Network Theory

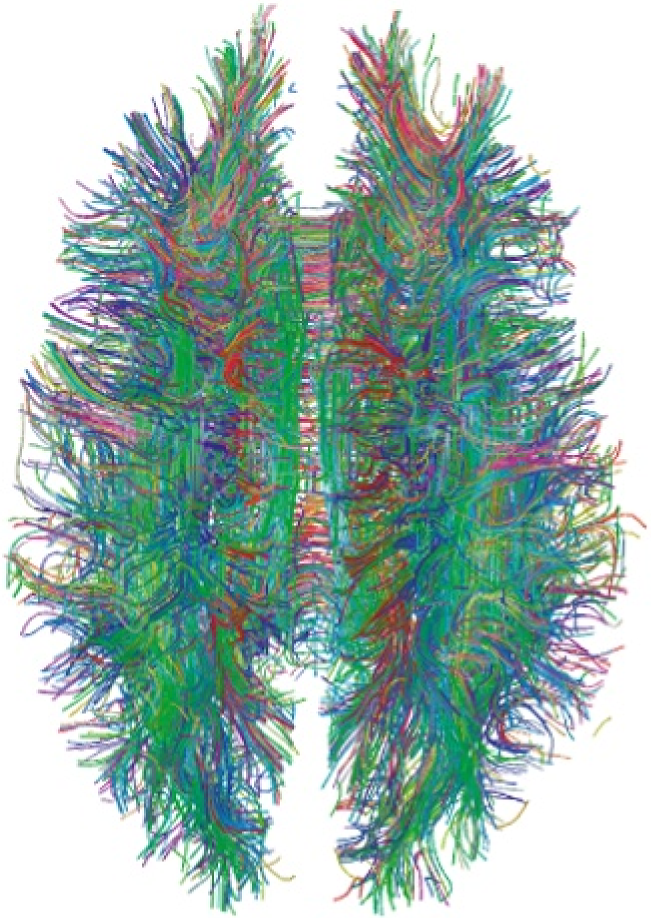
- Graph metrics for undirected binary graphs
  - ➢ degree, hubs, centrality
  - ➢ communities
  - ➢ scale-free and small-world graphs
- Directed and weighted graphs
- Application to synthetic and real datasets
  - ➢ Python library: networkx

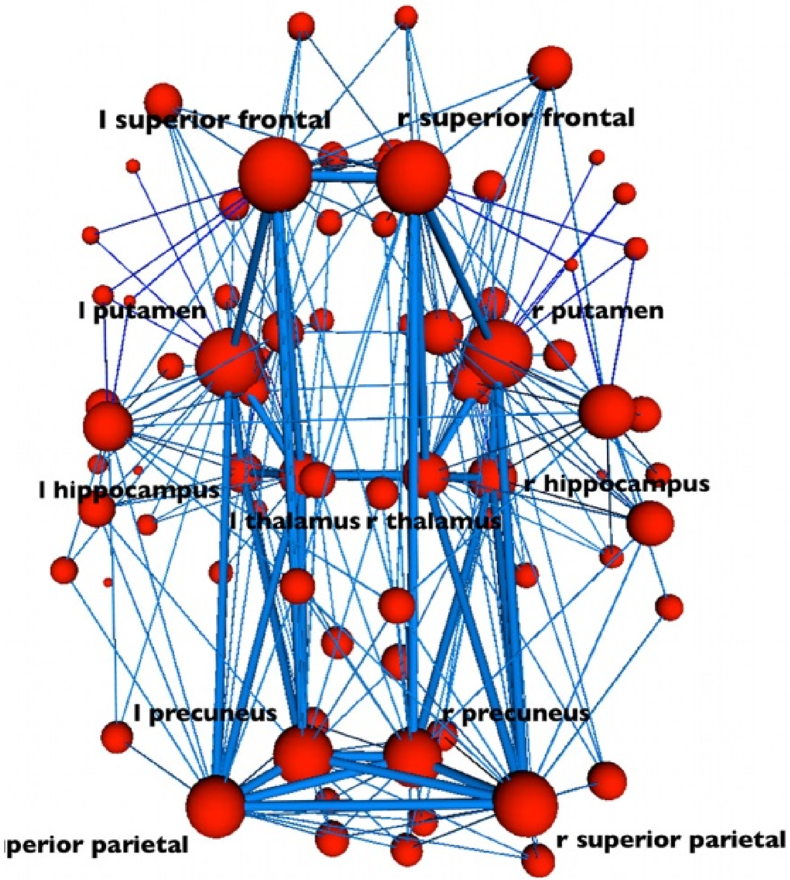
References

- Articles and reviews:
  - Krendl et al. (2022) *SCAN*: networks in social and cognitive neuroscience
  - Douw et al. (2023) *Netw Neurosci*: clinical perspectives
  - van den Heuvel and Sporns (2011) *J Neurosci*: rich-club in structural connectome

- To go further for application to fMRI dynamics:
  - Gilson et al. (2020) *Neuroimage*: model-based analysis of whole-brain dynamics based on fMRI data

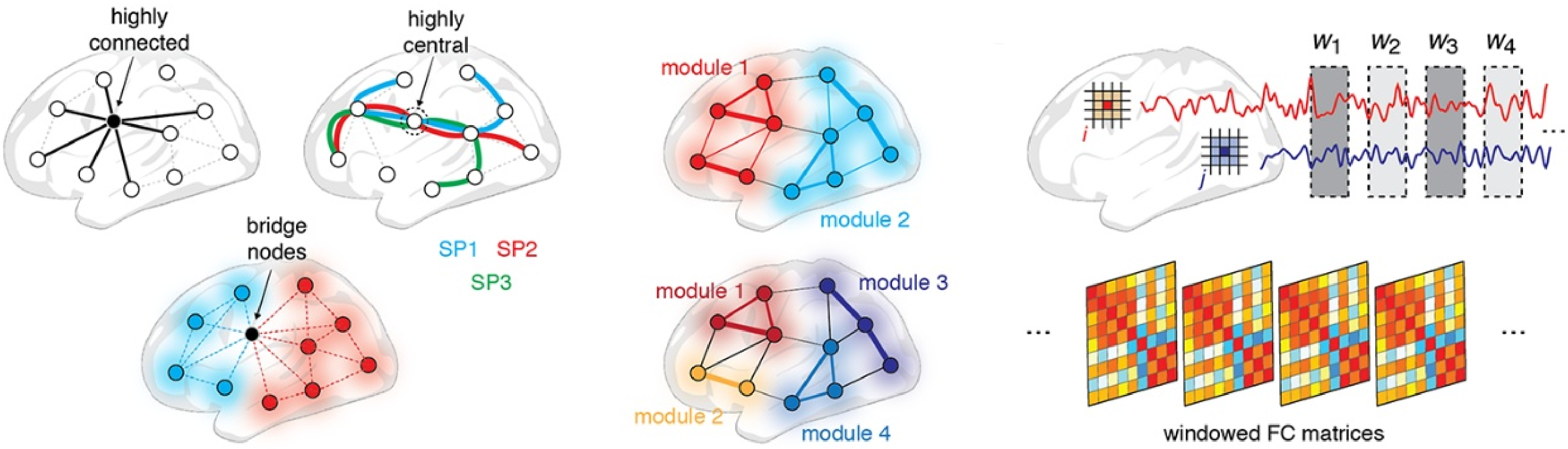
Rich-Club in Structural Connectome (White-Matter Fibers)





