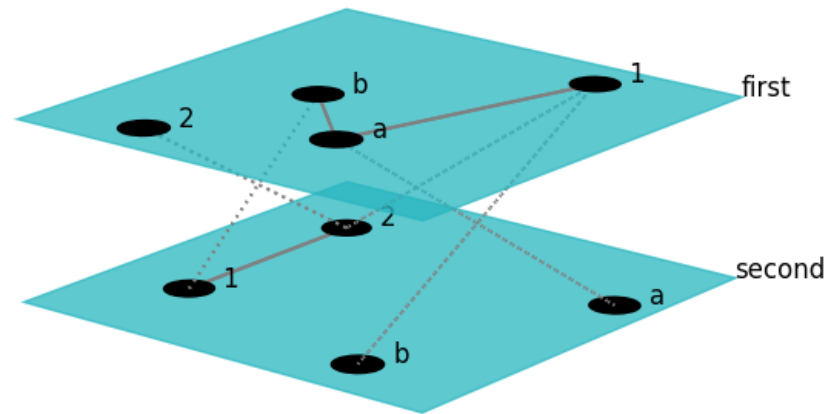


# Multilayer Networks



**Hiroki Sayama**  
sayama@binghamton.edu

# Multilayer networks

---

- Networks that are composed of multiple layers (tautological...)
- A number of different names:
  - Multilayer networks
  - Multiplex networks
  - Interconnected networks
  - Interdependent networks
  - Networks of networks
  - etc...

# Heads up

---

- This is even newer than temporal network research
- Tools for multilayer network research still under active development
  - But there are some tools now available 😊
- More math/theory-driven, and involve more dynamical models, than temporal network research

# Examples

---

- Social network that involve different types of connections
- Network of airports connected by different air carriers
- Multiple infrastructures of a nation that are connected to each other
- Species interaction patterns that involve distinct layers of habitat

# Fundamentals of Multilayer Networks

# General representation

---

- To represent a multilayer network, define the following function:

$$a(i, \alpha, j, \beta) = 0 \text{ or } 1 \text{ (or weight)}$$

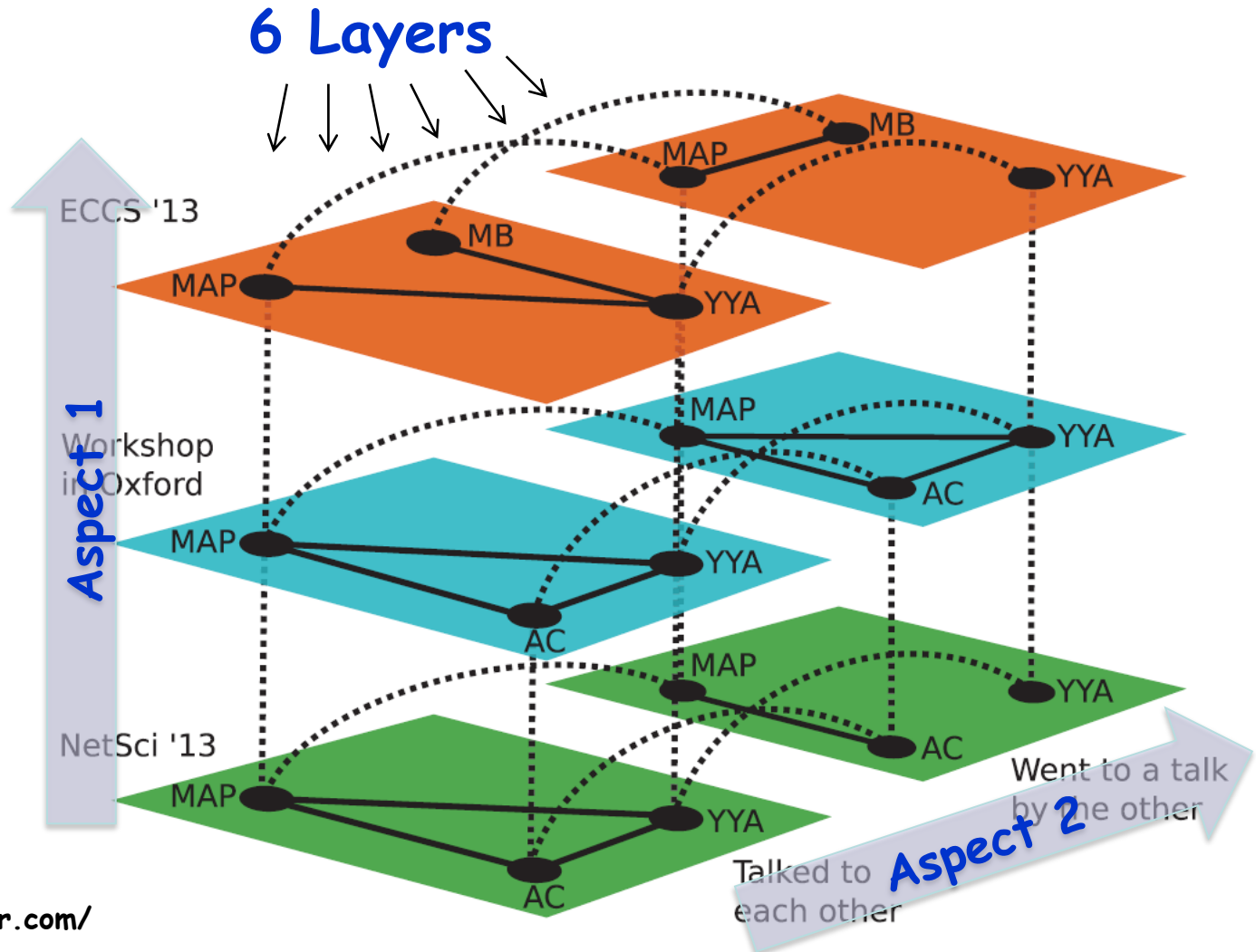
- $a(i, \alpha, j, \beta) = 1$  if node  $i$  in layer  $\alpha$  is connected to node  $j$  in layer  $\beta$ , otherwise 0  
(This can be represented mathematically by an adjacency tensor)

# Aspects

---

- Layers  $\alpha$ ,  $\beta$  can be mapped to a space made of multiple “aspects”
  - Time, location, type of connection, etc.
  - This means  $\alpha$ ,  $\beta$  can be vectors
- Aspects help organize the relationships between different layers

# Example: Zachary's Karate Club Club (ZKCC) network



<http://networkkarate.tumblr.com/>

Source: Kivela et al. (2014)



# Exercise

---

- Write down the values of the adjacency function  $a(i, \alpha, j, \beta)$  for some edges in the previous ZKCC network
- Represent the same information in an edge list

