Applications of graph clustering

Sushmita Roy

sroy@biostat.wisc.edu

Computational Network Biology

Biostatistics & Medical Informatics 826

https://compnetbiocourse.discovery.wisc.edu

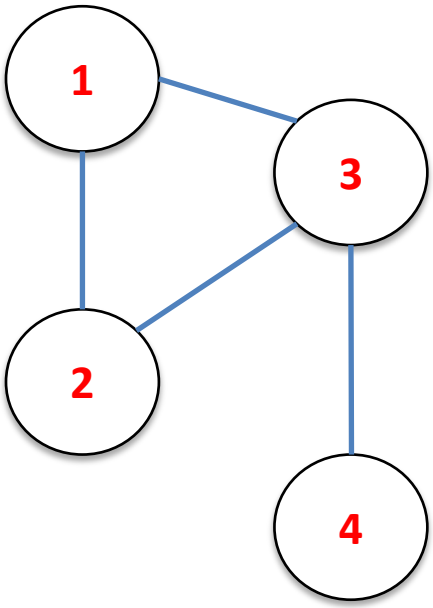
Nov 7th 2018

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)

	1	2	3	4
1	0	1	1	0
2	1	0	1	0
3	1	1	0	1
4	0	0	1	0

Degree matrix (D)

	1	2	3	4
1	2	0	0	0
2	0	2	0	0
3	0	0	3	0
4	0	0	0	1

Laplacian ($L=D-W$)

	1	2	3	4
1	2	-1	-1	0
2	-1	2	-1	0
3	-1	-1	3	-1
4	0	0	-1	1

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

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

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 , . . . , 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

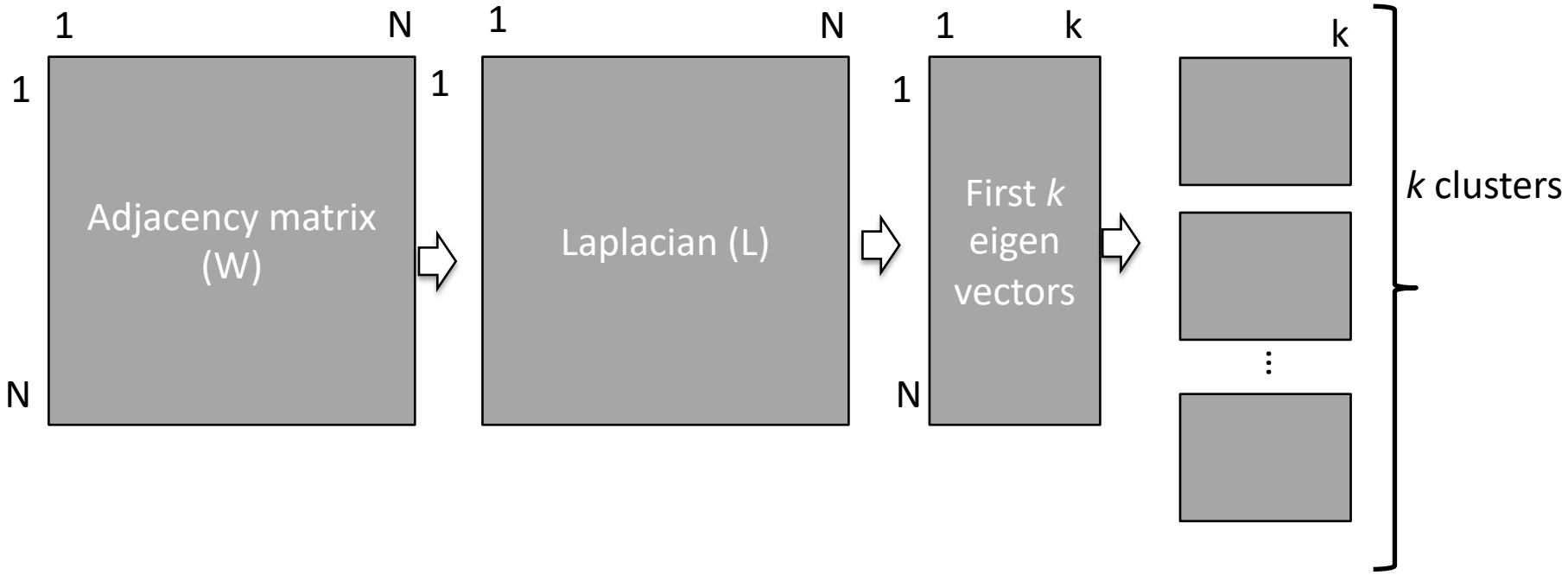
$$f'Lf = \frac{1}{2}\sum_{i,j} w_{ij}(f_i - f_j)^2$$

- 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

RECAP: 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

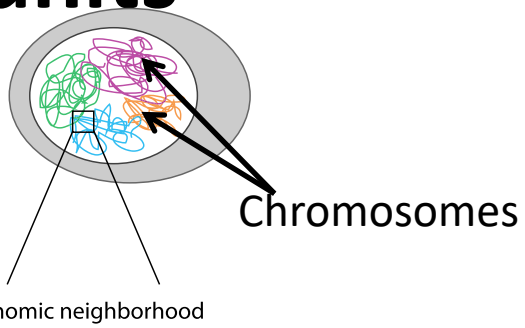


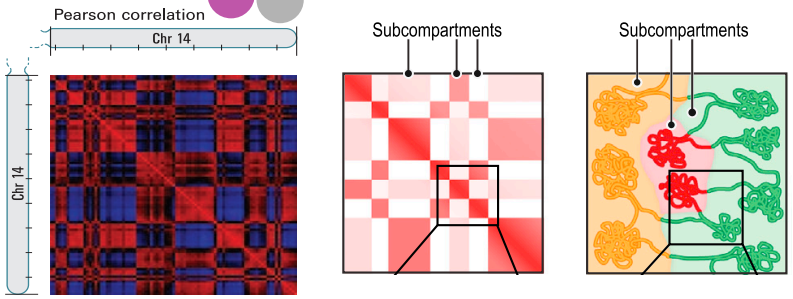
Application of graph clustering

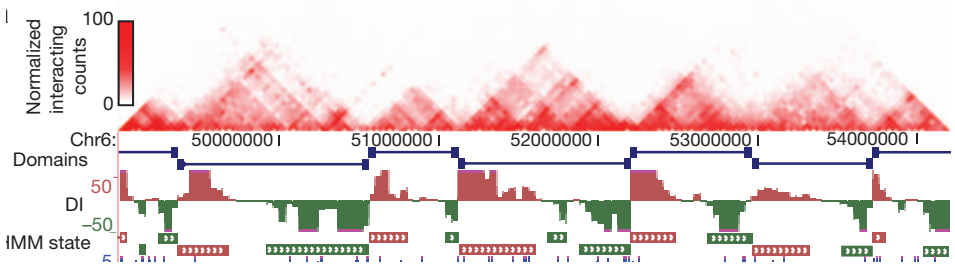
- Finding higher-order Topologically Associated Domains from Hi-C data
- Disease module identification
- Similarity network fusion for aggregating data types on a genomic scale

Genome is organized into multiple organizational units

- Chromosomal territories through inter-TAD interactions
- Compartments and sub-compartments
- Topologically associated domains (TADs) and sub-TADs