Hagman et al. (2008) *PLoS Biol*; van den Heuvel et al. (2011) *J Neurosci*

Network Analysis of Functional MRI



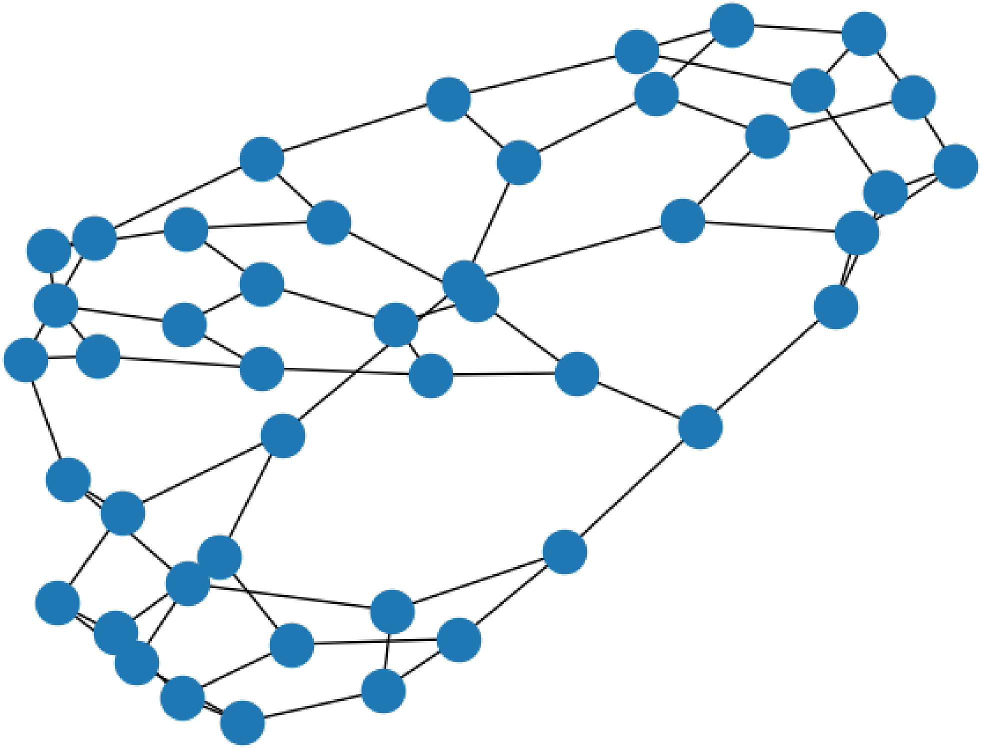
- Links reflect correlations of fMRI signals
- How to interpret network measures?

Lrendl et al. (2022) *SCAN*

Class 2: Network Theory

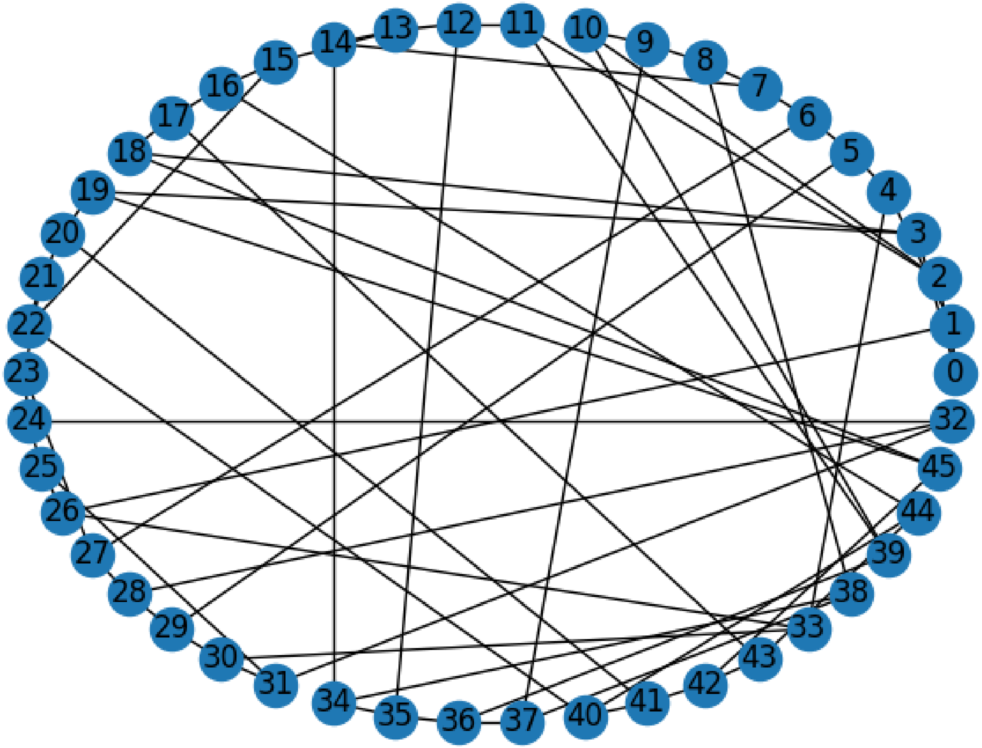
- **Graph metrics for undirected binary graphs**
  - **degree, hubs, centrality**
  - communities
  - scale-free and small-world graphs
- Directed and weighted graphs
- Application to synthetic and real datasets
  - Python library: networkx

What is a Network/Graph?



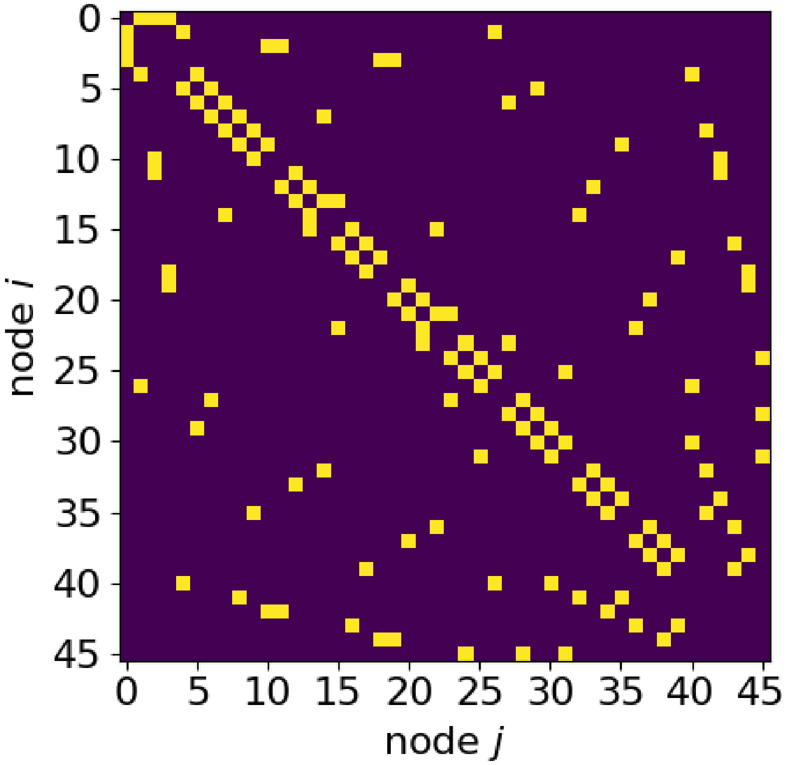
- Nodes
- Edges

What is a Network/Graph?



- Nodes
- Edges

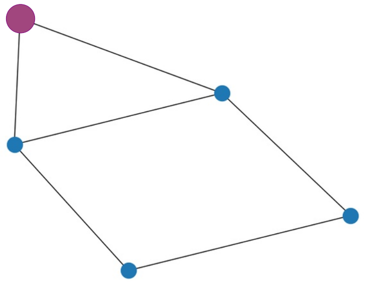
What is a Network/Graph?



- Nodes *i*
- Edges *A_ij*
- Adjacency matrix *A*

How important is a node?

