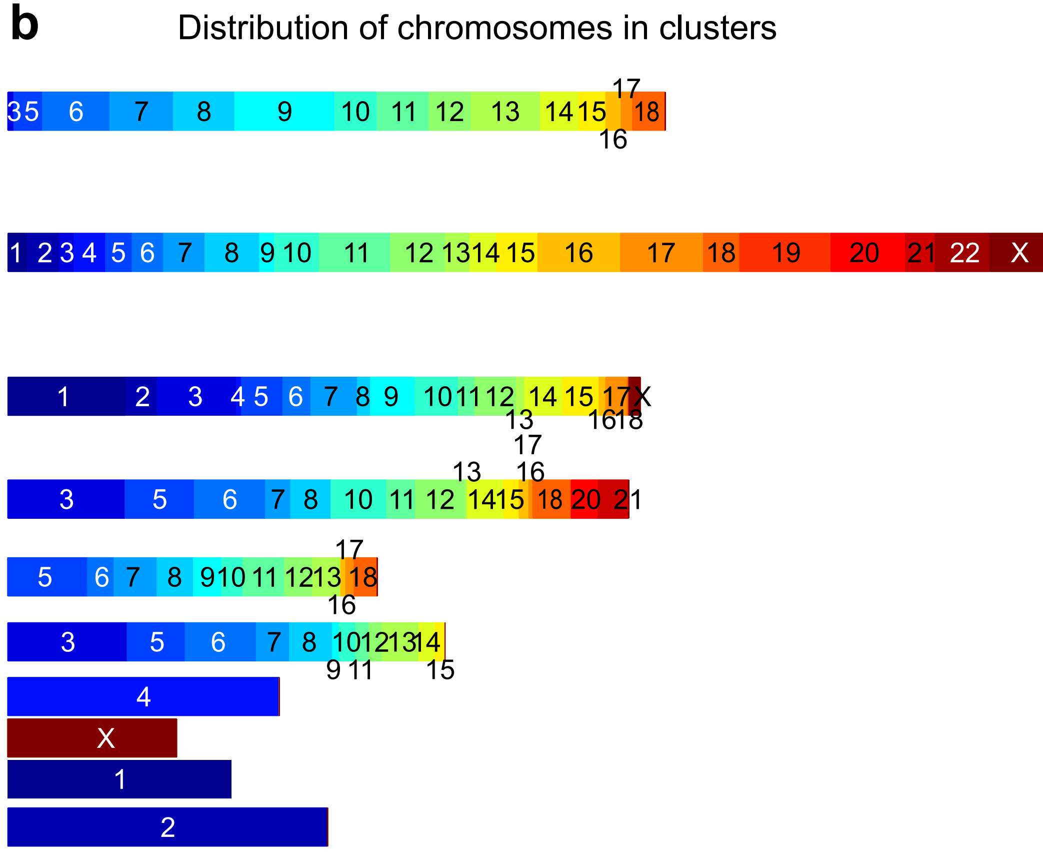
Does graph clustering help?