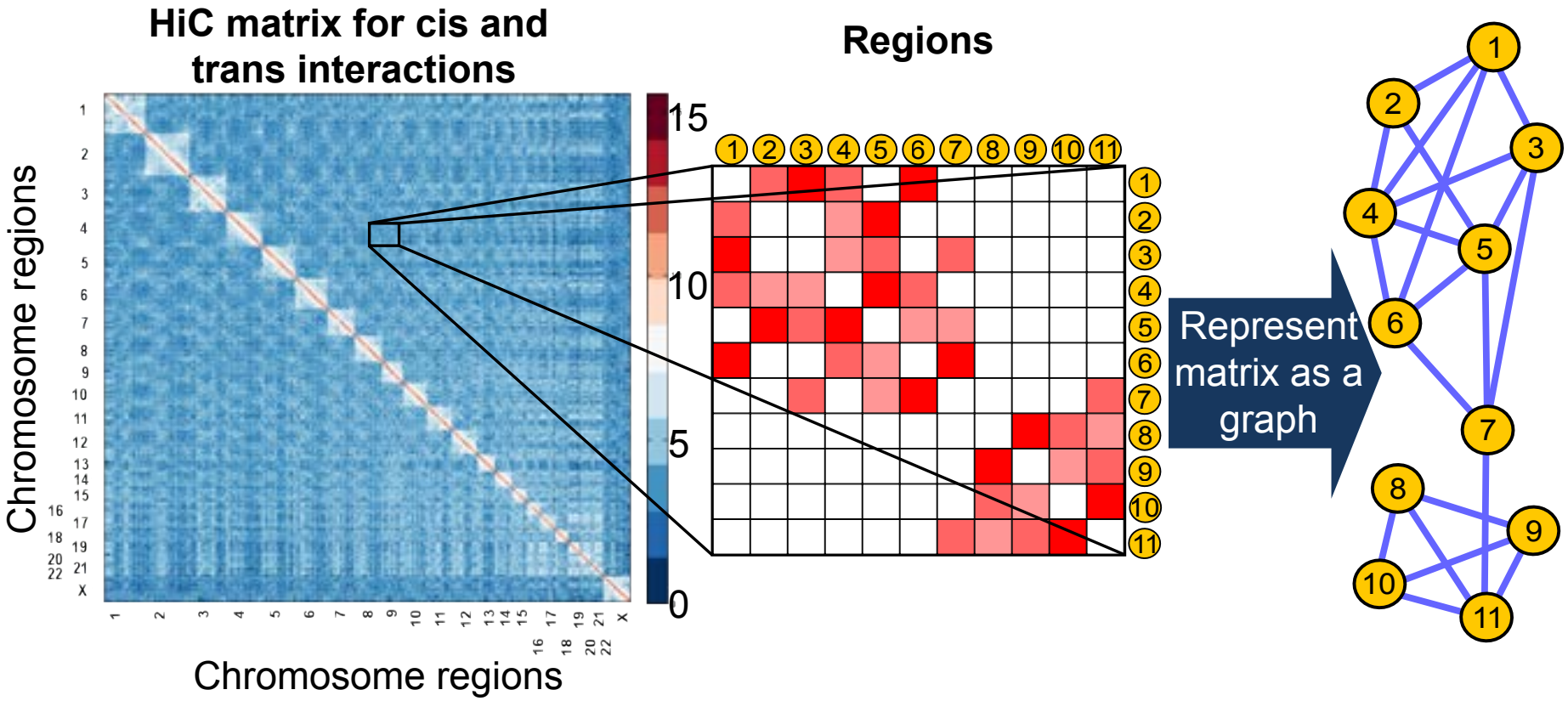




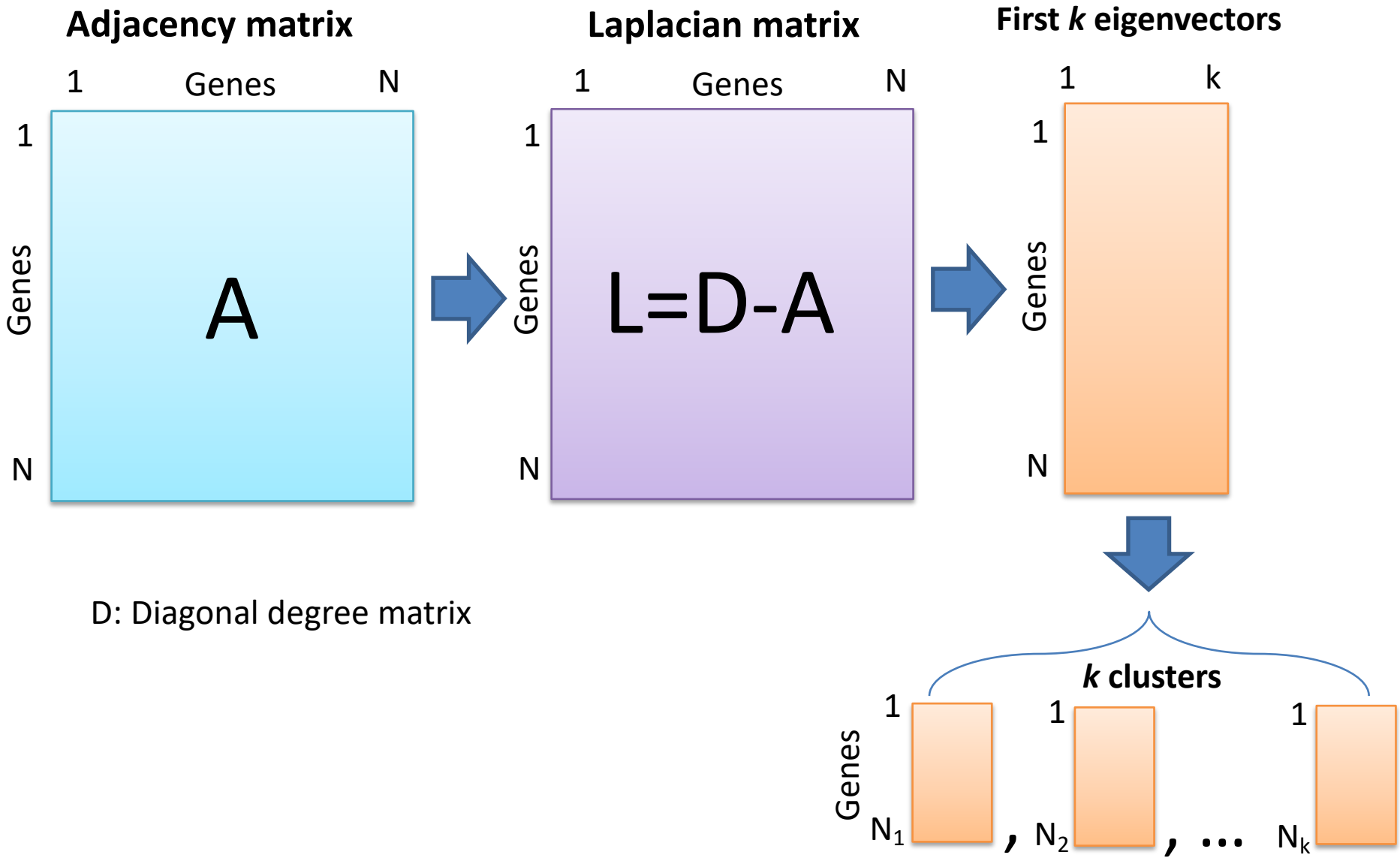


Lierberman et al 2009, Rao et 2014, Dixon et al 2012

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

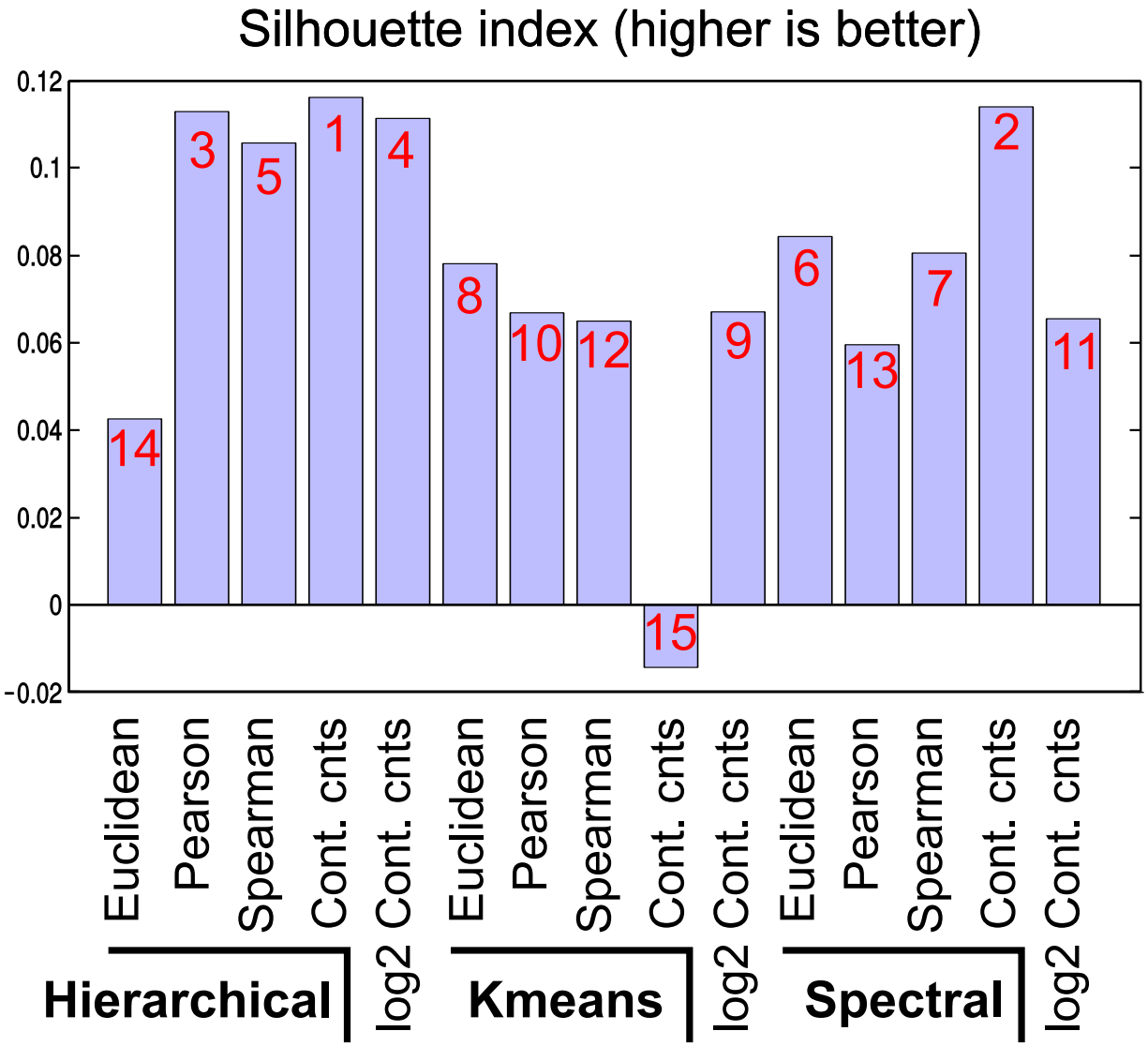