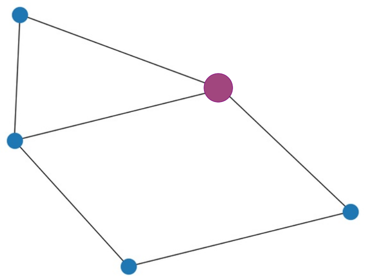
- Degree of a node: number of connections
  - a node with high degree is called a ‘hub’



degree = 2

How important is a node?

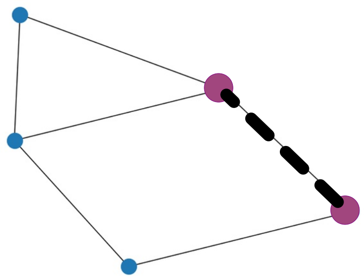
- Degree of a node: number of connections
  - a node with high degree is called a ‘hub’



degree = 3

How important is a node?

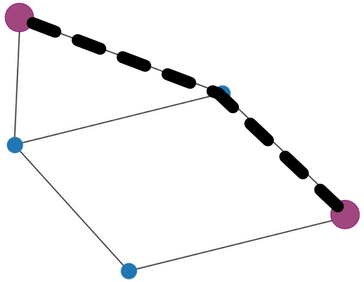
- Degree of a node: number of connections
  - a node with high degree is called a ‘hub’
- Shortest path: for a given pair of nodes, the shortest path is the minimum number of links between them (if they are directly or indirectly connected)
  - the distribution of shortest paths shows how “easy” it is to “travel” in the network



shortest path length = 1

How important is a node?

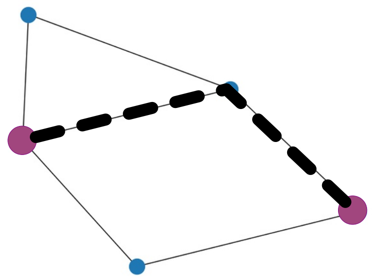
- Degree of a node: number of connections
  - a node with high degree is called a ‘hub’
- Shortest path: for a given pair of nodes, the shortest path is the minimum number of links between them (if they are directly or indirectly connected)
  - the distribution of shortest paths shows how “easy” it is to “travel” in the network



shortest path length = 2

How important is a node?

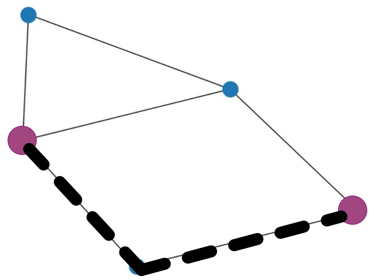
- Degree of a node: number of connections
  - a node with high degree is called a ‘hub’
- Shortest path: for a given pair of nodes, the shortest path is the minimum number of links between them (if they are directly or indirectly connected)
  - the distribution of shortest paths shows how “easy” it is to “travel” in the network



shortest path length = 2

How important is a node?

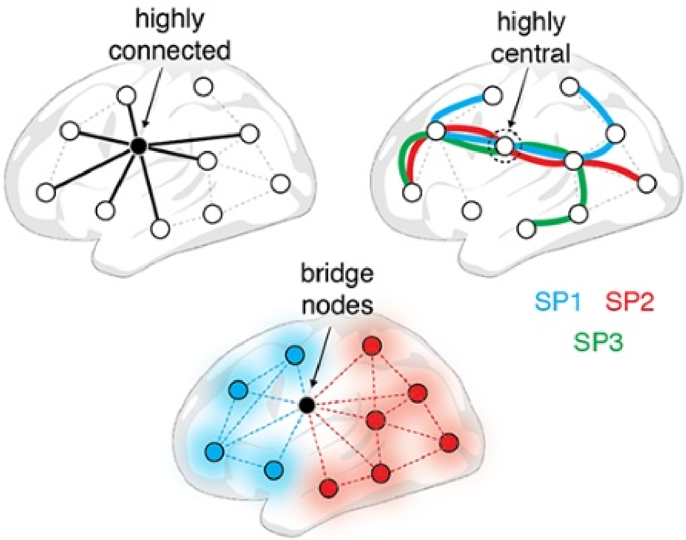
- Degree of a node: number of connections
  - a node with high degree is called a ‘hub’
- Shortest path: for a given pair of nodes, the shortest path is the minimum number of links between them (if they are directly or indirectly connected)
  - the distribution of shortest paths shows how “easy” it is to “travel” in the network



shortest path length = 2

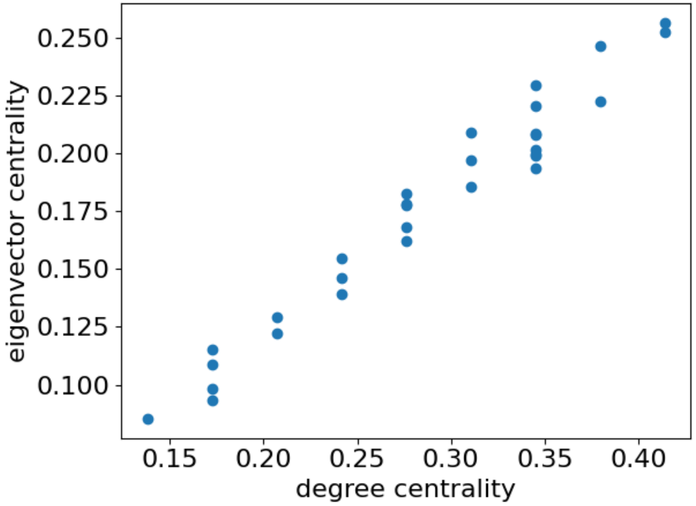
How important is a node?

- Centrality: one concept, many flavors
  - degree centrality for a node is the fraction of neighbors (w.r.t. all remaining nodes)
  - betweenness centrality of a node is the sum of the fraction of all-pairs' shortest paths that pass through
  - eigenvector centrality comes from the spectral analysis of the adjacency matrix



How important is a node?

- Centrality: one concept, many flavors
  - degree centrality for a node is the fraction of neighbors (w.r.t. all remaining nodes)
  - betweenness centrality of a node is the sum of the fraction of all-pairs' shortest paths that pass through
  - eigenvector centrality comes from the spectral analysis of the adjacency matrix



Mathematical Formulation based on Adjacency Matrix

- Adjacency matrix: A_{ij}
- Degree: $d_i = \sum_j A_{ij}$ so in matrix form: $d = A\, e$ with $e = (1, \ldots, 1)^T$

Mathematical Formulation based on Adjacency Matrix

- Adjacency matrix: A_{ij}
- Degree: $d_i = \sum_j A_{ij}$ so in matrix form: $d = A\,e$ with $e = (1, \ldots, 1)^T$
- Number of paths from *j* to *i* of length *l* $\left(A^l\right)_{ij}$

Mathematical Formulation based on Adjacency Matrix

- Adjacency matrix: A_{ij}
- Degree: $d_i = \sum_j A_{ij}$ so in matrix form: $d = A\,e$ with $e = (1, \ldots, 1)^T$
- Number of paths from *j* to *i* of length *l* $(A^l)_{ij}$ $(A^2)_{ij} = \sum_k A_{ik} A_{kj}$

Mathematical Formulation based on Adjacency Matrix

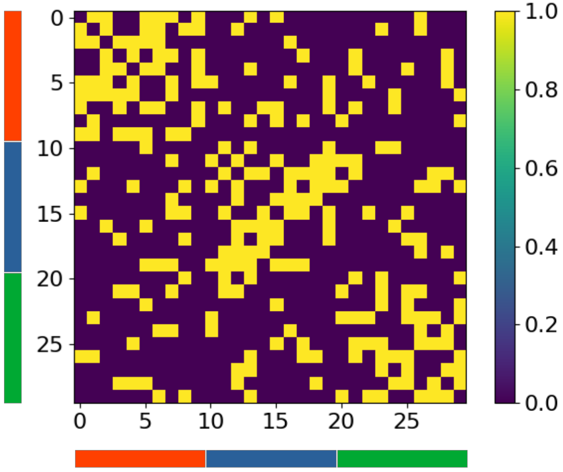
- Adjacency matrix: A_{ij}
- Degree: $d_i = \sum_j A_{ij}$ so in matrix form: $d = A e$ with $e = (1, \ldots, 1)^T$
- Number of paths from *j* to *i* of length *l* $(A^l)_{ij}$
- Dominating eigenvector and eigenvalue: $A v = \lambda v$
  - ➢ λ with largest real part among all eigenvalues
  - ➢ eigenvector centrality $|v_i|$