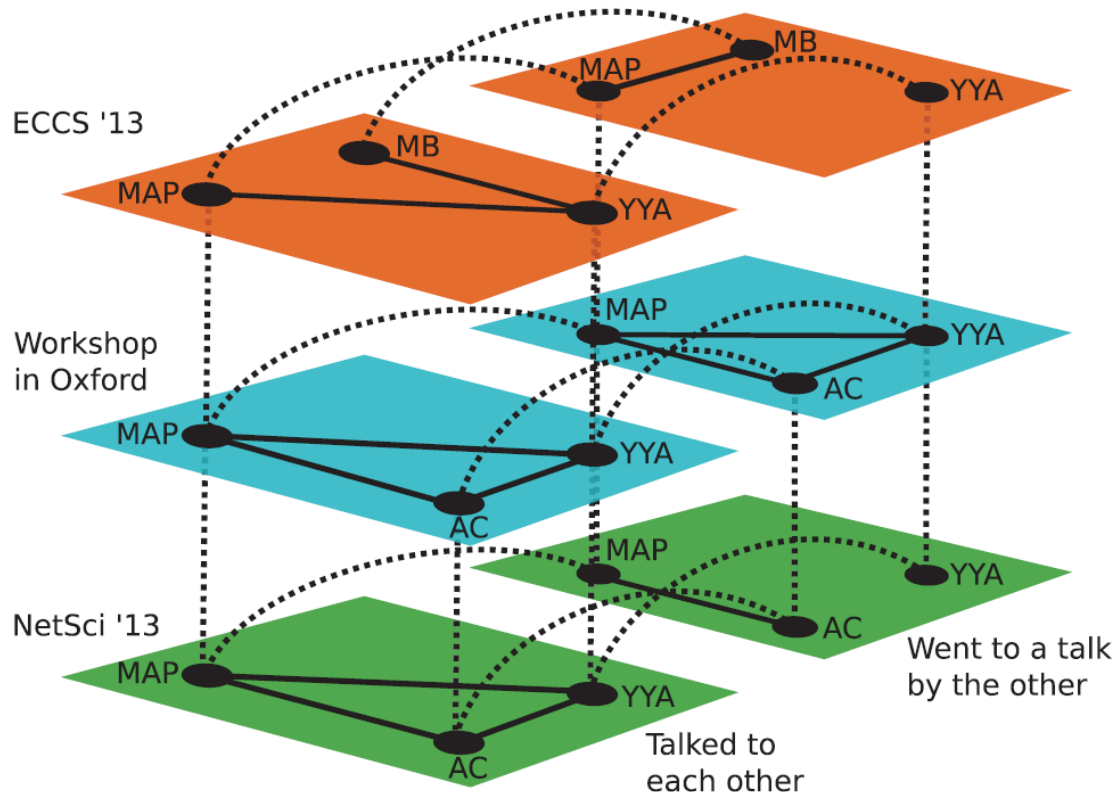
# Different classes of edges

---

- Intra-layer edge  $[(i, \alpha), (j, \alpha)]$
- Inter-layer edge  $[(i, \alpha), (j, \beta)] \quad \alpha \neq \beta$ 
  - In particular:  
Coupling edge  $[(i, \alpha), (i, \beta)] \quad \alpha \neq \beta$

# Exercise

- Identify (a) intra-layer edges, (b) inter-layer edges and (c) coupling edges in this multilayer network



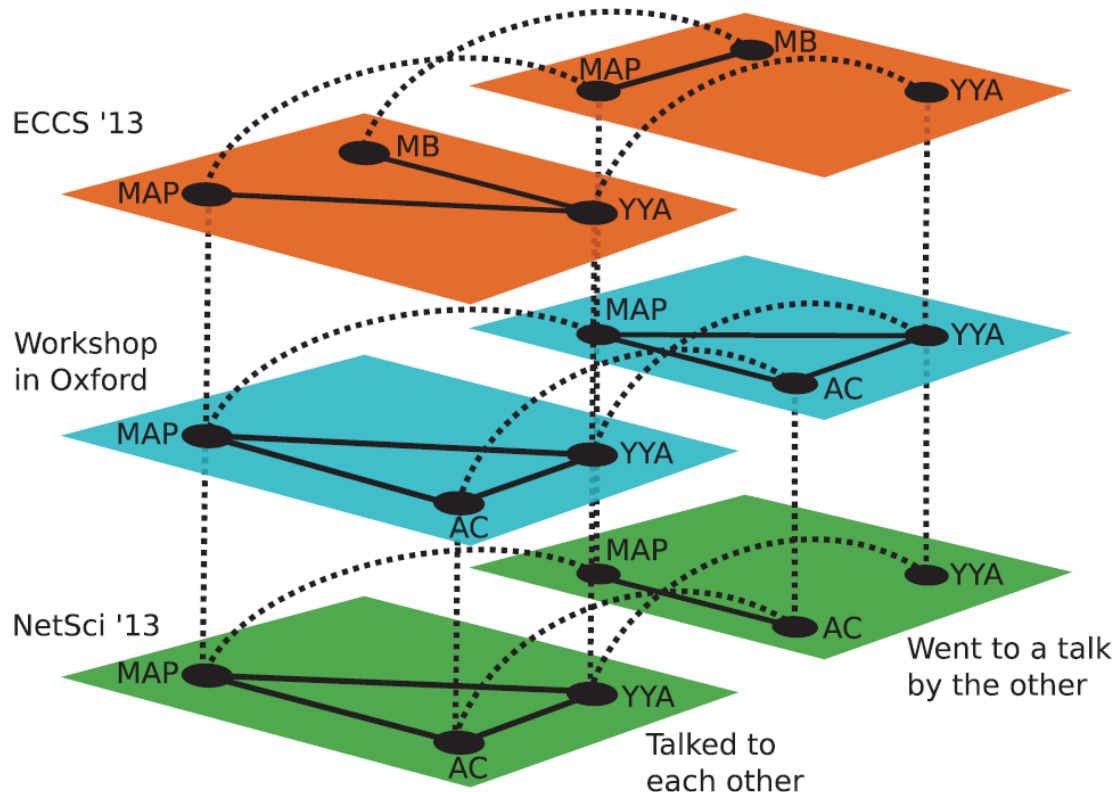
# Properties of multilayer networks

---

- Do all the layers contain the same set of nodes? → **node-aligned**
- Or, does each node appear only in one layer? → **layer-disjoint**
- Are all the inter-layer edges coupling ones? → **diagonally coupled**
  - Are the diagonal coupling edges independent of nodes? → **layer-coupled**

# Exercise

- Is this network (a) node-aligned or layer-disjoint?  
(b) diagonally coupled? (c) layer-coupled?