An overview of spectral clustering

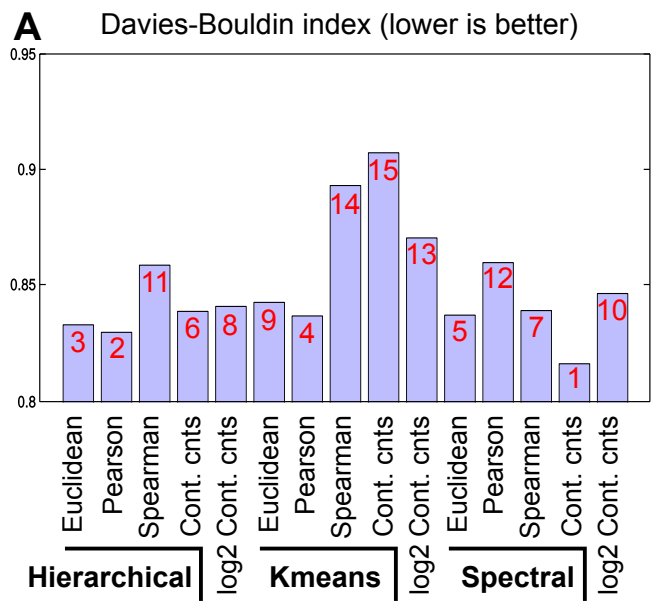


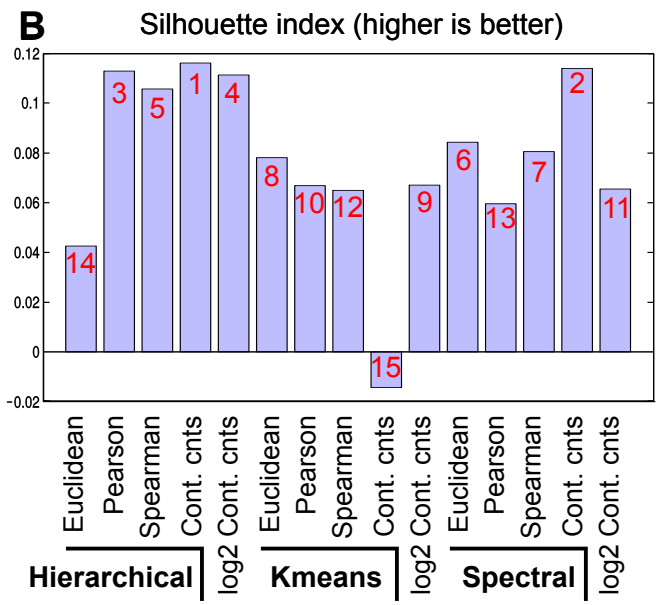
D: Diagonal degree matrix

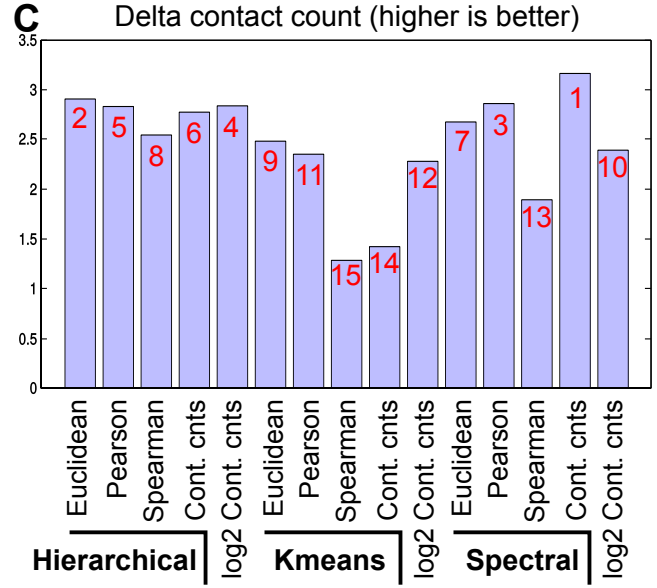
Does graph clustering help?