Class 2: Network Theory

- **Graph metrics for undirected binary graphs**
  - degree, hubs, centrality
  - **communities**
  - scale-free and small-world graphs
- Directed and weighted graphs
- Application to synthetic and real datasets
  - Python library: networkx

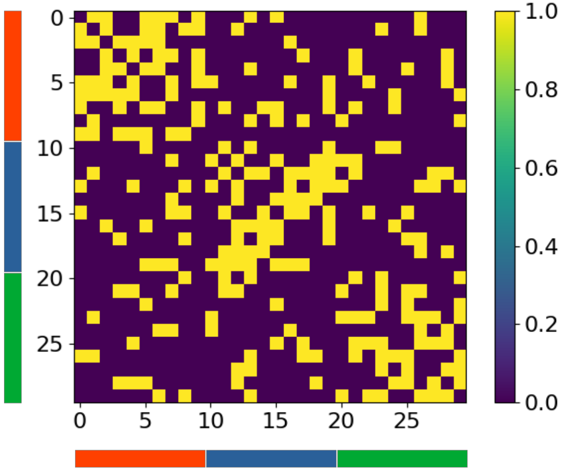
Communities

- Groups of nodes that are densely connected
- Detection using metrics that quantify within-group density compared to outside
- How likely are two nodes to be connected?



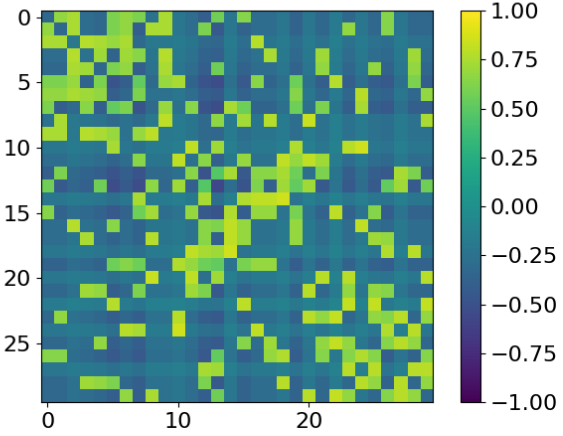
Communities

- Groups of nodes that are densely connected
- Detection using metrics that quantify within-group density compared to outside
- How likely are two nodes to be connected?



modularity matrix

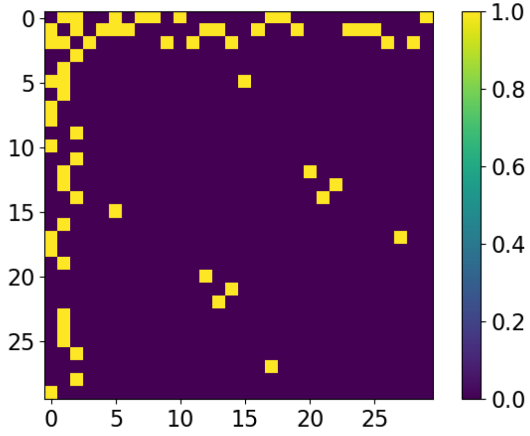
$$A_{ij} - A_{ij}^{ref}$$



$$A_{ij}^{ref} = \frac{d_i d_j}{\sum_k d_k}$$ ref calculated from degree d_i

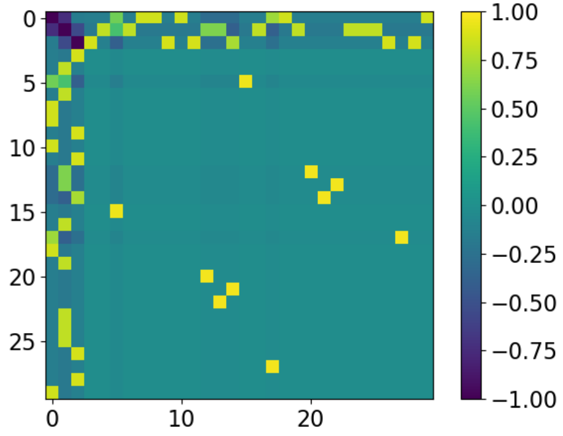
Communities

- Groups of nodes that are densely connected
- Detection using metrics that quantify within-group density compared to outside
- How likely are two nodes to be connected?



modularity matrix

$$A_{ij} - A_{ij}^{ref}$$



$$A_{ij}^{ref} = \frac{d_i d_j}{\sum_k d_k}$$

ref calculated from degree d_i

Greedy Algorithm based on Modularity

- Modularity to assess quality of community grouping (C_1, C_2, ...)

$$Q=\sum_{i,j\in C_l} A_{ij}-\frac{d_i d_j}{\sum_k d_k}$$

connection probability
“by chance”

- Start with singletons of nodes $\{i\},\{j\},\ldots$
- Check nodes that are “strongly connected” $\Delta Q=A_{ij}-\frac{d_i d_j}{\sum_k d_k}$
- Iterative process to merge subgroups $\{i\},\{j\}\rightarrow\{i,j\}$
- Repeat so long as Q increases
  - list of communities: $\{0,2,5,\ldots\},\{1,3,4,\ldots\},\ldots$

Girvan & Newman (2002) *PNAS*, Newman (2006) *Phys Rev E*

Class 2: Network Theory

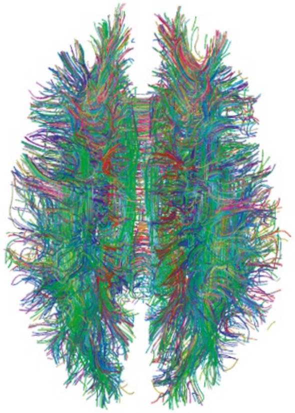
- **Graph metrics for undirected binary graphs**
  - degree, hubs, centrality
  - communities
  - **scale-free and small-world graphs**

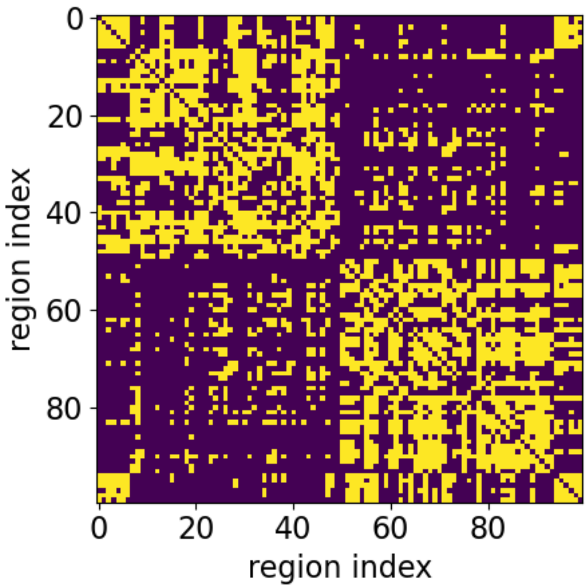
- Directed and weighted graphs

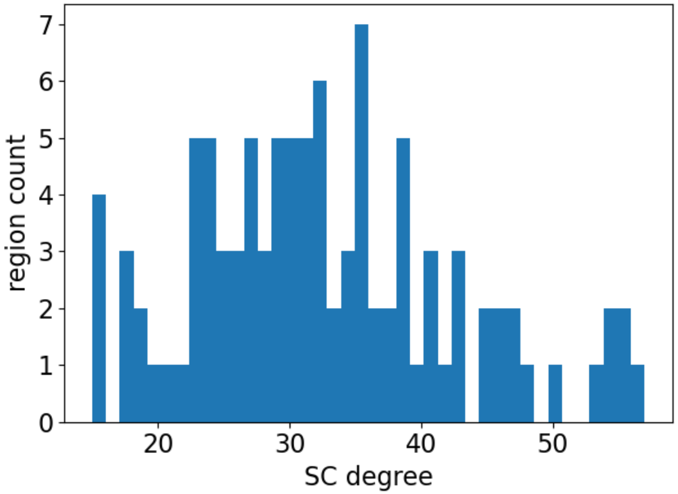
- Application to synthetic and real datasets
  - Python library: networkx

Synthetic Models for Real Data

- Connectivity in real data? Example of structural connectome
- Distribution of degree across cortical regions?