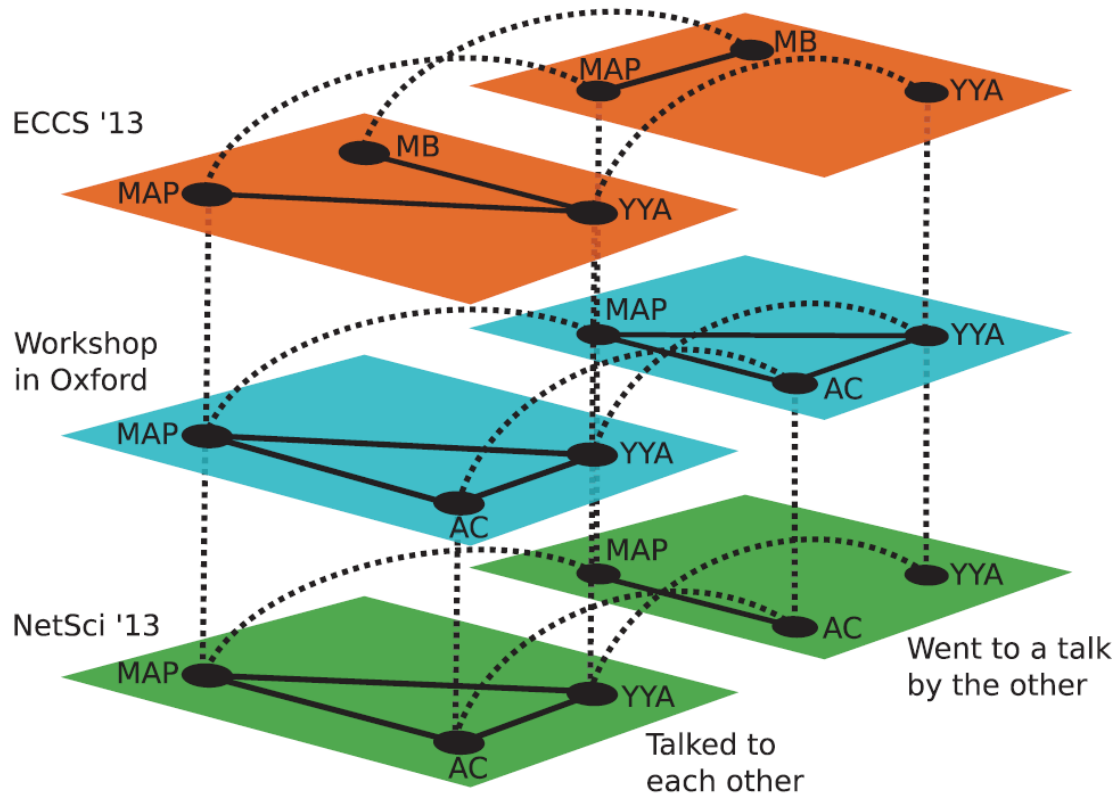
# Different kinds of diagonal coupling along an aspect

---

- Do the diagonal edges connect a node to its counterparts in all other layers along an aspect? → **categorical**
- Or, do the diagonal edges connect a node to its counterparts only in “nearby” layers along an aspect?  
→ **ordinal**

# Exercise

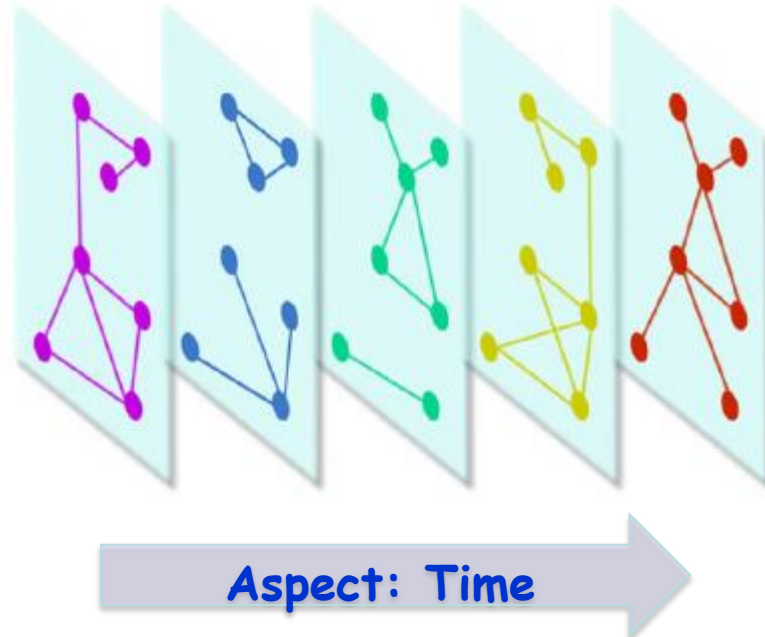
- Does each of the two aspects of this network (a) categorical or (b) ordinal?



# Temporal networks as multilayer networks

---

- Temporal networks can also be considered a special case of multilayer networks that have only one ordinal aspect: **time**



Source: De Domenico  
et al. (2013)



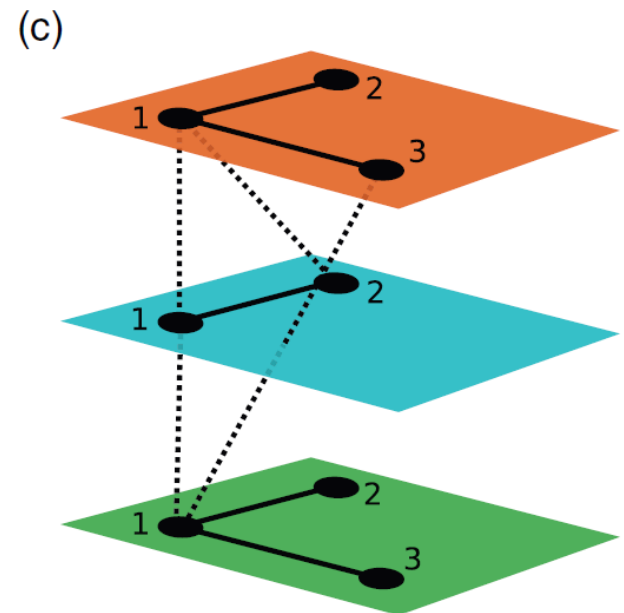
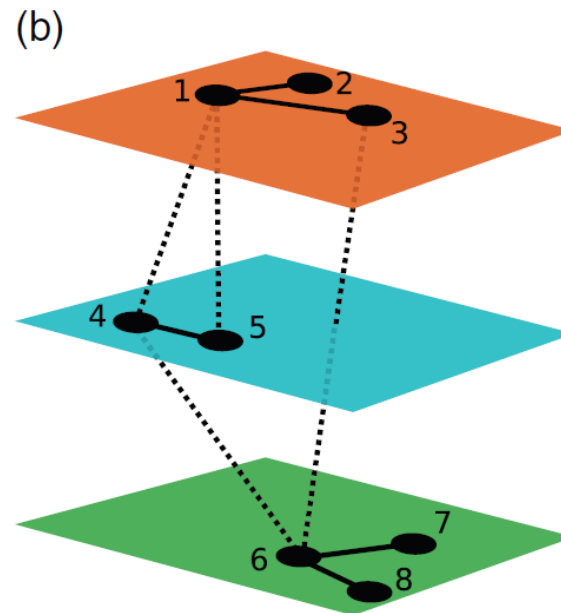
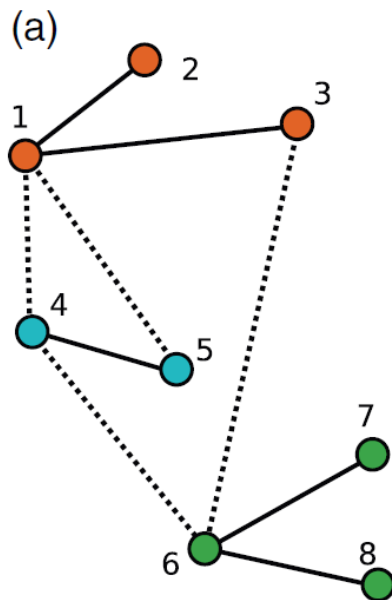
# Two major subclasses of multilayer networks

---

- **Node-colored networks**  
(a.k.a. “interconnected networks”, “interdependent networks”, “networks of networks”)
- **Edge-colored networks**  
(a.k.a. “multiplex networks”, “multirelational networks”)