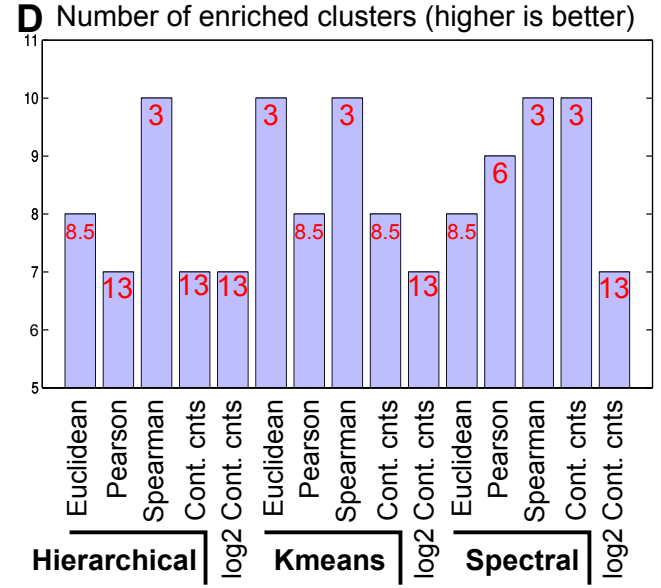


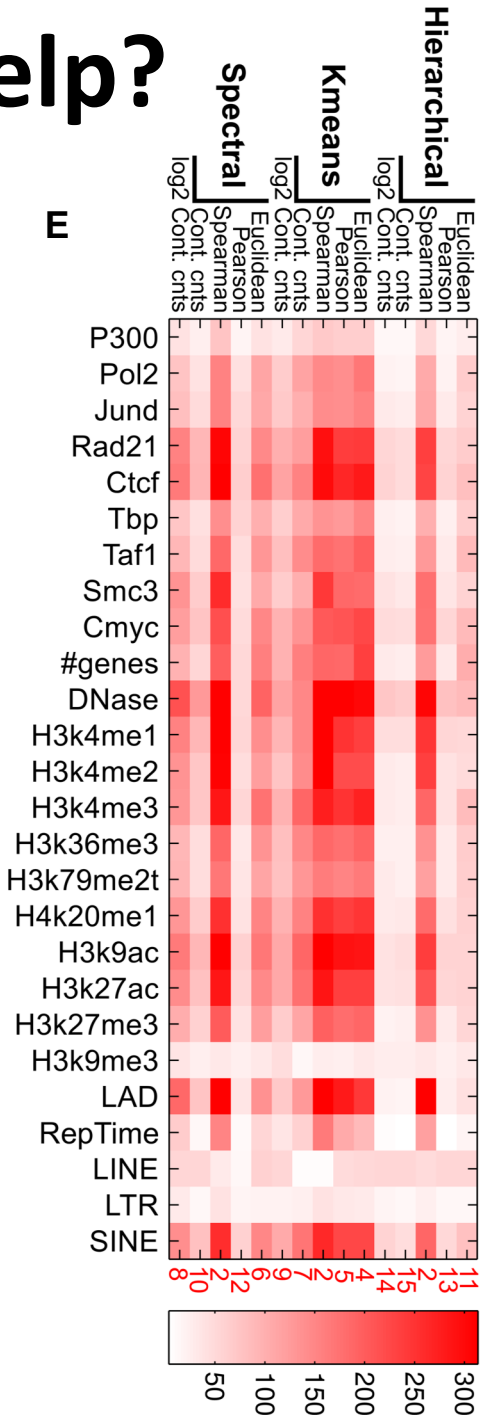
Does graph clustering help?



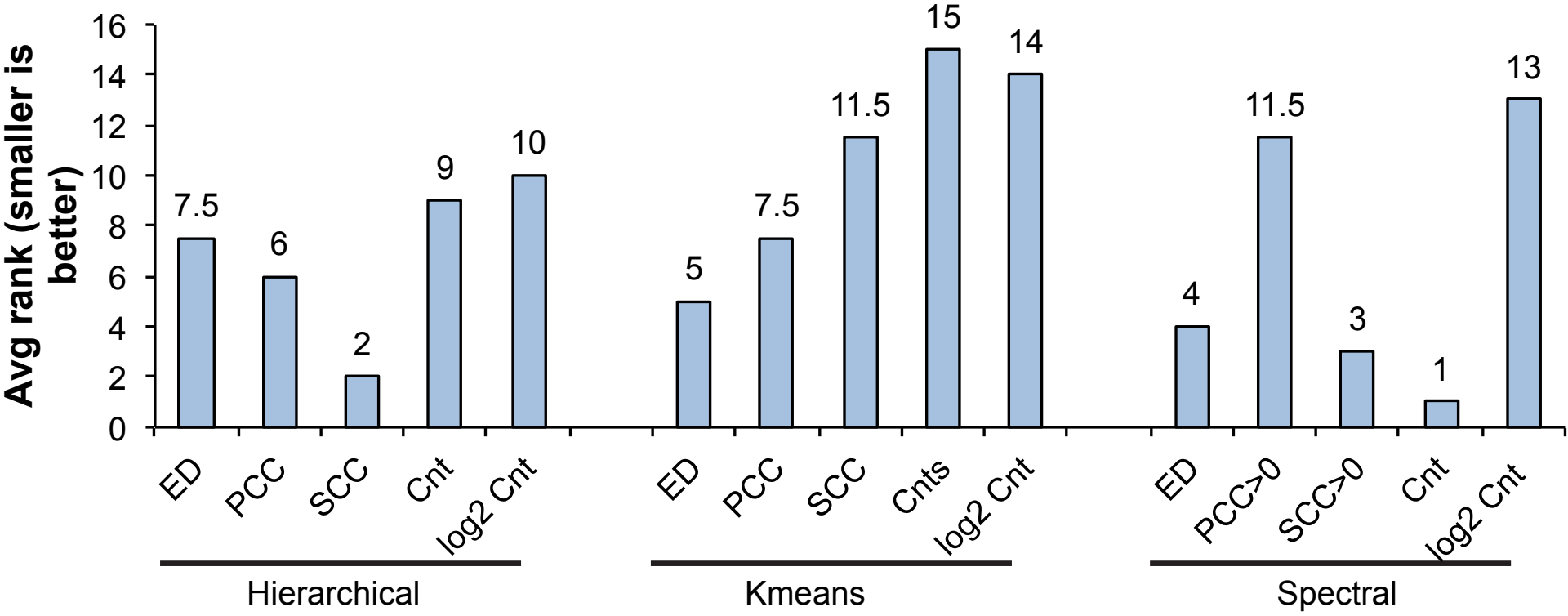






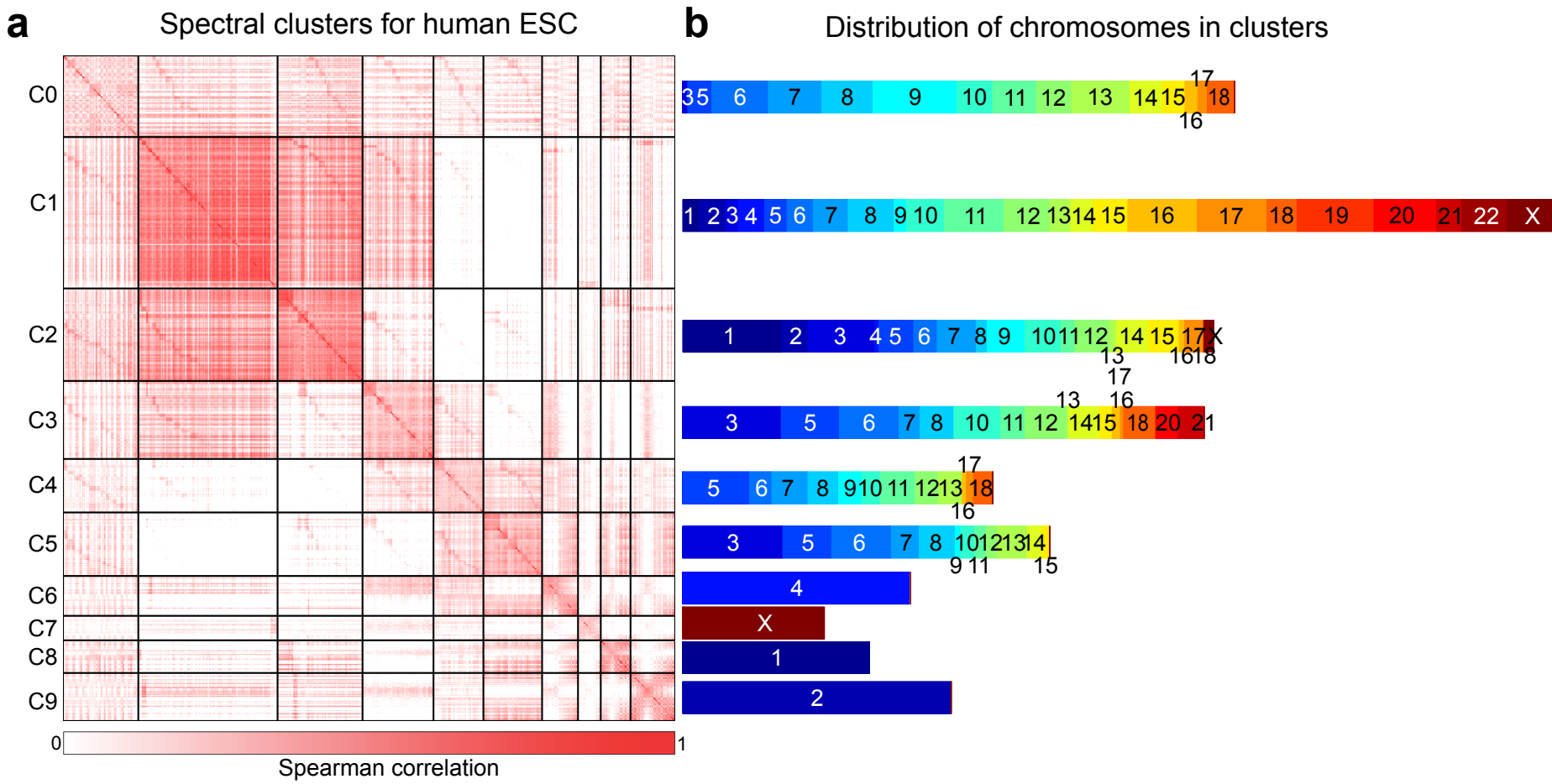


Does graph clustering help?