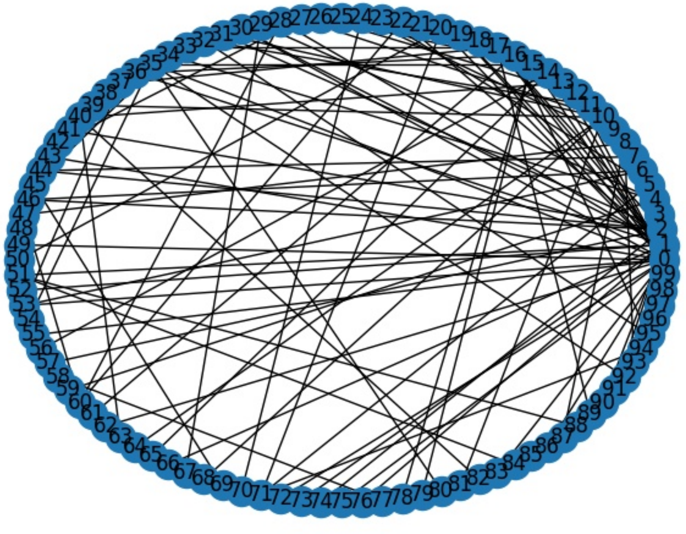


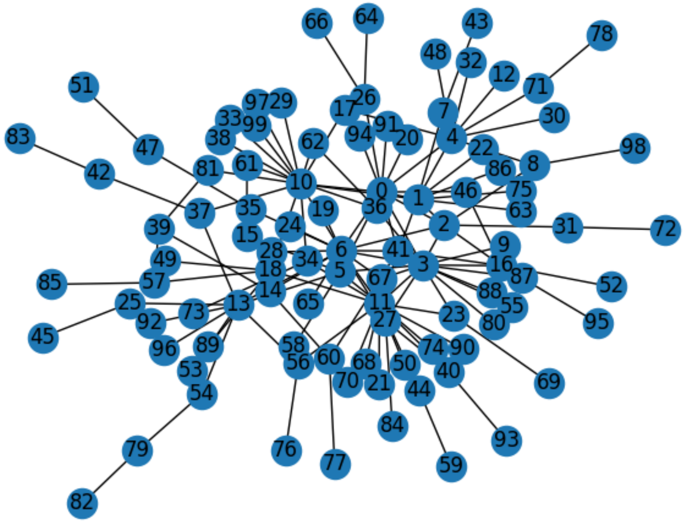




Scale-Free Networks

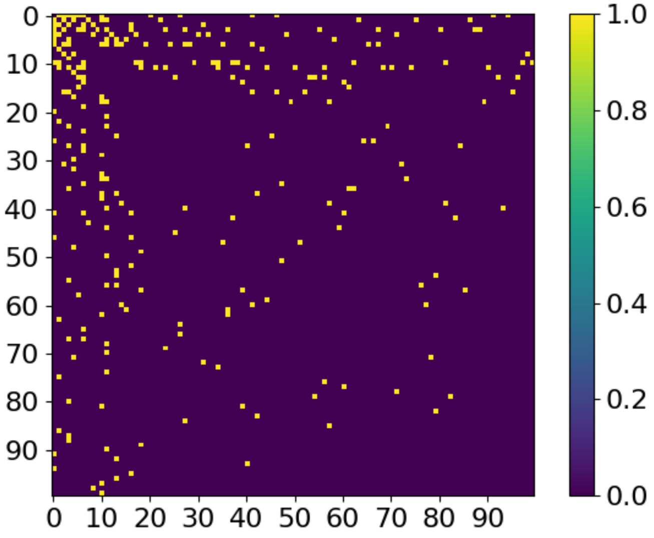
- Construction with preferential to given nodes (hubs)

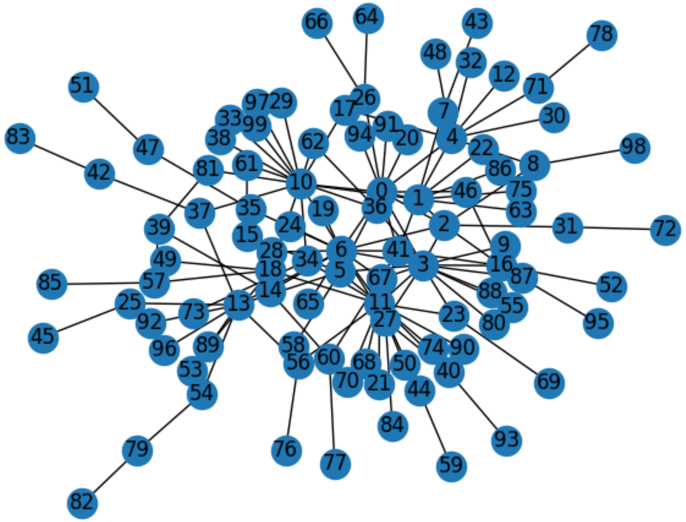




Scale-Free Networks

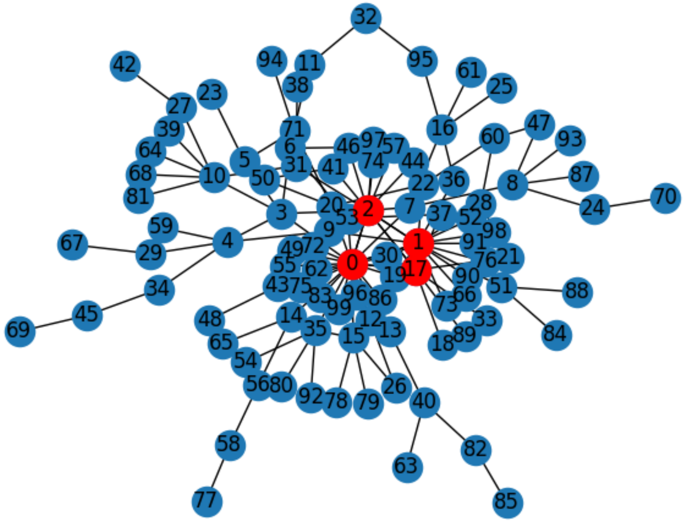
- Construction with preferential to given nodes (hubs)