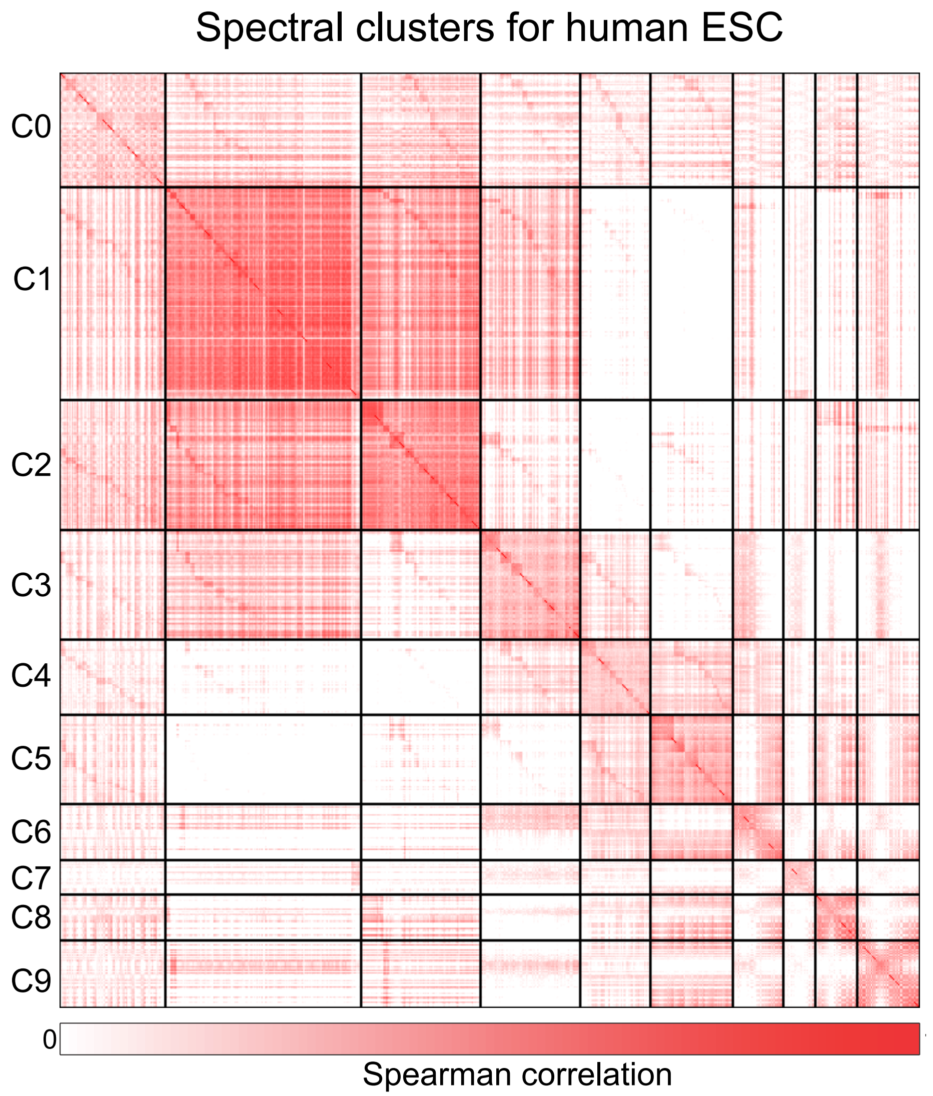
Spectral (graph) clustering methods tend to do better on different measures

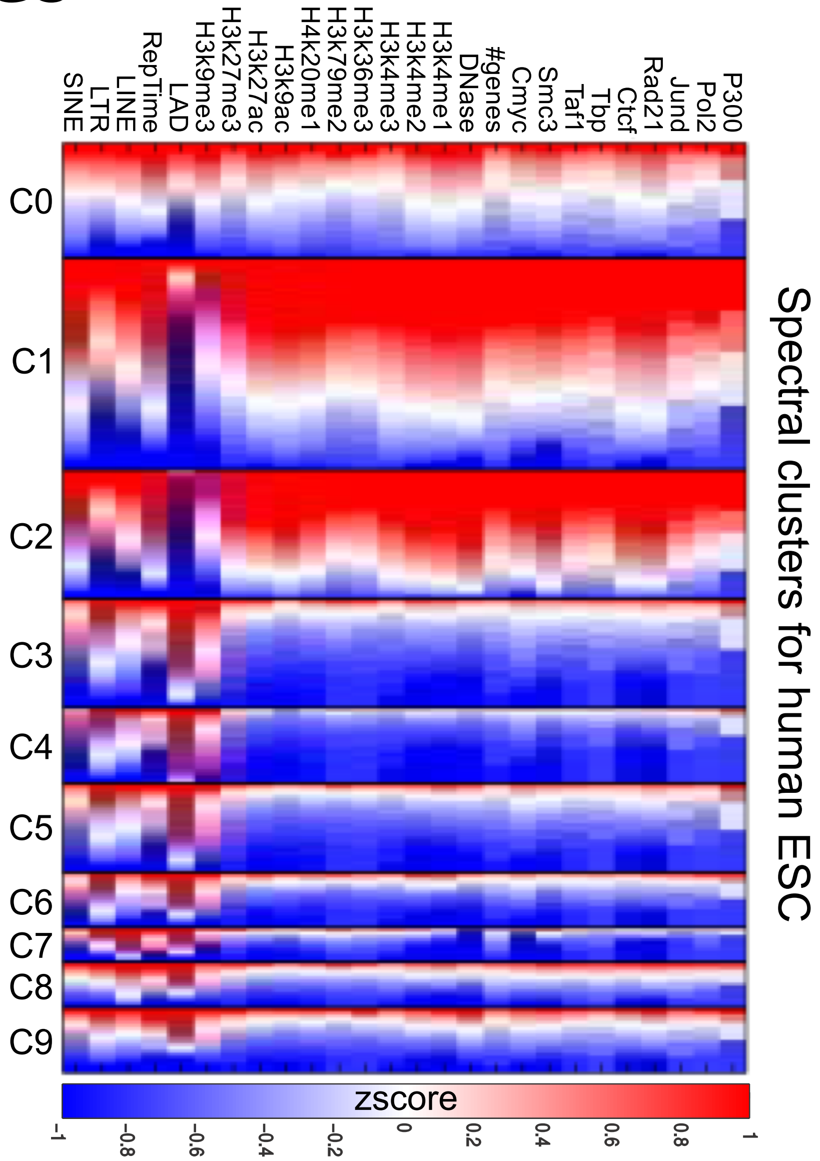
Spectral clustering of Hi-C data of human ESC