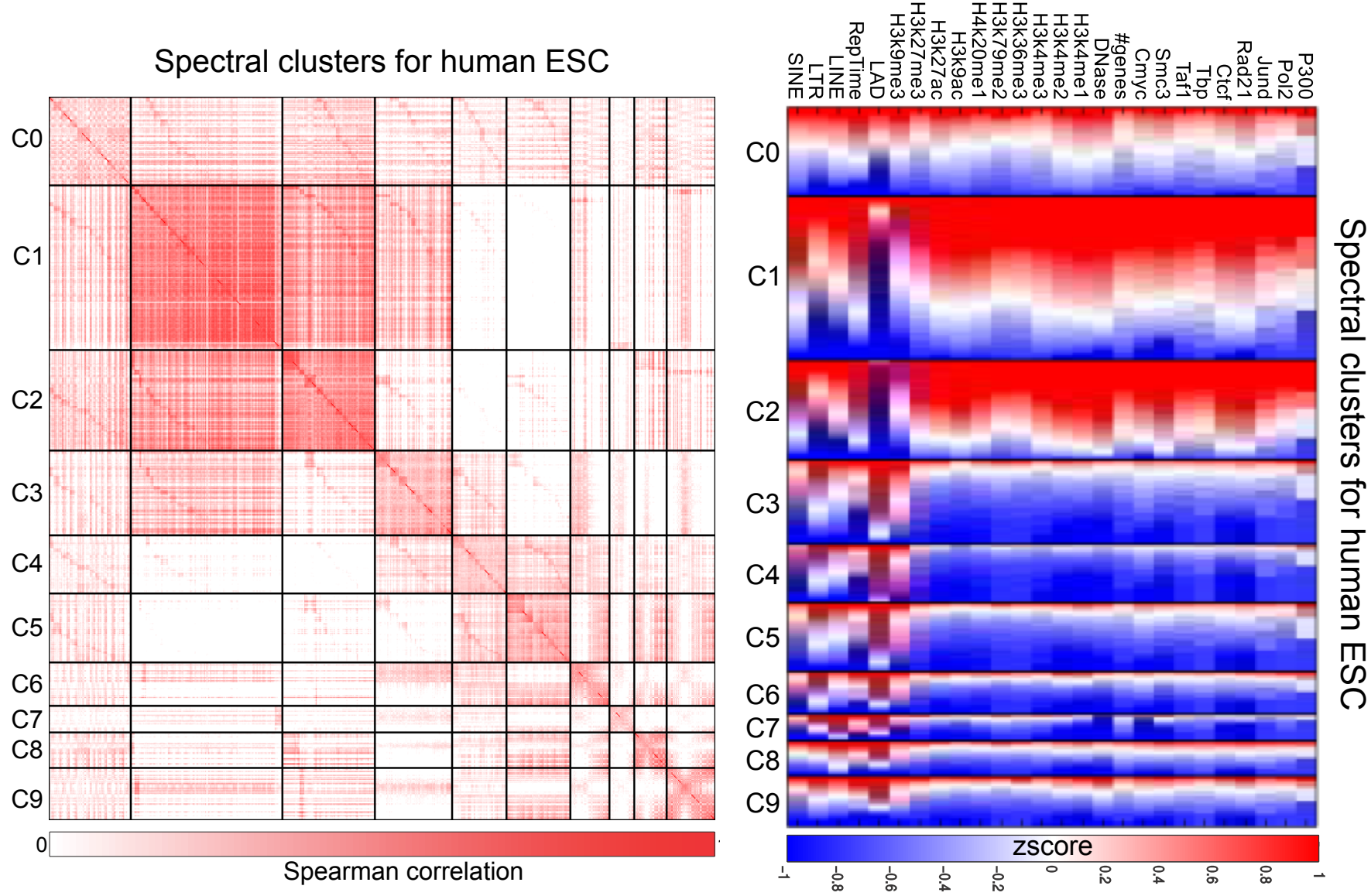


Spectral (graph) clustering methods tend to do better on different measures

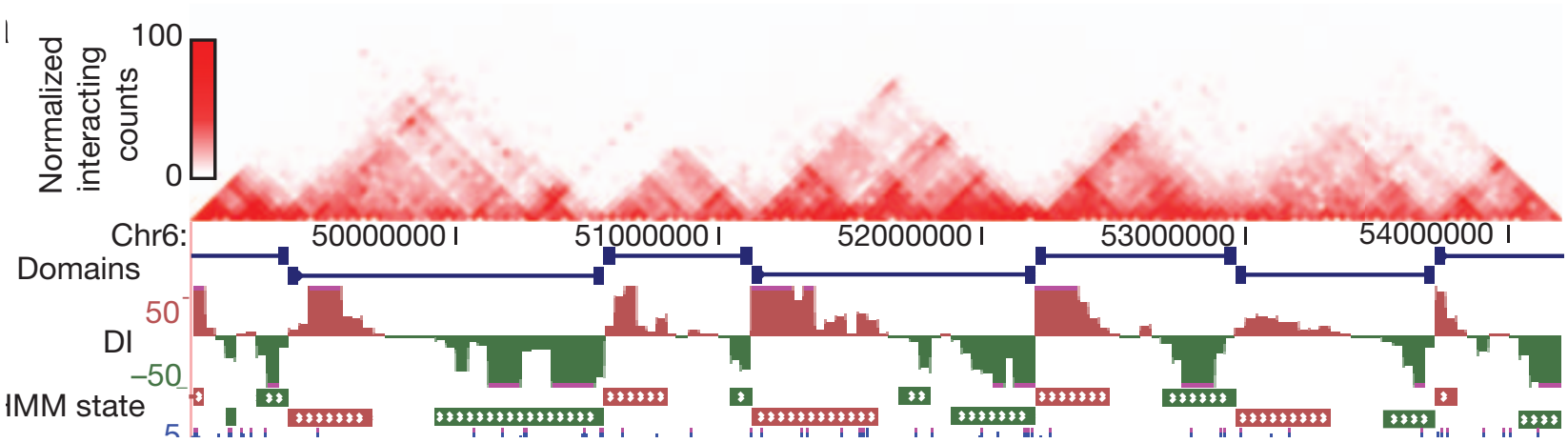
Spectral clustering of Hi-C data of human ESC



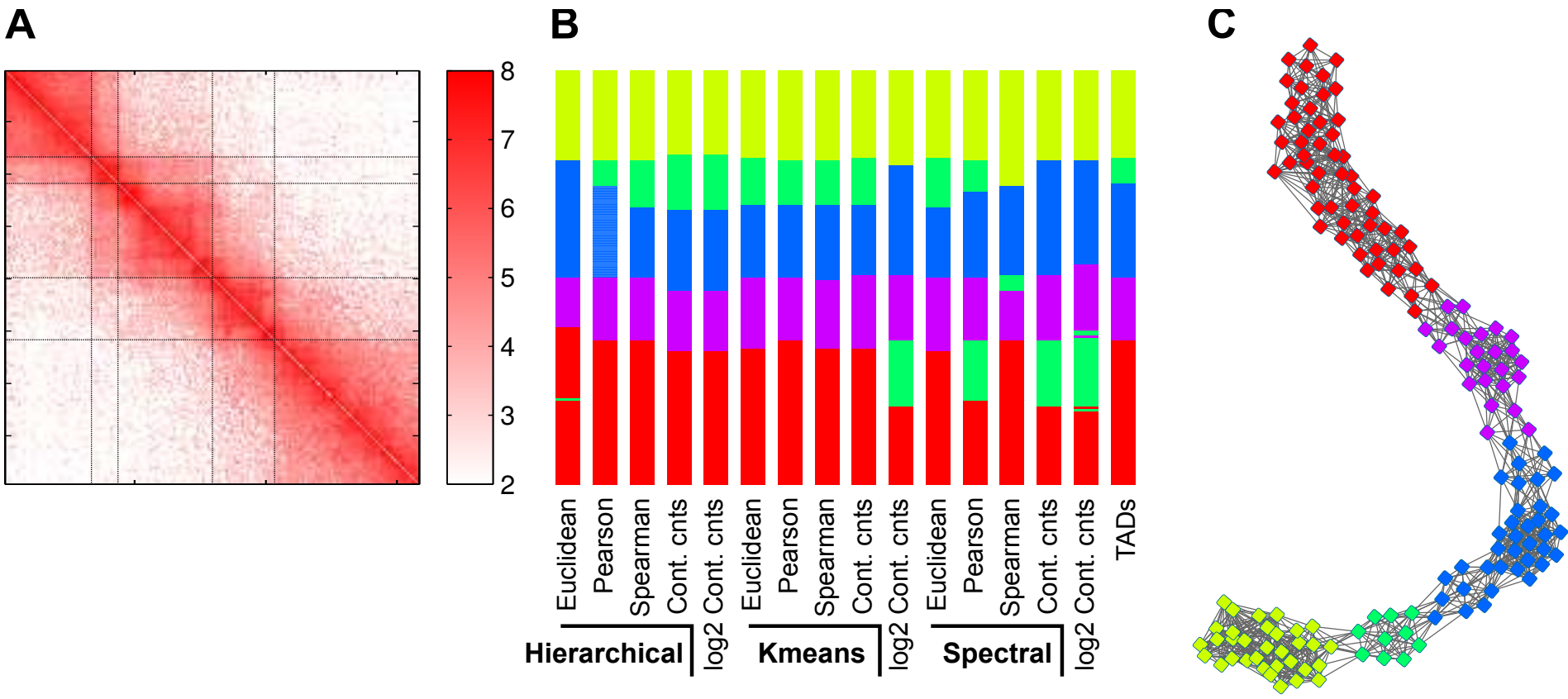
Two main types of chromatin interaction modules