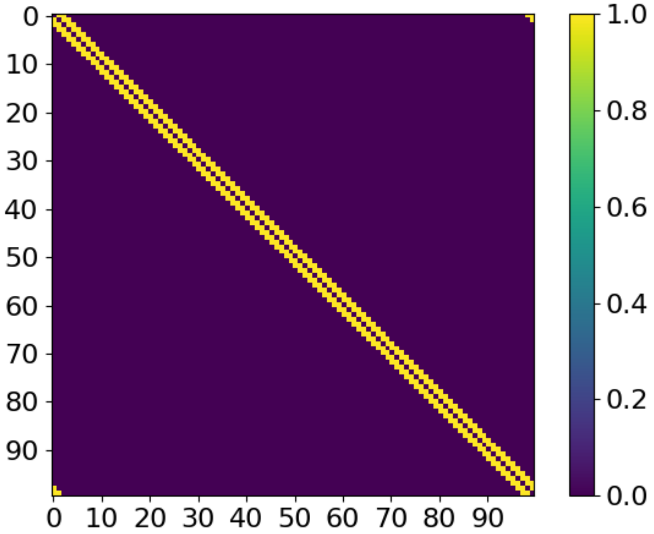
Scale-Free Networks

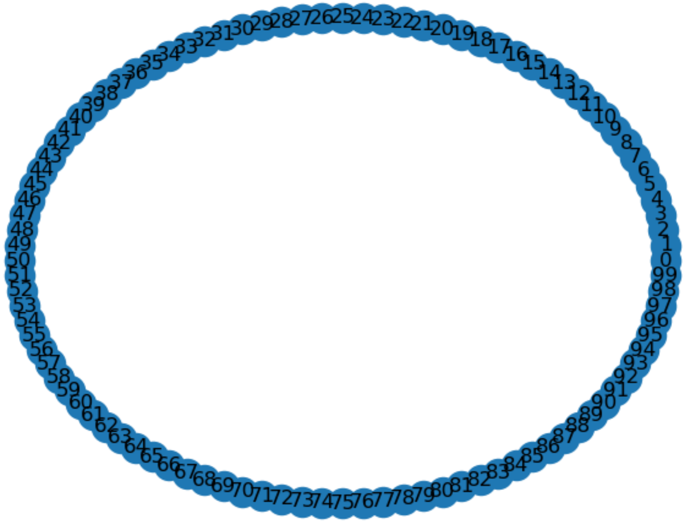
- Construction with preferential to given nodes (hubs)
- Core: subgraph with high degree hubs (here in red)
- Rich club: hubs that are connected together



Small-World Networks

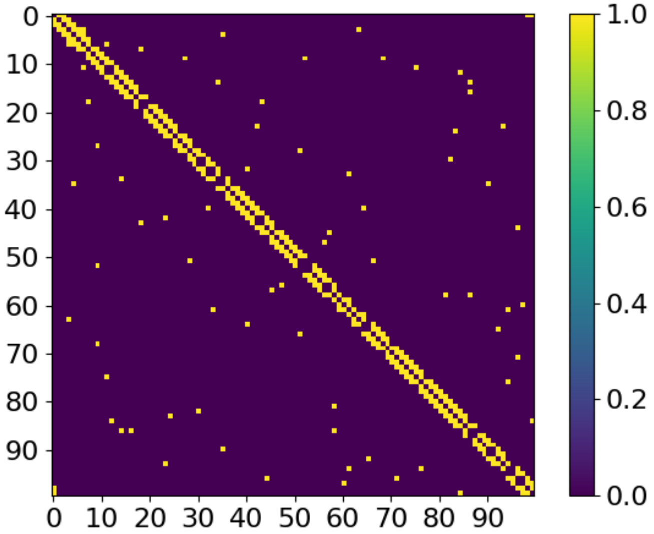
- Ring lattice with rewiring

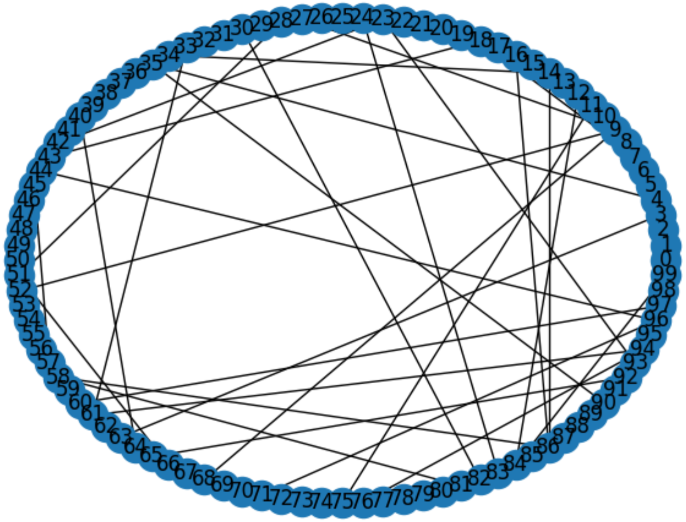




Small-World Networks

- Ring lattice with rewiring
- Rewired links are shortcuts to travel the graph
- No single important node, but distributed architecture with “easy traveling”

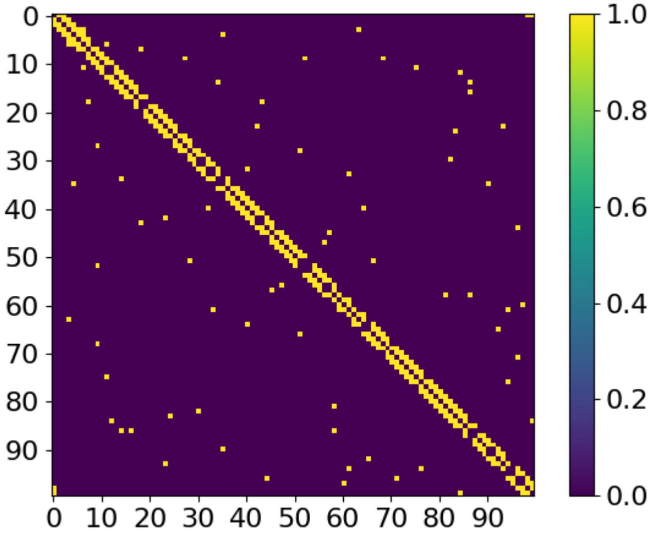


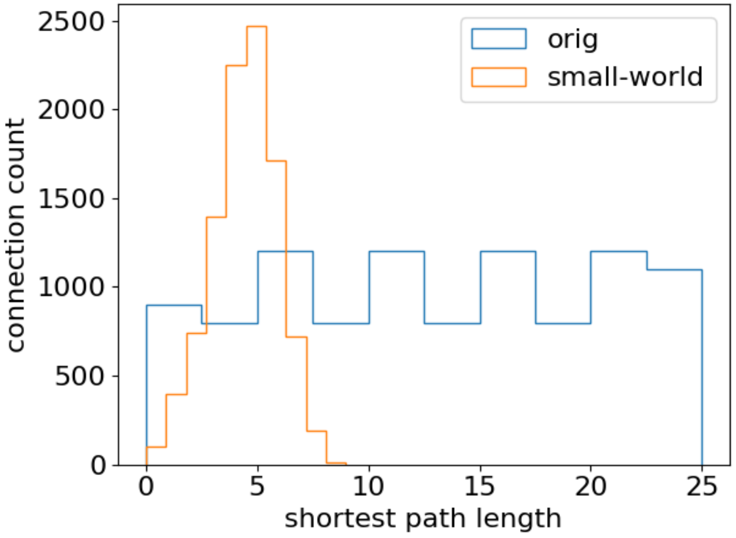


Small-World Networks

- Ring lattice with rewiring
- Rewired links are shortcuts to travel the graph

No single important node, but distributed architecture with “easy traveling”