Two main types of chromatin interaction modules





Application of spectral clustering

- Finding higher-order Topologically Associated Domains from Hi-C data
- Disease module identification
- Similarity network fusion for aggregating data types on a genomic scale

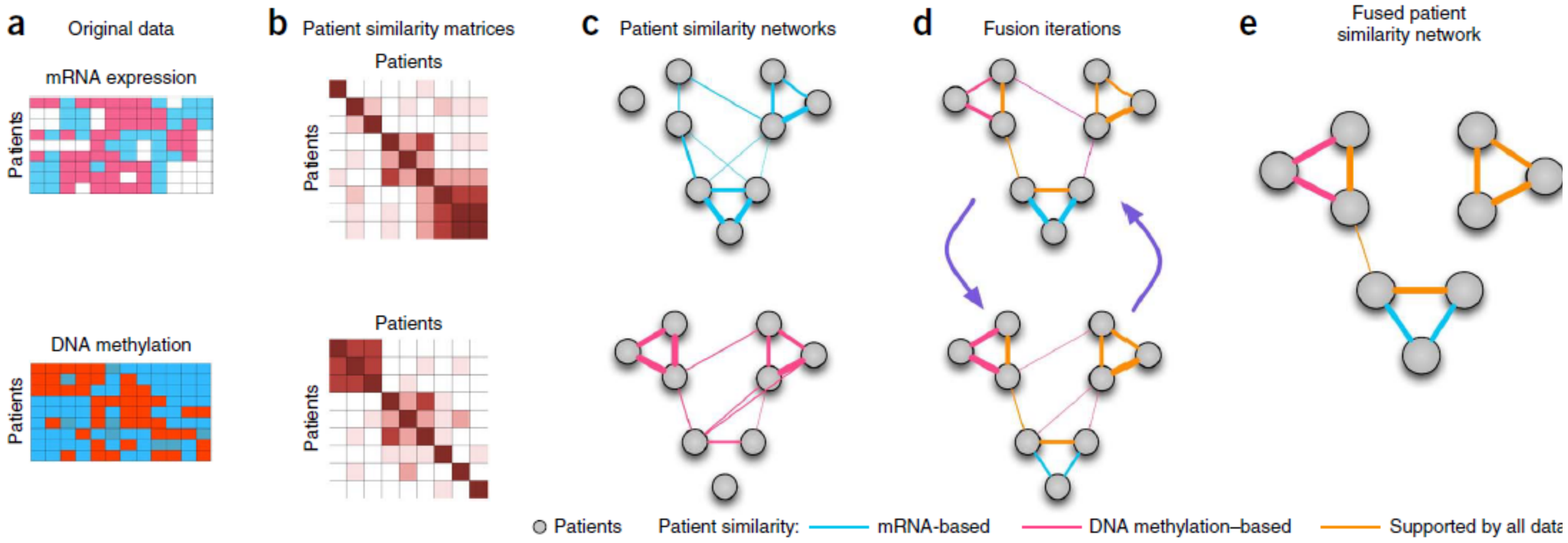
Similarity network fusion for aggregating data types on a genomic scale