Topologically associated domains



Graph clustering to find TADs



Application of spectral clustering

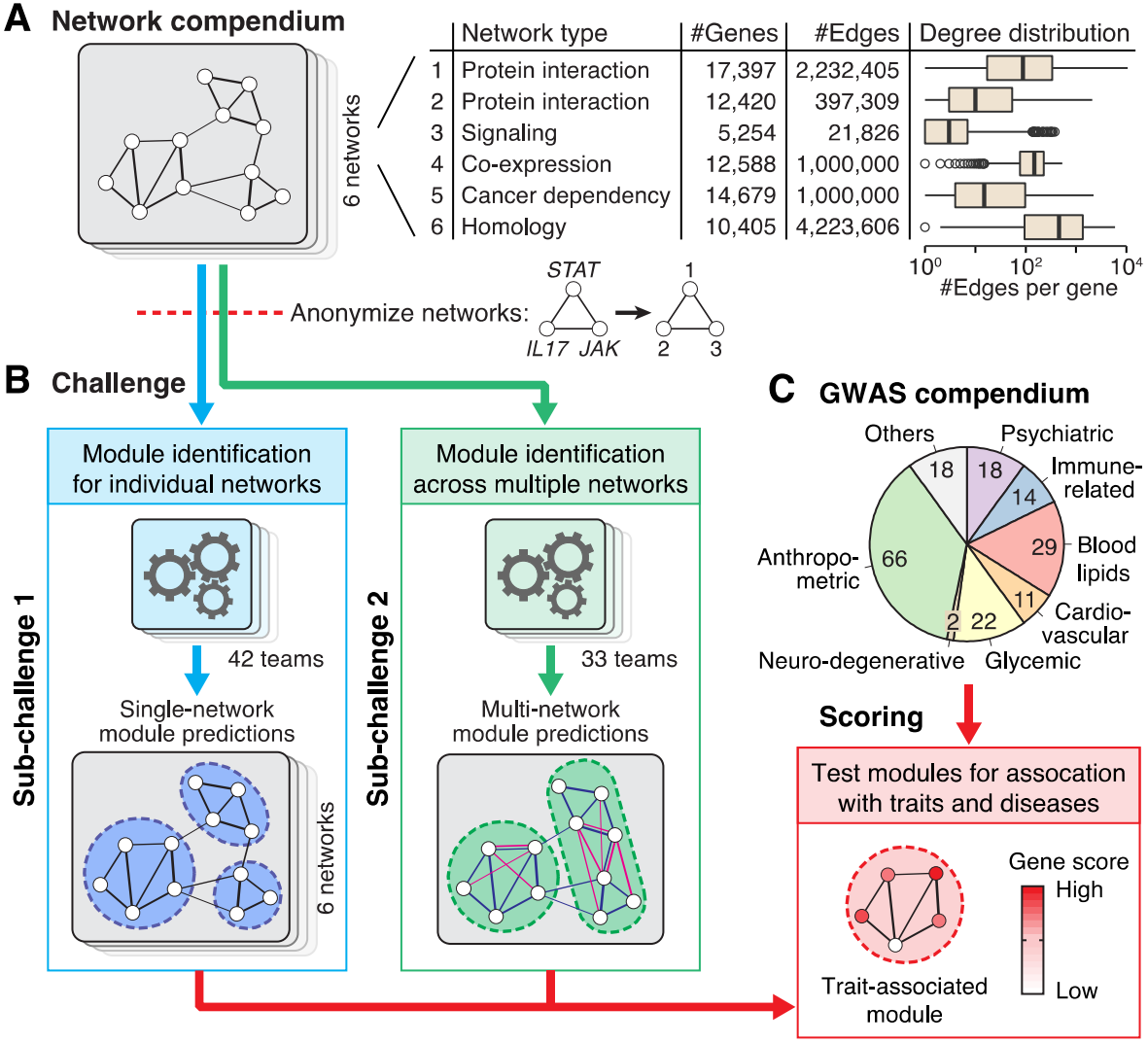
- Finding higher-order Topologically Associated Domains from Hi-C data
- Disease module identification
- Similarity network fusion for aggregating data types on a genomic scale

DREAM community challenge for module identification

- A community challenge to assess algorithms for module identification across diverse molecular networks
- Six different networks
- Sub challenge 1: predict modules within a single network
- Sub challenge 2: predict modules across multiple networks.
- Evaluation: how many modules are associated with GWAS traits.

Choodbar et al., 2018 Bioarxiv

Overview of the DREAM disease module identification challenge



Challenge organization

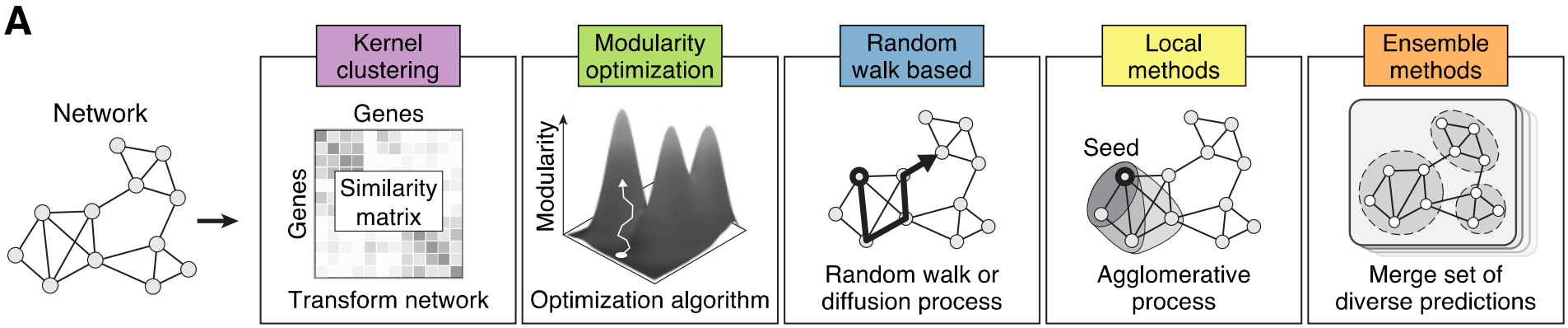
- Challenge was executed on Synapse
- Submissions accepted over a 2 month period where submitters could use benchmark data to assess and improve their predictions
- Final submissions were done on a separate GWAS dataset

Evaluation pipeline

- Six networks which were anonymized and given to challenge participants
- Consider modules of size 3-100 genes
- Assess modules based on GWAS association