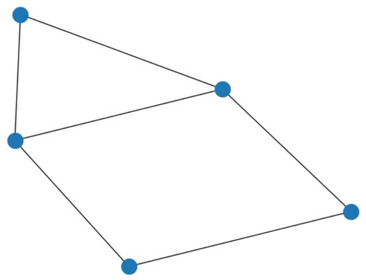


Class 2: Network Theory

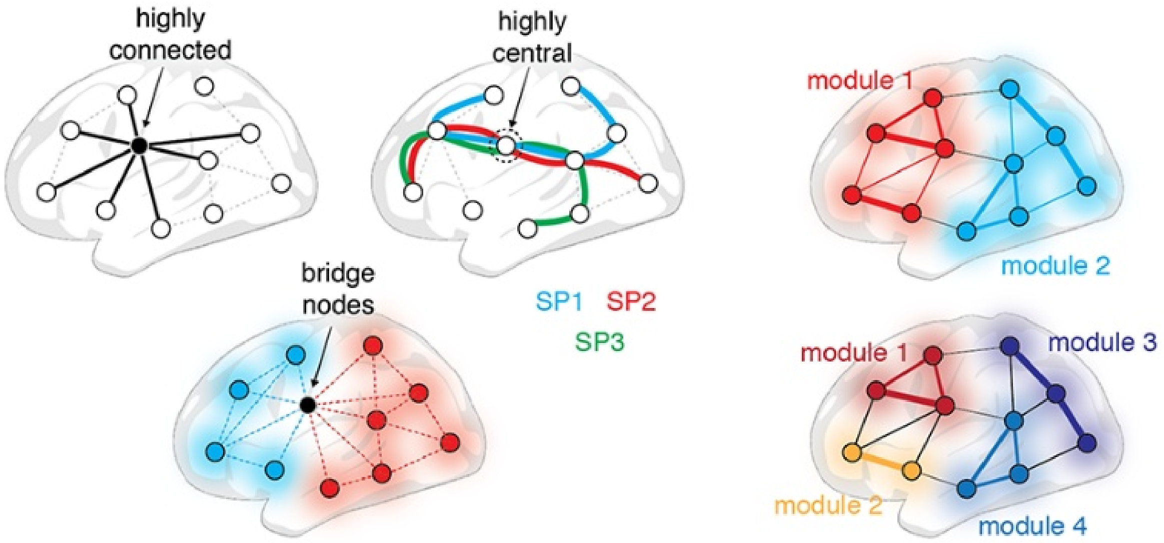
- Graph metrics for undirected binary graphs
  - degree, hubs, centrality
  - communities
  - scale-free and small-world graphs
- **Directed and weighted graphs**
- Application to synthetic and real datasets
  - Python library: networkx

Directed and weighted graphs

- Node degree replaced by node strength $\sum_j A_{ij}$
- Some concepts and graph metrics can be easily transposed, others can't…
  - yes: hubs, community detection based on modularity, ...
  - no: shortest path (is the weight of a link a cost or an efficacy…?)



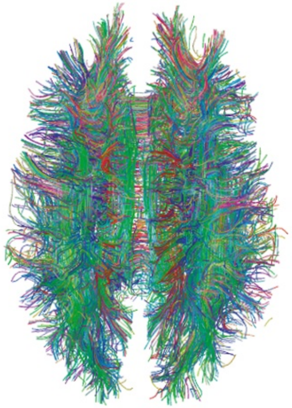
Network Analysis for MRI Data



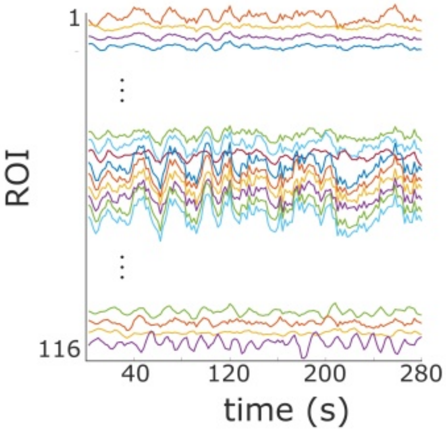
- **How to interpret network measures?**
- Structural connectivity (SC): anatomical white-matter fibers (e.g. density)
- Functional connectivity (FC): correlations of fMRI signals reflecting neuronal activity

Community Detection based on Structural versus Functional Data?

tractography (anatomy)

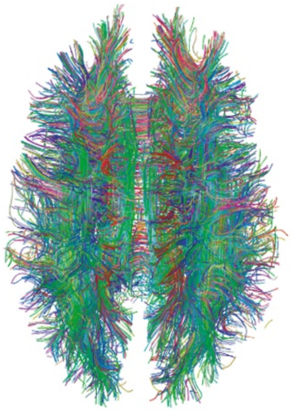


regional activity

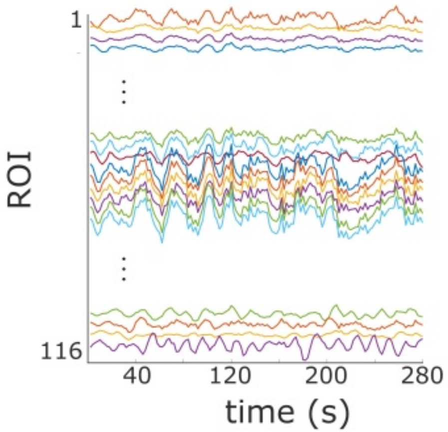


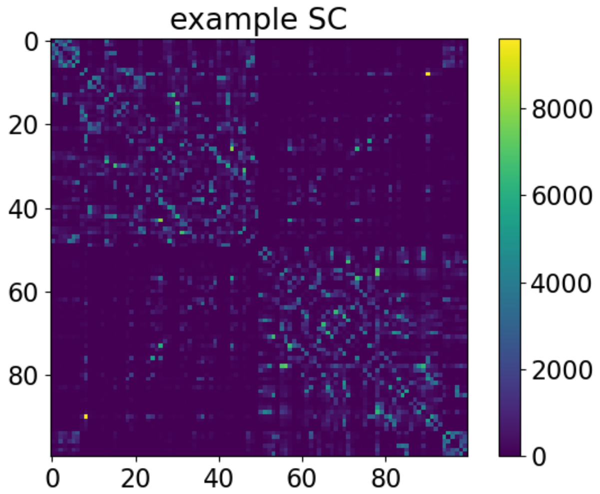
Community Detection based on Structural versus Functional Data?

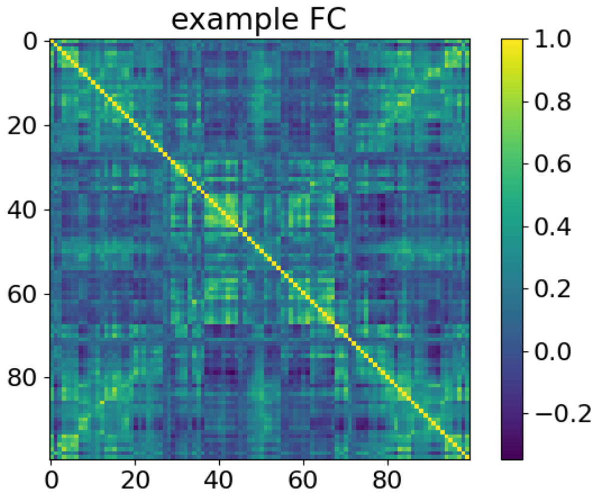
tractography (anatomy)



regional activity

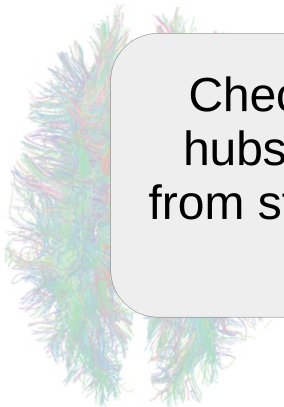




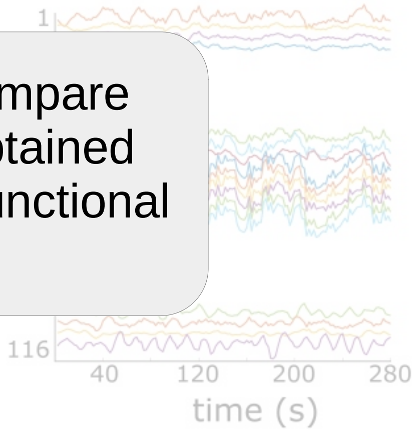


Community Detection based on Structural versus Functional Data?