- This paper had two goals:
  - Integrate different types of data using a network-based approach
  - Identify groups of samples representing integrated data types
- Recent high throughput technologies have made it possible to collect many different types of genomic data for individual patients
- How do we combine patient data to describe a disease?
- This is challenging because of the following issues:
  - Noisy samples
  - Small number of samples than variables
  - Complimentary nature of the data

Similarity Network Fusion

- Given N different types of measurements for different individuals
- Do
  - Construct a similarity matrix of individuals for each data type
  - Integrate the networks using a single similarity matrix using an iterative algorithm
  - Cluster the network into a groups of individuals

Similarity network fusion with two data types



Similarity network fusion (Nodes are patients, edges represent similarities).

Defining a similarity graph over patient samples

- For each data type, create a weighted graph, with vertices corresponding to patients
- Let x_i and x_j denote the measurements of patients i and j
- Edge weights, $W(i,j)$ correspond to how similar patient i is to patient j based on x_i and x_j

Euclidean distance

$$W(i,j) = exp(-\frac{\rho^2(x_i, x_j)}{\mu\epsilon_{i,j}})$$

Hyper-parameter

Scaling term (average of the distance between each node and its neighborhood)

Creating a fused matrix

- Define two matrices for each data type
- A full matrix: normalized weight matrix

$$\mathbf{P}(i,j) = \begin{cases} \dfrac{\mathbf{W}(i,j)}{2\Sigma_{k \neq i}\mathbf{W}(i,k)}, j \neq i \\ 1/2, j = i \end{cases}$$

- A sparse matrix (based on k nearest neighbors or each node)

$$\mathbf{S}(i,j) = \begin{cases} \dfrac{\mathbf{W}(i,j)}{\Sigma_{k \in N_i}\mathbf{W}(i,k)}, & j \in N_i \\ 0 & \text{otherwise} \end{cases}$$

This makes the assumption that the local similarities are the most reliable

Iterate for fusion

- Input m data types
- Construct $W^{(v)}$ for each data type v
- Construct dense matrix $P^{(v)}$ and sparse matrix $S^{(v)}$
- At each iteration, update the dense similarity matrix of one data type using the similarity matrix of the other data type

Iteration with m=2 data types

For iteration *t+1*

Update similarity matrix of data type 1

$$\mathbf{P}_{t+1}^{(1)} = \mathbf{S}^{(1)} \times \mathbf{P}_t^{(2)} \times (\mathbf{S}^{(1)})^T$$

Update similarity matrix of data type 2

$$\mathbf{P}_{t+1}^{(2)} = \mathbf{S}^{(2)} \times \mathbf{P}_t^{(1)} \times (\mathbf{S}^{(2)})^T$$

Update similarity matrix of data type 1 using weight matrix from data type 2 and vice-versa

What is going on in the iteration step

Neighbors of i

$$\mathbf{P}^{(1)}_{t+1}(i,j) = \sum_{k \in N_i} \sum_{l \in N_j} \mathbf{S}^{(1)}(i,k) \times \mathbf{S}^{(1)}(j,l) \times \mathbf{P}^{(2)}_{t}(k,l)$$

Neighbors of j

We are updating the similarity matrix using the most confident common neighbors of *i* and *j*

Extending to m>2 data types

$$\mathbf{P}^{(v)} = \mathbf{S}^{(v)} \times \left(\frac{\Sigma_{k \neq v} \mathbf{P}^{(k)}}{m-1} \right) \times (\mathbf{S}^{(v)})^T, v = 1, 2, \cdots, m$$