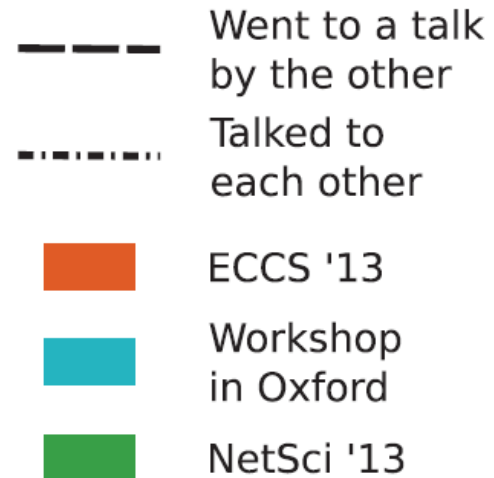
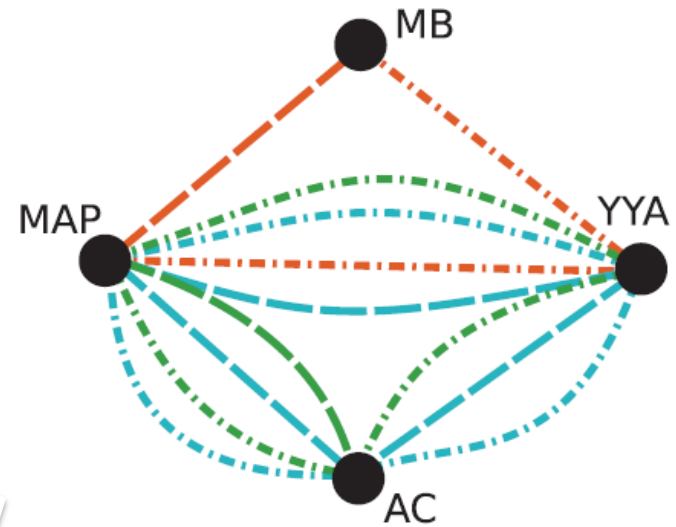
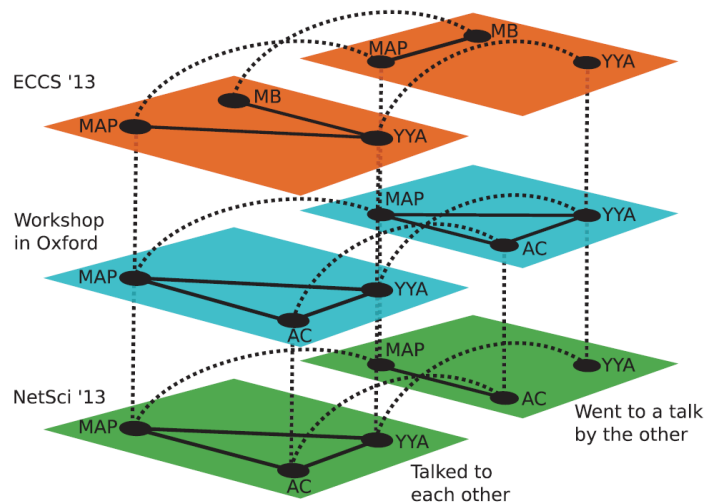
# Node-colored networks

- Interconnected networks, interdependent networks, networks of networks, etc.



# Edge-colored networks

- Multiplex networks, multirelational networks, etc.



Source: Kivelä et al. (2014)

# Exercise

---

- Give a few real-world examples of (a) node-colored and (b) edge-colored multilayer networks
- Discuss properties of those networks, especially the types of their inter-layer connections

# Computational Modeling and Analysis of Multilayer Networks

# PlexMath project

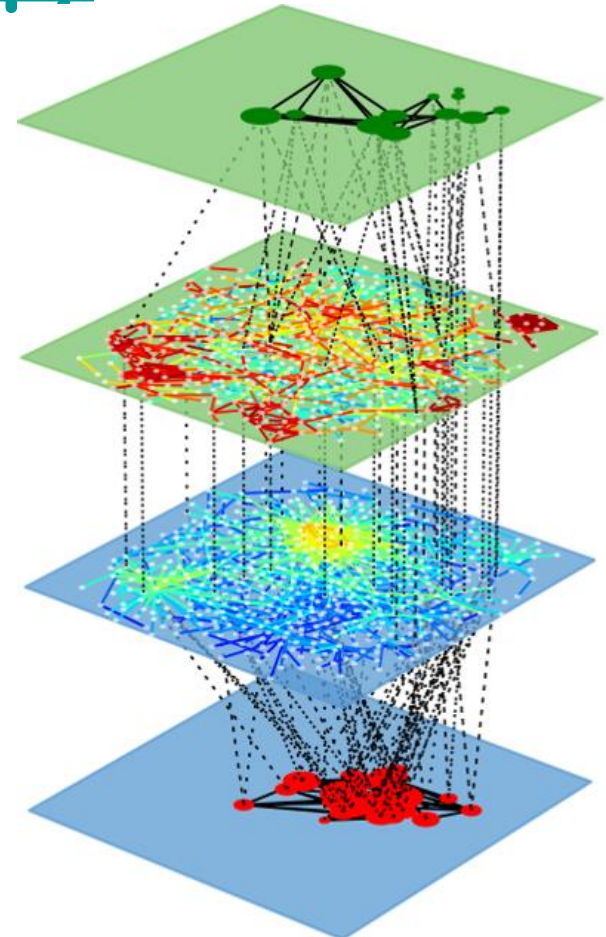
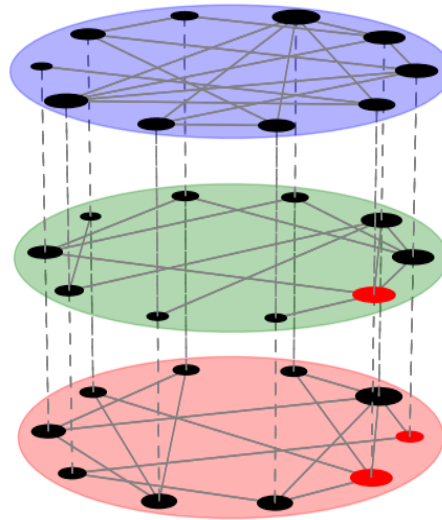
- <http://deim.urv.cat/~alephysys/plexmath/>

The screenshot shows the PlexMath website homepage. At the top left is the PlexMath logo, which consists of three overlapping rectangular planes in red, green, and blue, followed by the text "Plexmath". To the right of the logo is a navigation menu with the following items: HOME, PEOPLE, PUBLICATIONS, RESOURCES » (which is expanded to show a sub-menu with "Software" circled in yellow), and NEWS. Below the navigation menu is a large banner for "SCIENTIFIC DATA" featuring a MuxViz visualization of a multi-modal transportation system. The banner text reads: "MuxViz visualization of multi-modal transportation system featured on Nature Scientific Data". To the right of the banner is a vertical stack of three maps: the top one is labeled "Rail", the middle one "Metro", and the bottom one "Bus". Below the banner is a section titled "Nature Scientific Data" with the text "MuxViz visualization of multi-modal transportation system featured on Nature Scientific Data" and a "More »" link. At the bottom of the page, there is a "Home" section with a date "June 26, 2013" and a "Home" link, and a search bar.