Methods used

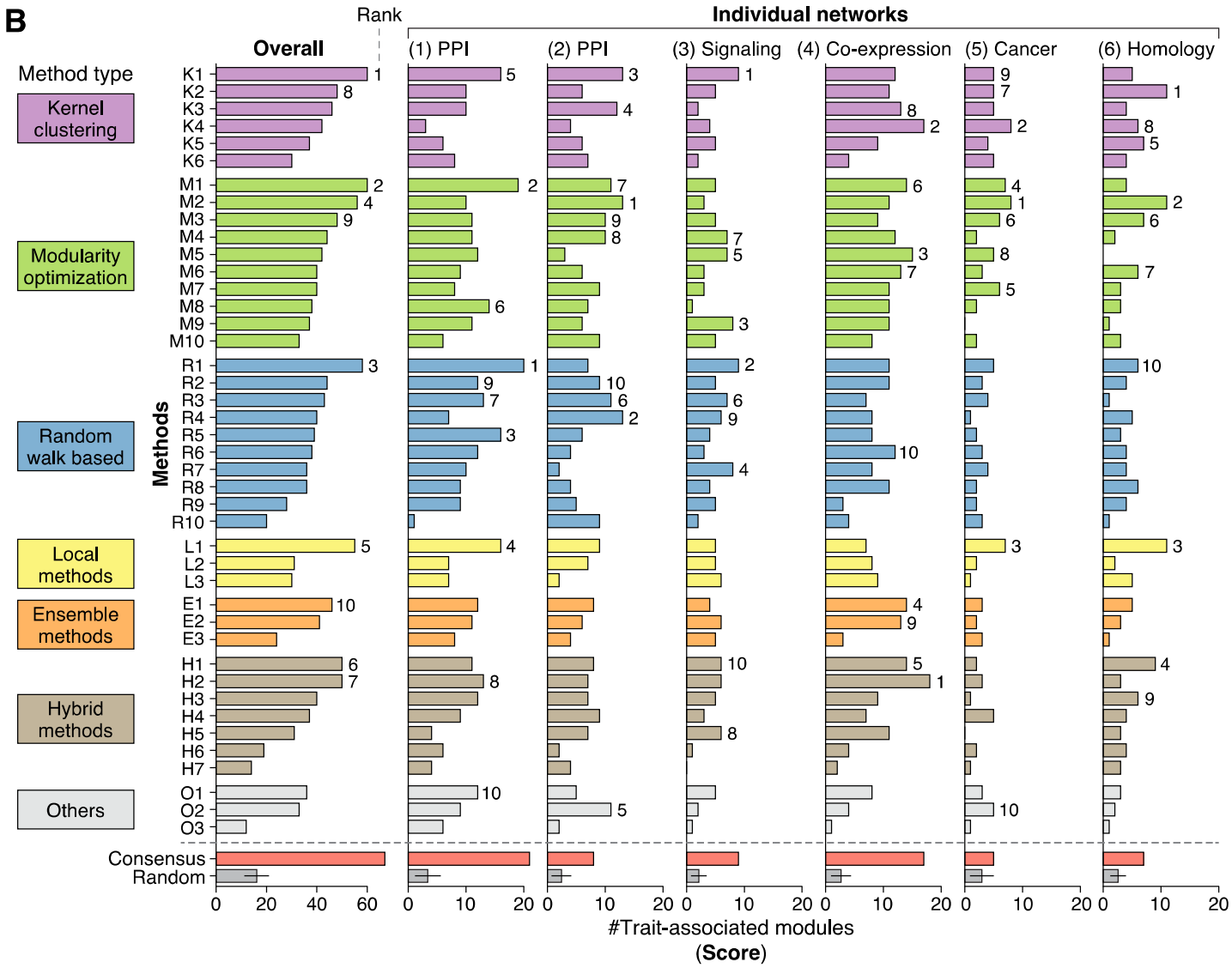
- 42 different methods from the following categories



Overview of results

The top teams used different approaches: the best performers (*K1*) developed a novel kernel approach leveraging a diffusion-based distance metric (Cao et al., 2013, 2014) and spectral clustering (Ng et al., 2001); the runner-up team (*M1*) extended different modularity optimization methods with a resistance parameter that controls the granularity of modules (Arenas et al., 2008); and the third-ranking team (*R1*) used a random-walk method based on multi-level Markov clustering with locally adaptive granularity to balance module sizes (Satuluri et al., 2010). Interestingly, teams employing the widely-used Weighted Gene Co-expression Network Analysis tool (WGCNA) (Langfelder and Horvath, 2008), which relies on hierarchical clustering to detect modules, did not perform competitively in this challenge (rank 35, 37 and 41).

Overview of results



Top performing method

- Use diffusion state distance (DSD) for each pair of vertices
- Convert into a similarity by passing it through the Gaussian kernel
- Apply spectral clustering

Other takeaways from disease module identification

- Co-expression and Protein-protein interaction network based modules were most informative
- Top methods covered different categories
  - But spectral clustering based methods worked best.
- Determining the right resolution can impact the results

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

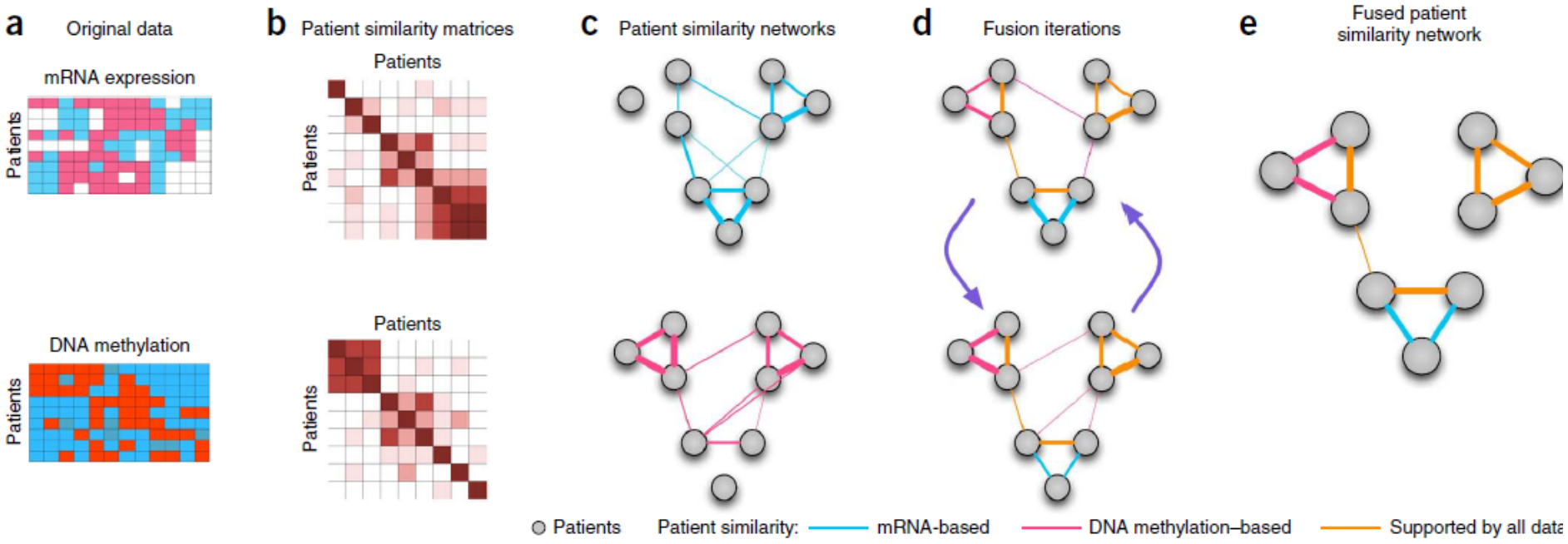
Similarity network fusion for aggregating data types on a genomic scale