$$\downarrow$$

Just average over all other data types

SNF termination

- After repeating the iterative updates for t steps, final similarity matrix is

$$\mathbf{P} = \frac{1}{m}\sum_{k=1}^{m}\mathbf{P}_t^k$$

- This is then clustered using spectral clustering

Application of SNF to Glioblastoma

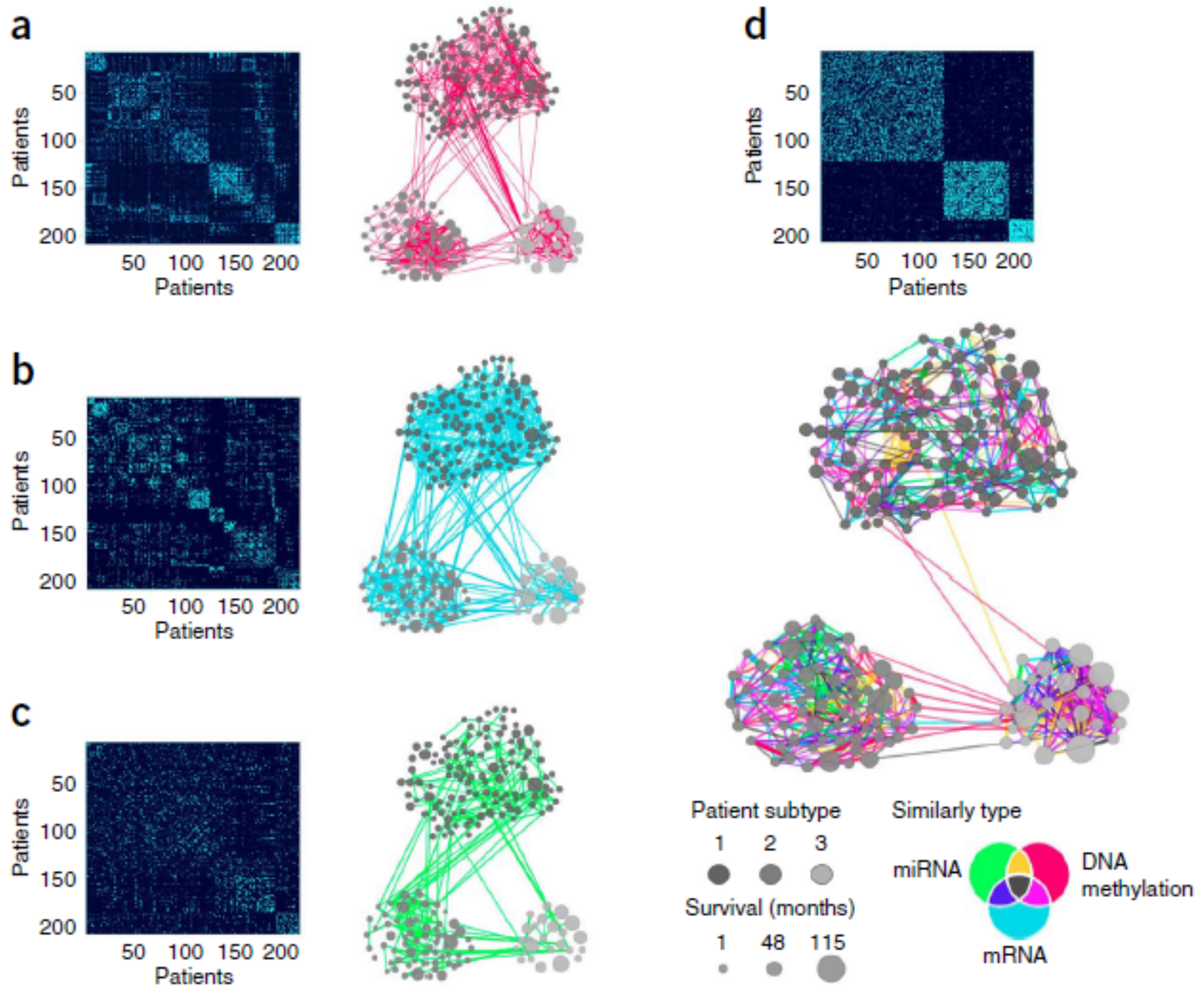
- Contradicting information about subtypes depending upon the type of data used
- Glioblastoma dataset
- Three data types among 215 patients
    - DNA methylation (1491 genes)
    - mRNA (12,042 genes)
    - miRNA (534 miRNAs)