# Pymnet library

---

- <http://www.mkivela.com/pymnet/>
  - Developed by Mikko Kivelä



# Exercise

---

- Download the pymnet library and its documentation
- Follow its tutorial to learn how to build multilayer/multiplex networks
  - Place the "pymnet" folder to your working directory, or add it to your Python path
  - Commands are different from NetworkX, but access to all NetworkX commands are provided under "nx." prefix



# Pymnet's data structure

---

- **MultilayerNetwork(aspects = \*\*)**
  - To represent general multilayer networks
- **MultiplexNetwork(couplings = \*\*)**
  - To specifically represent multiplex networks
  - **`nx.Graph(g)`** converts these into a NetworkX Graph (monolayer only)

# Pymnet: Basics

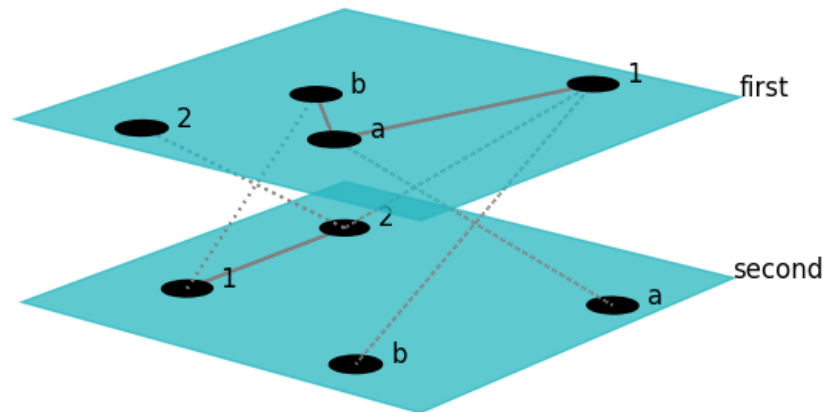
---

- `add_node()`, `add_layer()`
- `g[ID][ID] = 1` adds edge (ID: sequence of indices, including aspects)
- `list(g)` gives node list
- `list(g.iter_**())` gives node, layer, or node-layer list
- `list(g.edges)` gives edge list
- `list(g[ID])` gives neighbor list
- `g[ID].deg()` or `.str()` gives degree or strength

# Drawing multilayer networks

---

- **Just `draw(g)` !!**



- Use “`%matplotlib auto`” so you can click and drag to rotate the plot
- Many options are available to customize visualization results

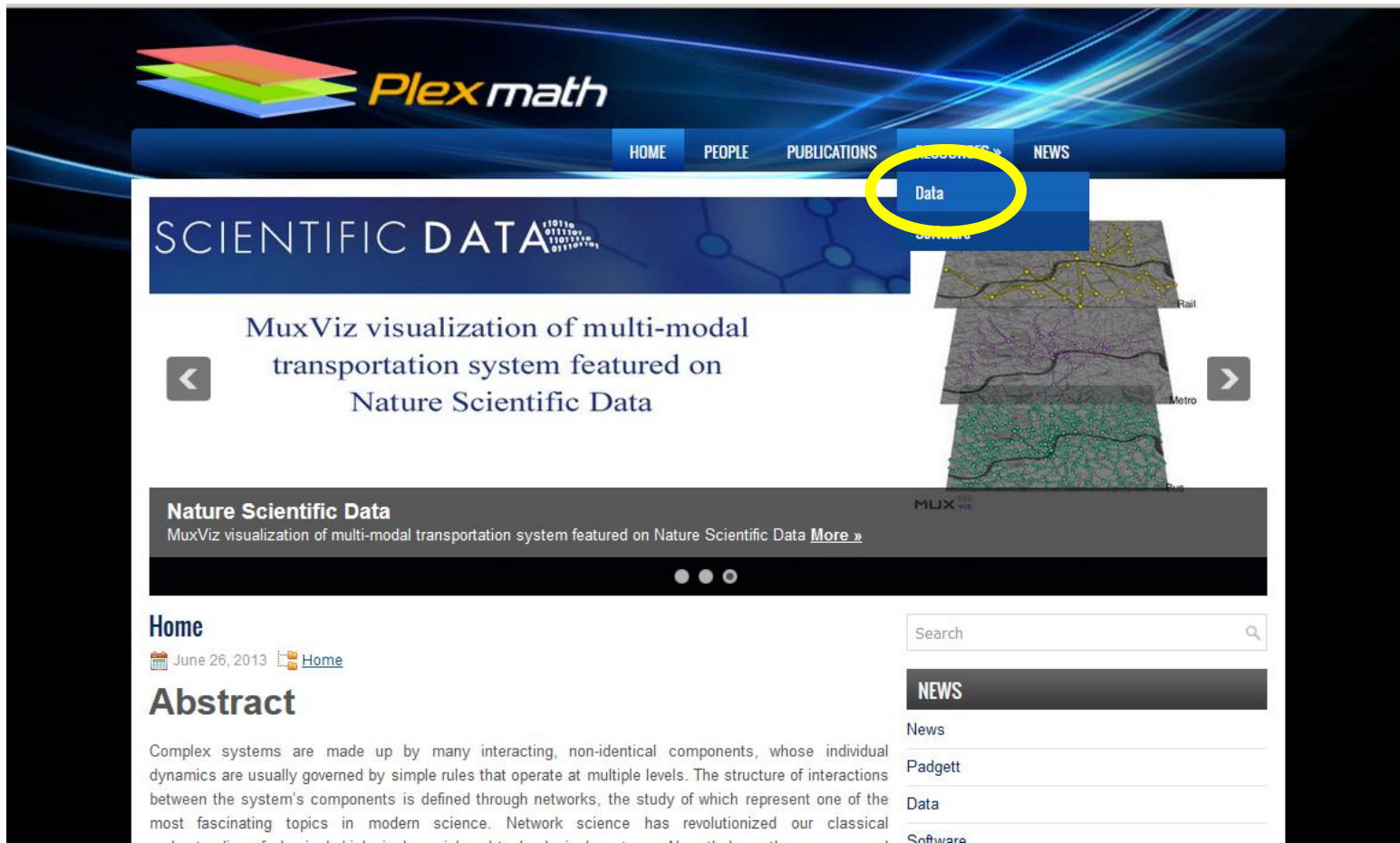
# Exercise

---

- Check out the references of pymnet
  - <http://www.mkivela.com/pymnet/reference.html>
- Create a few multilayer/multiplex networks and visualize them
- Explore customization of visualization as you like

# Get the data!

- <http://deim.urv.cat/~alephysys/plexmath/>



The screenshot shows the Plexmath website interface. At the top left is the Plexmath logo, which consists of three overlapping rectangular planes in red, green, and blue. To the right of the logo is the text "Plexmath". Below the logo is a dark blue navigation bar with white text for "HOME", "PEOPLE", "PUBLICATIONS", "RESOURCES", and "NEWS". A dropdown menu is open under "RESOURCES", and the "Data" option is highlighted with a yellow circle. Below the navigation bar is a large banner with the text "SCIENTIFIC DATA" and a molecular structure background. The banner features a central text block: "MuxViz visualization of multi-modal transportation system featured on Nature Scientific Data". To the right of this text is a 3D visualization of a transportation network with nodes and edges, labeled "Rail" and "Metro". Below the banner is a dark grey section with the text "Nature Scientific Data" and a link "More »". At the bottom left, there is a "Home" section with a date "June 26, 2013" and a "Home" link. Below that is an "Abstract" section with a paragraph of text. On the right side, there is a search bar and a "NEWS" section with a list of items: "News", "Padgett", "Data", and "Software".