- 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

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

Euclidean distance

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*

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

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

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

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

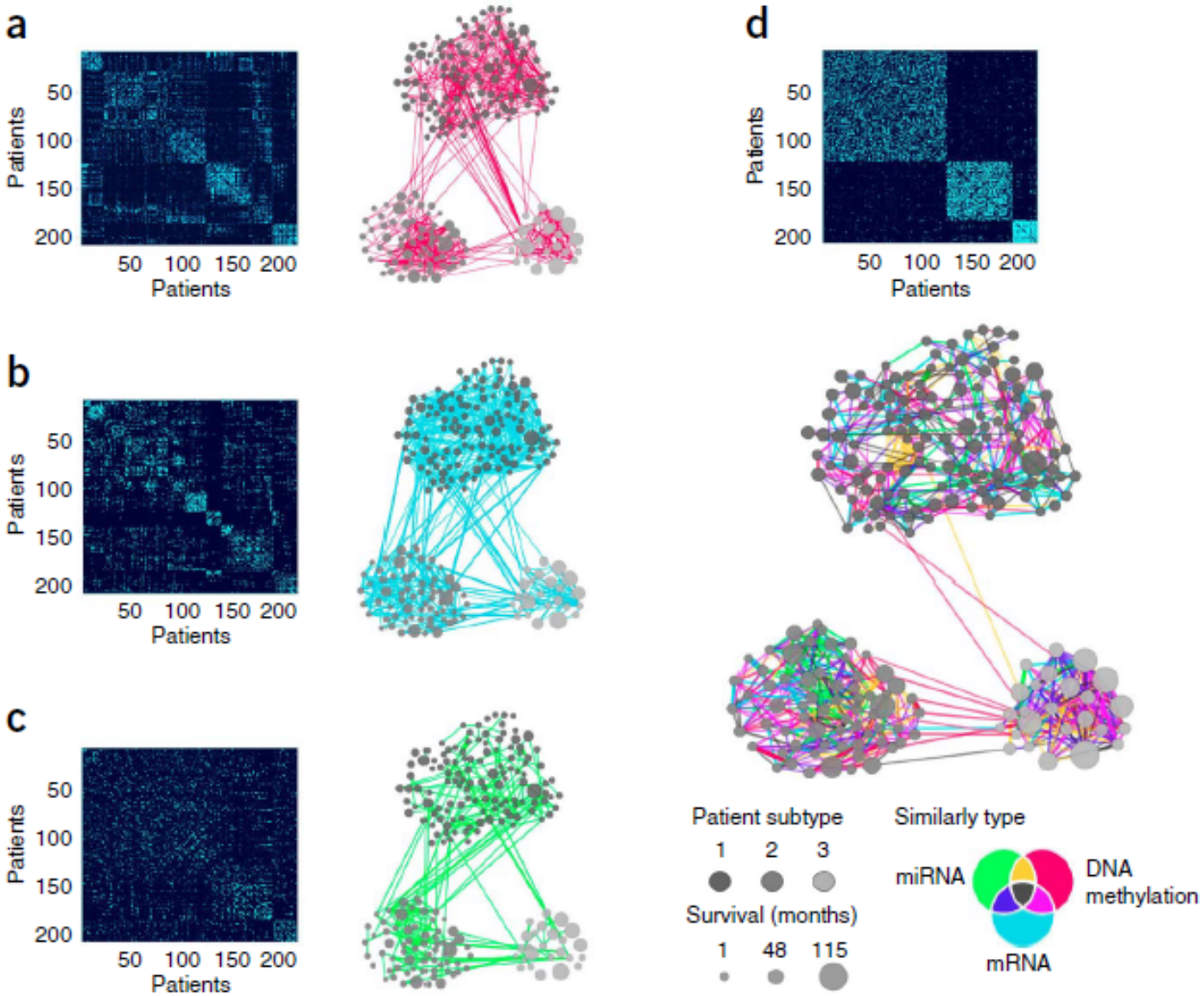
- Contradicting information about subtypes depending upon the type of data used
- Glioblastoma dataset
- Three data types among 215 patients
    - DNA methylation (1491 genes)
    - mRNA (12,042 genes)
    - miRNA (534 miRNAs)

SNF application to GBM identifies 3 subtypes

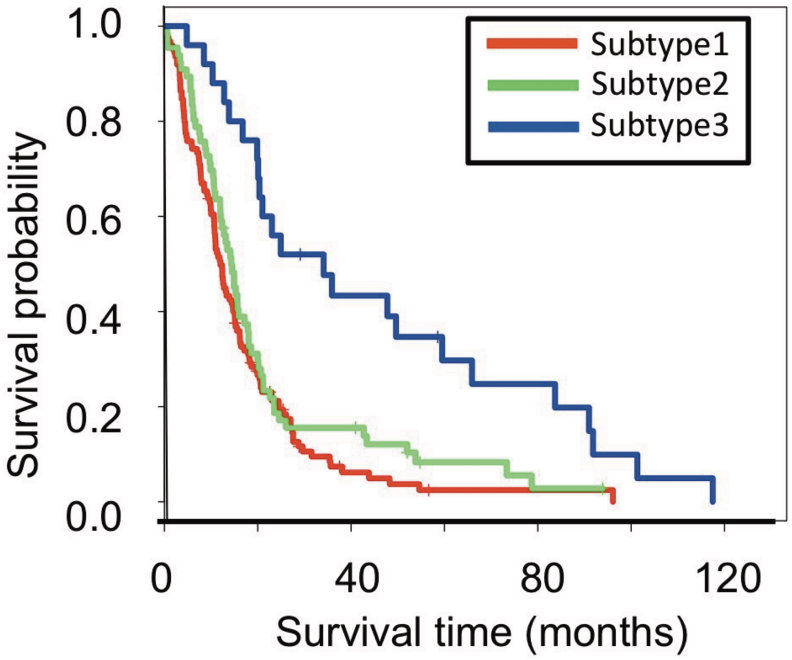
DNA methylation

mRNA expression

miRNA expression

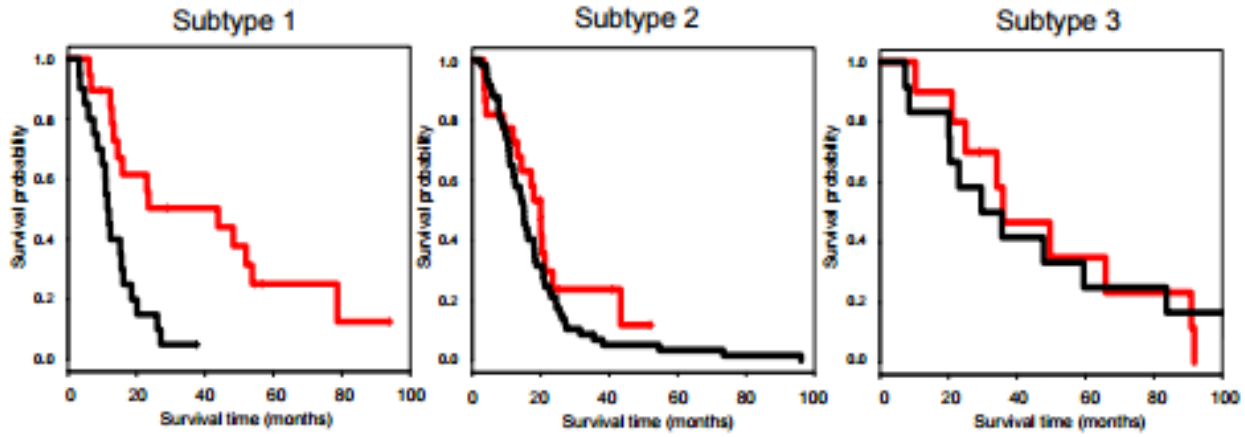


Validation of SNF identified subtypes



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

Red: treated
Black: untreated.



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

- von Luxburg, Ulrike. 2007. “A Tutorial on Spectral Clustering.” *Statistics and Computing* 17(4): 395–416. http://dx.doi.org/10.1007/s11222-007-9033-z.
- Wang, Bo et al. 2014. “Similarity Network Fusion for Aggregating Data Types on a Genomic Scale.” *Nature methods* 11(3): 333–337. http://dx.doi.org/10.1038/nmeth.2810.
- Fotuhi Siahpirani, Alireza, Ferhat Ay, and Sushmita Roy. 2016. “A Multi-Task Graph-Clustering Approach for Chromosome Conformation Capture Data Sets Identifies Conserved Modules of Chromosomal Interactions.” *Genome Biology* 17(1). http://dx.doi.org/10.1186/s13059-016-0962-8.
- Choobdar, Sarvenaz et al. 2018. “Open Community Challenge Reveals Molecular Network Modules with Key Roles in Diseases.” *bioRxiv*: 265553. https://www.biorxiv.org/content/early/2018/02/15/265553(November 7, 2018).