SNF application to GBM identifies 3 subtypes

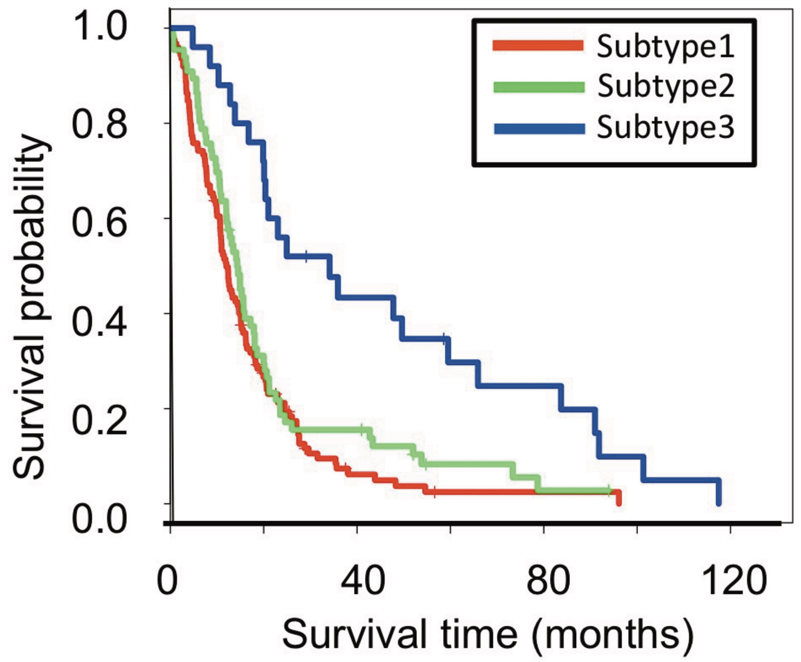
DNA methylation

mRNA expression

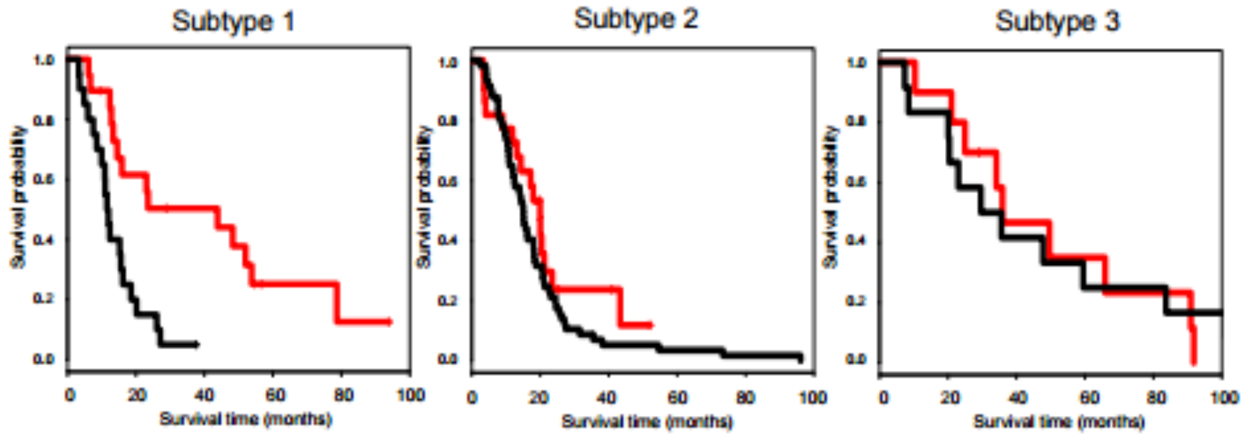
miRNA expression



Validation of SNF identified subtypes



Subtypes are associated with patient populations of different survival.
Blue curve (subtype 3) are patients with more favorable prognosis



Key points of graph clustering algorithms

- Flat or hierarchical clustering
- Algorithms differ in
  - how they define the similarity/distance measure
    - Local topology measures
    - Global measures
  - Whether the algorithm takes as input the number of clusters or the goodness of clusters (e.g. the approximate cluster algorithm)

References

- A Tutorial on Spectral Clustering.
  - Ulrike von Luxburg, 2007