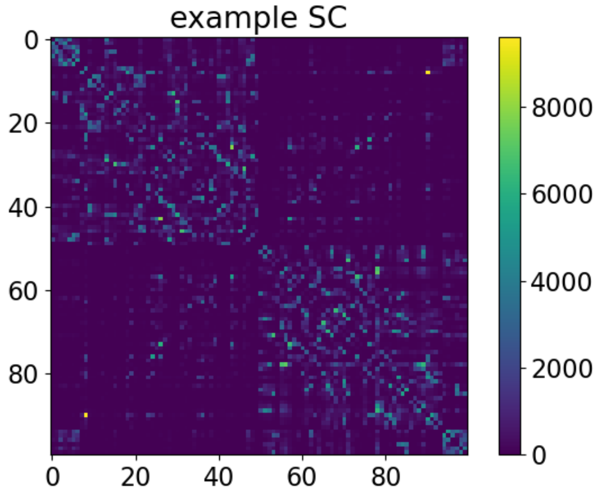
tractography (anatomy)

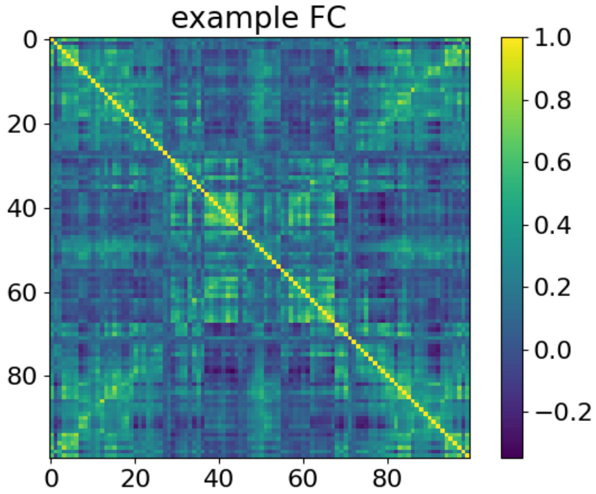


regional activity



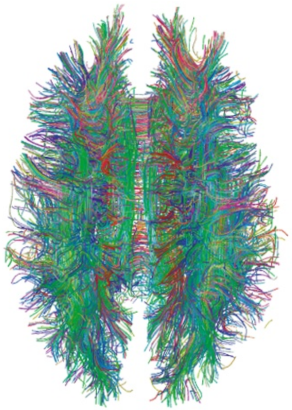
Check notebook to compare hubs / communities obtained from structural versus functional connectomes

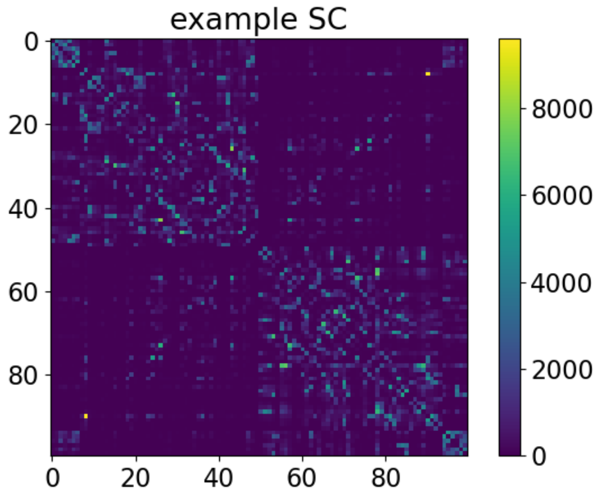




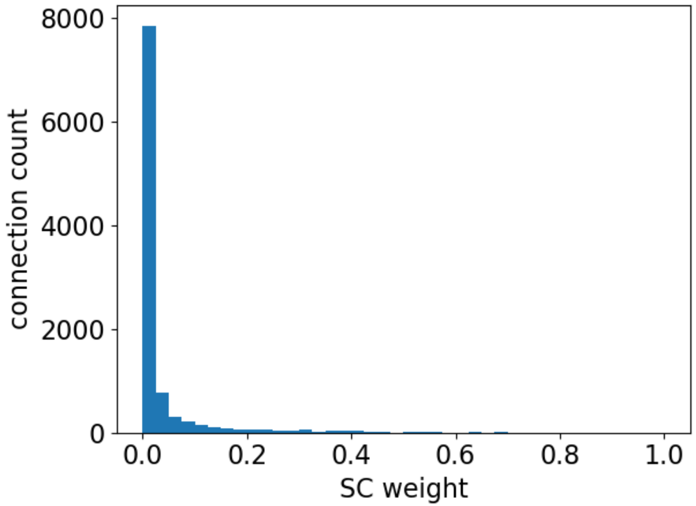
Binarization of Matrices?

tractography (anatomy)



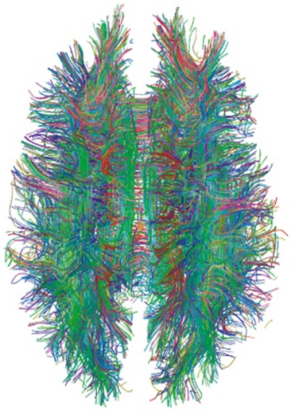


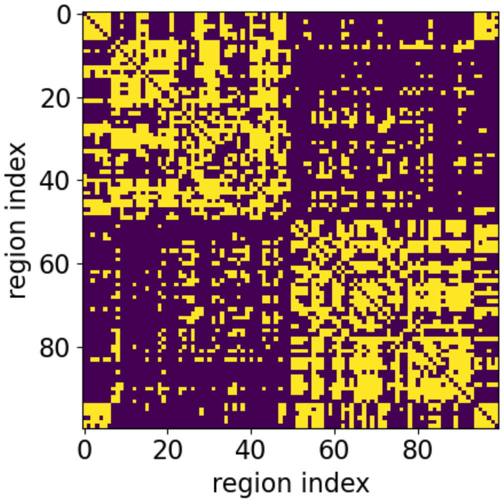
histogram of normalized weights



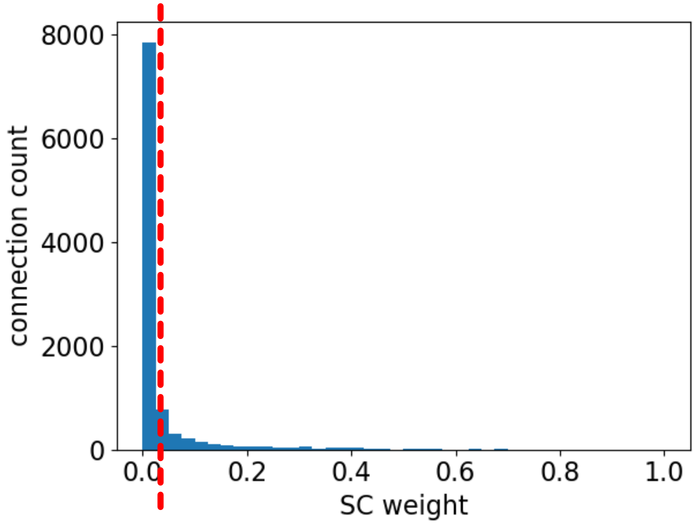
Binarization of Matrices?

tractography (anatomy)





histogram of normalized weights



threshold?

Class 2: Network Theory

- Graph metrics for undirected binary graphs
  - degree, hubs, centrality
  - communities
  - scale-free and small-world graphs
- Directed and weighted graphs
- **Application to synthetic and real datasets**
  - **Python library: networkx**

Practice

- Basic exercises in *basic_network_analysis*
  - undirected binary graphs
  - also first steps toward directed / weighted graphs
- Try real data: *ana_SC_FC.ipynb*
  - *ex_SC_HCP.npy*, *ex_SC.npy*, *ex_FC.npy*
  - identify hubs, core, communities