# Exercise

---

- Download the “Vickers & Chan 7th graders social network” data from the PlexMath website
- Write a Python code to read the data and construct a multilayer network
  - As a **MultilayerNetwork** object
  - As a **MultiplexNetwork** object
- Visualize it

# Some built-in measurements

---

- `degs(g)` gives degree distribution
- `density(g)` gives network density
- `multiplex_degs(g)` for each layer
- `multiplex_density(g)` for each layer
- Various clustering measurements
  - But not so many functions available yet...  
(the author Mikko said he would welcome any contributions!)

# Exercise

---

- Plot the degree distribution for each layer in the Vickers & Chan 7<sup>th</sup> graders multiplex network
- Measure the density of each layer and determine which layer was most/least dense



# Other network measurements

---

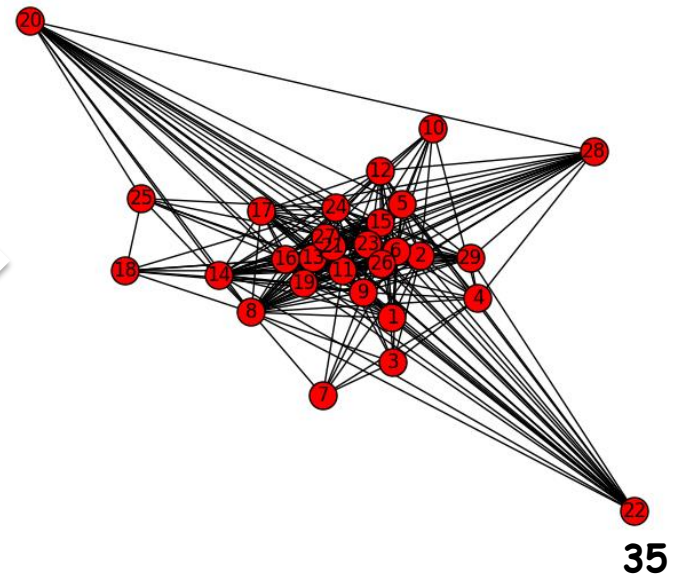
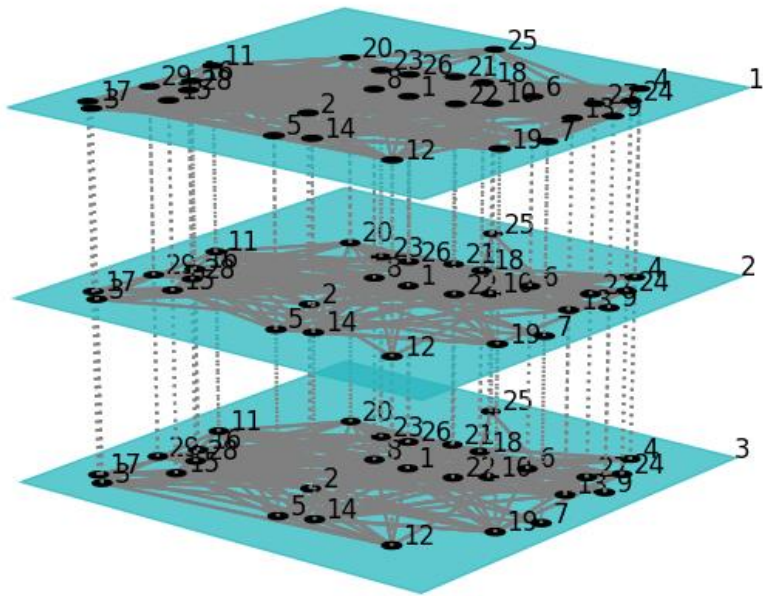
- You can represent a multilayer network as a plain monolayer network and apply various measurements
- Is it a multiplex network and does changing layers take no cost or distance?
  - Yes to both → Aggregate the network
  - Otherwise → Flatten the network

# Note: When you create a monolayer network...

- Consider how the weights of intra-layer edges compare to each other between different layers
- Each layer may represent different connectivity; adjust their weights as needed
  - E.g. :  
Best friend, friend, acquaintance  
Flight, train, bus, bike, walk

# Aggregating layers

- `aggregate(g, aspects)` creates a simple network aggregated over aspects
  - Works for both multilayer and multiplex, but most meaningful for multiplex



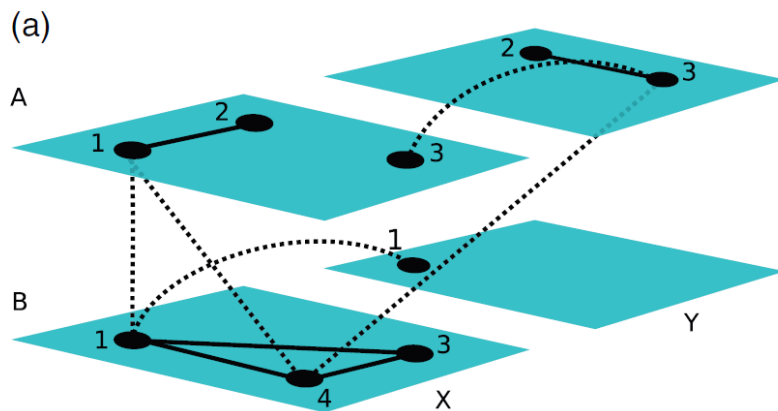
# Exercise

---

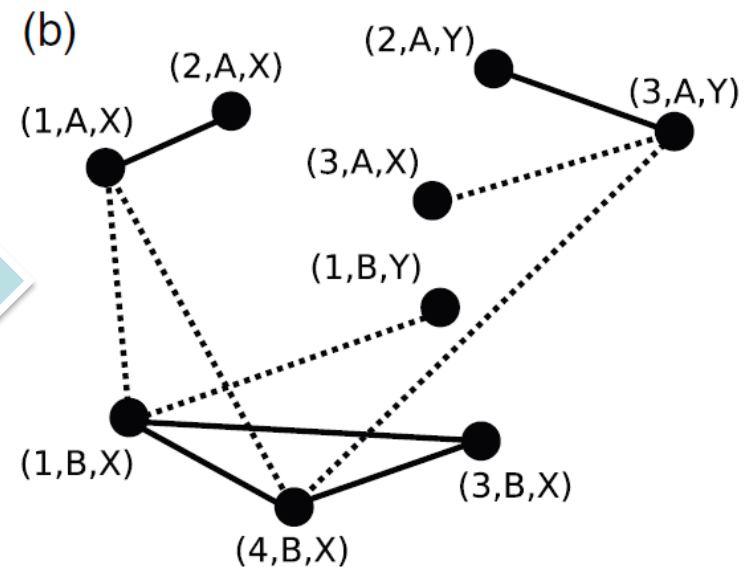
- Create an aggregated network of the Vickers & Chan 7th graders multiplex network
- Calculate several shortest paths, average shortest path length, and several centralities of the nodes
- Detect communities using the Louvain method

# “Flattening” multilayer networks

- Ignore layers and represent all the connections in a plain network



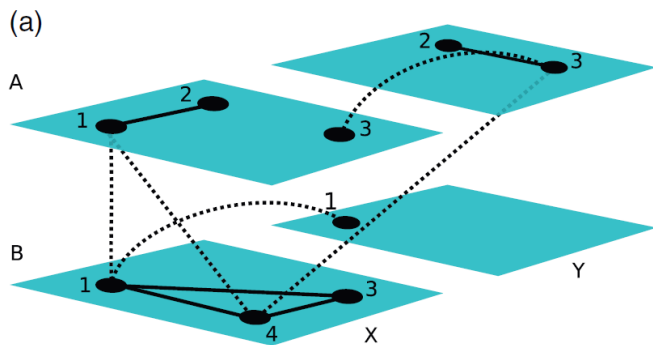
Source: Kivelä et al. (2014)



- (This is essentially the same as the construction of a space-time network for a temporal network)

# Supra-adjacency matrix

- An adjacency matrix of a flattened version of a multilayer network



	A		B		
	X	Y	X	Y	
A	Adj. matrix of (A,X)	Inter-layer matrix	Inter-layer matrix	Inter-layer matrix	X
	Inter-layer matrix	Adj. matrix of (A,Y)	Inter-layer matrix	Inter-layer matrix	Y
B	Inter-layer matrix	Inter-layer matrix	Adj. matrix of (B,X)	Inter-layer matrix	X
	Inter-layer matrix	Inter-layer matrix	Inter-layer matrix	Adj. matrix of (B,Y)	Y

If inter-layer edges are “diagonal” couplings, the inter-layer matrices are diagonal too (hence the name)

# Supra-adjacency matrix in pymnet

---

- `supra_adjacency_matrix(g)` creates the supra-adjacency matrix of  $g$  and a list of node IDs (arranged in the order used in the matrix)
- `nx.from_numpy_matrix(A)`, with  $A$  being the supra-adjacency matrix, creates a flattened monolayer network

# Exercise

---

- Create a flattened network of the Vickers & Chan 7th graders multiplex network
- Calculate several shortest paths, average shortest path length, and several centralities of the nodes
- Detect communities using the Louvain method



# Inter-layer measurement

---

- “Interdependence”

Ratio of the number of shortest paths that use multiple layers to the total number of shortest paths

High interdependence

→ Communication/transportation in a multilayer network uses multiple layers more often (i.e., those layers depend on each other)

# Exercise

---

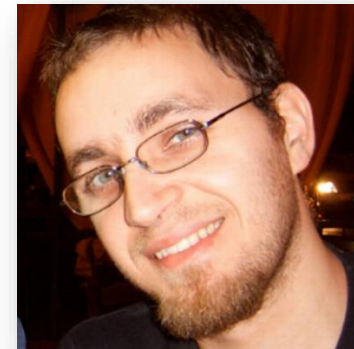
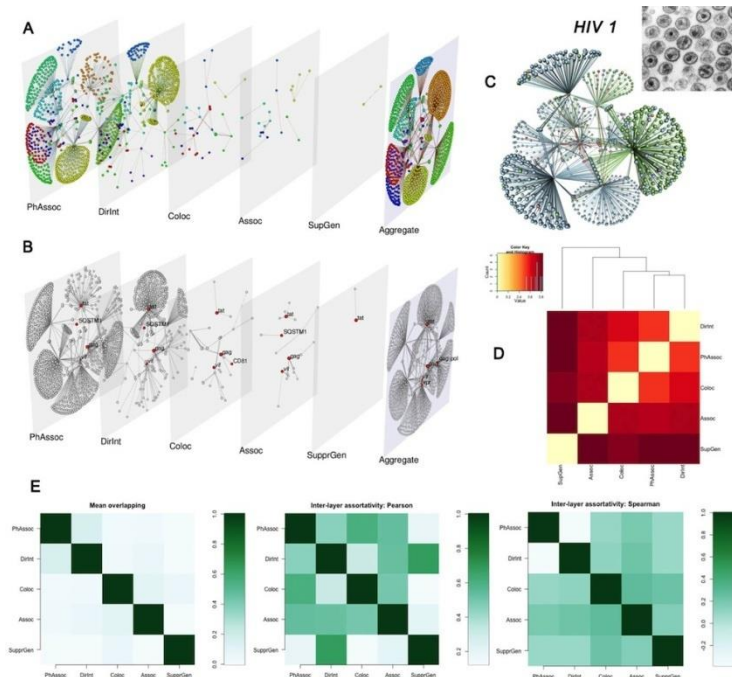
- Design and implement a code to measure the level of interdependence in the Vickers & Chan 7th graders multiplex network
- Apply the code to other multilayer network data

# FYI: If you use R...

- muxViz

- <http://muxviz.net>

- [http://www.youtube.com/watch?v=gcpYSdi\\_xI](http://www.youtube.com/watch?v=gcpYSdi_xI)



Developed by  
Manlio De Domenico

# Dynamical Processes on Multilayer Networks

# Dynamics on multilayer networks

- Several dynamical models have been studied recently
- Mostly focused on **two-layer multiplex networks**
- `pymnet`'s network objects cannot contain additional attributes such as states, so you need to use `NetworkX`

# Cascade of failures

---

- Buldyrev, S. V., Parshani, R., Paul, G., Stanley, H. E., & Havlin, S. (2010). Catastrophic cascade of failures in interdependent networks. *Nature*, 464(7291), 1025-1028.

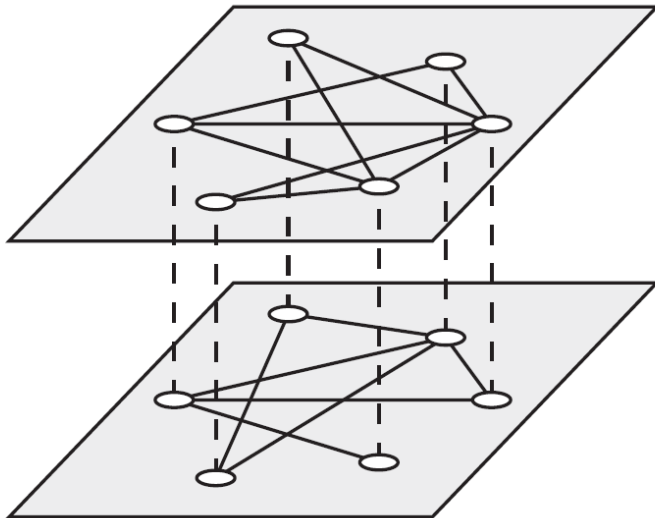


- Proposed a model of two-layer interdependent network (e.g., power grid and the Internet)
- Fragmentation by a cascade of failures occurs quite differently on this model than monolayer networks

# Diffusion

---

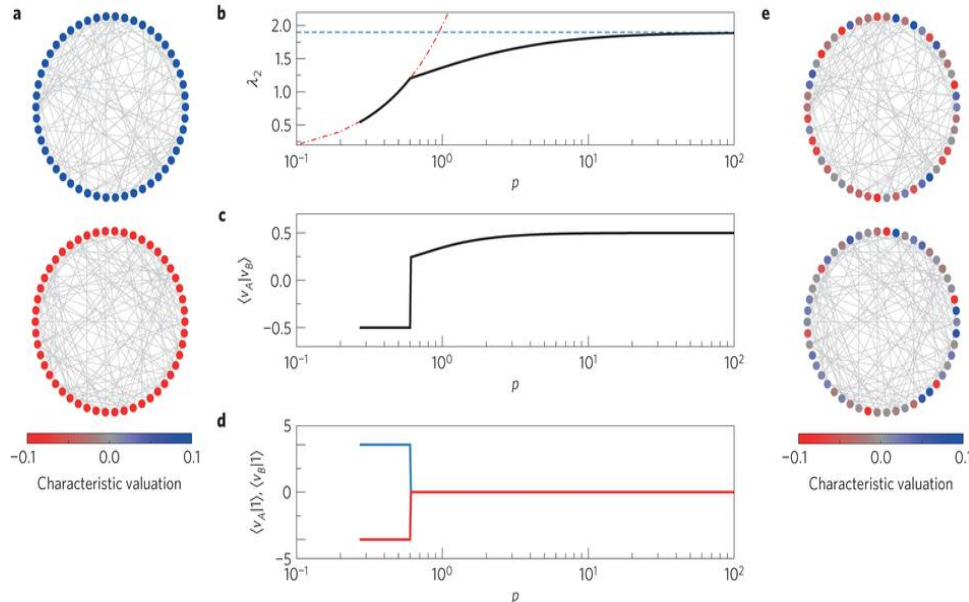
- Gomez, S., Diaz-Guilera, A., Gomez-Gardeñes, J., Perez-Vicente, C. J., Moreno, Y., & Arenas, A. (2013). Diffusion dynamics on multiplex networks. *Physical Review Letters*, 110(2), 028701.



- Studied diffusion processes on two-layer multiplex networks
- Analyzed the spectrum of a “supra-Laplacian” matrix of the network

# Structural transition

- Radicchi, F., & Arenas, A. (2013). Abrupt transition in the structural formation of interconnected networks. *Nature Physics*, 9(11), 717-720.



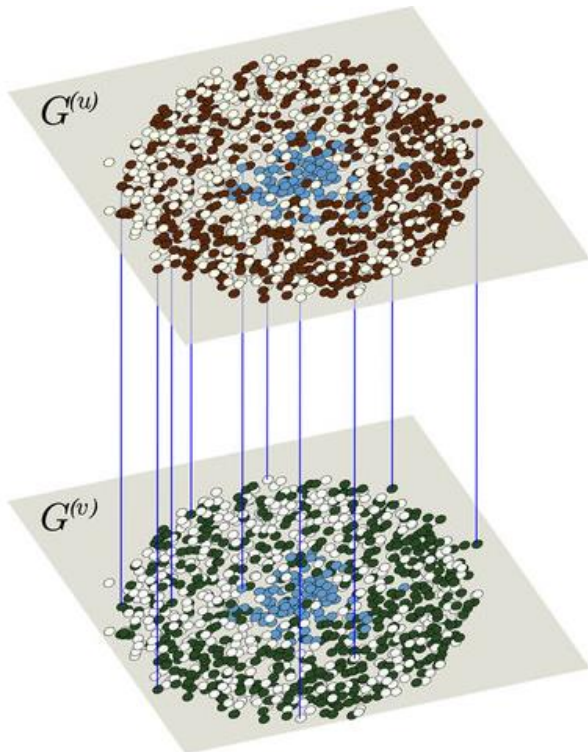
- Discovered that the algebraic connectivity and Fiedler's vector undergo a sharp transition as inter-layer connectivity is increased on two-layer multiplex networks



# Pattern formation

---

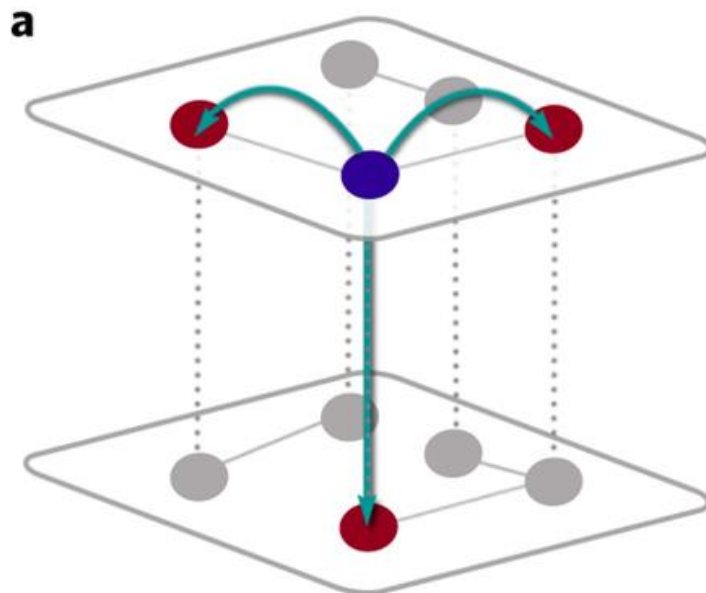
- Kouvaris, N. E., Hata, S., & Díaz-Guilera, A. (2015). Pattern formation in multiplex networks. *Scientific reports*, 5, 10840.



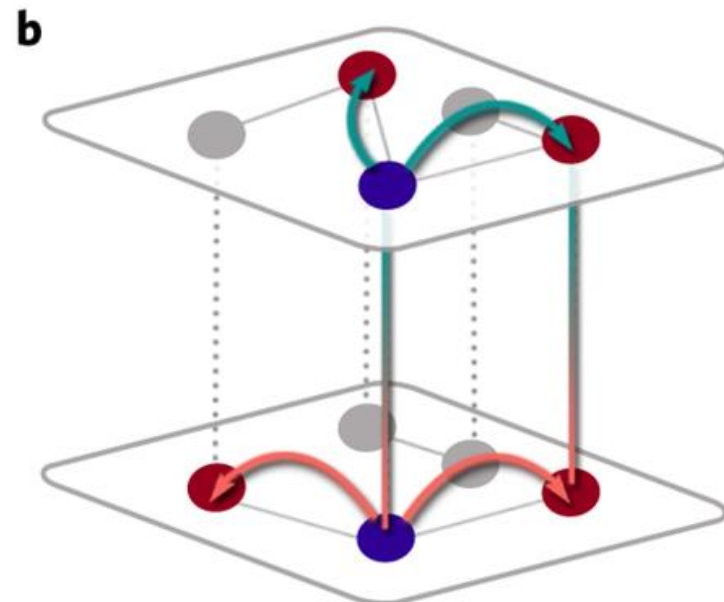
- Studied reaction-diffusion dynamics on two-layer multiplex networks
- Showed that the difference in topologies between two layers can cause pattern formation (a.k.a. Turing instability) *even if diffusion constants are the same*

# Spreading

- De Domenico, M., Granell, C., Porter, M. A., & Arenas, A. (2016). The physics of spreading processes in multilayer networks. *Nature Physics* 12, 901-906.



Single dynamics



Coupled dynamics

# Exercise

---

- Implement a “cascade of failure” model on a random multilayer network with  $k$  layers
- Conduct numerical simulations with the probabilities of intra- and inter-layer connections systematically varied
- Identify the parameter values with which large-scale failure occurs