



**Pacific Northwest**  
NATIONAL LABORATORY

# Modeling Complex Data with Hypergraphs

October 8, 2019

**Emilie Purvine**

Sinan Aksoy, Song Feng, Jesun Firoz,  
Emily Heath (UIUC), Louis Jenkins (U Rochester),  
Cliff Joslyn, Jason McDermott, Hugh Mitchell,  
Brenda Praggastis, Marcin Zalewski (NVIDIA),  
Mark Raugas

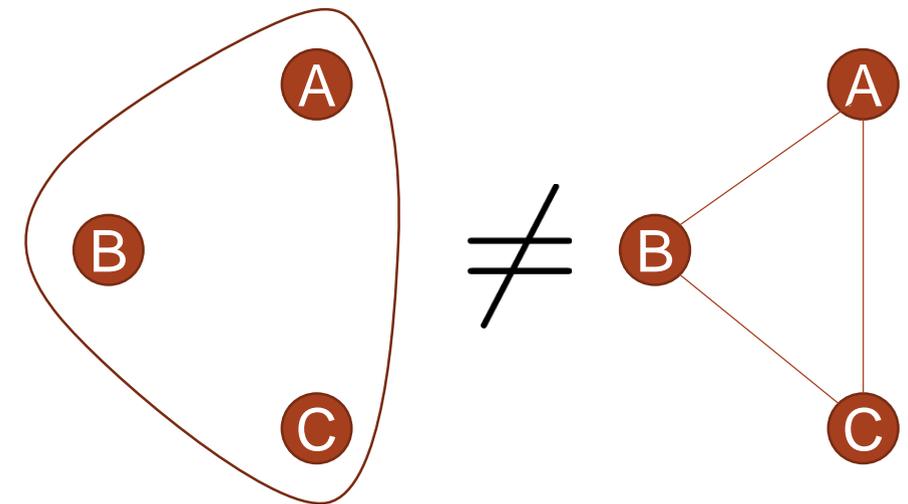


PNNL is operated by Battelle for the U.S. Department of Energy



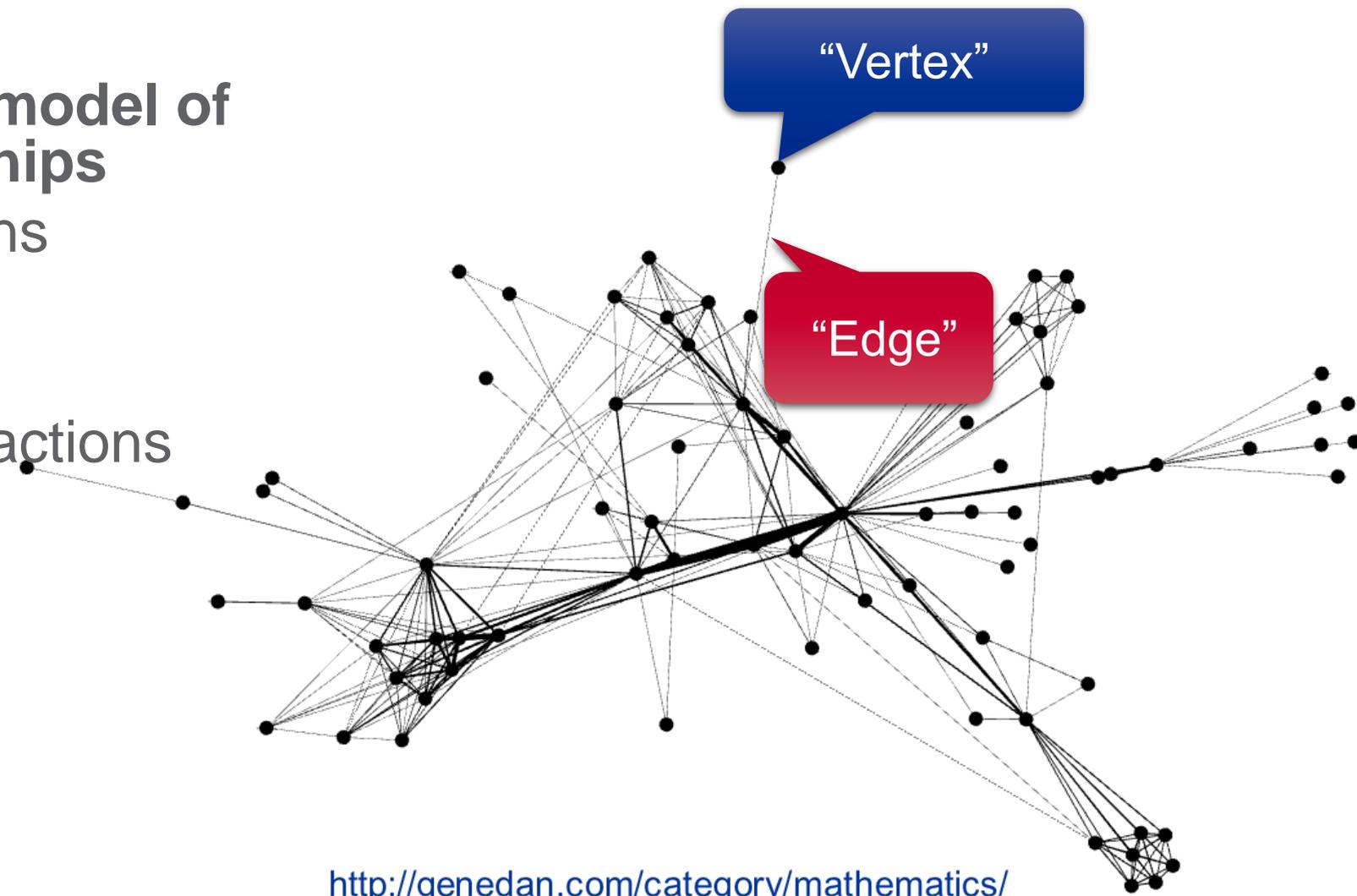
# Complex relational data

- Many real world data sets have complex relational structure
  - Social networks: People x Groups
  - Collaboration: Authors x Papers
  - Transportation: People x Shared transit (planes, trains, and automobiles!)
  - Computer network: Domains x IP addresses ( x MAC addresses x ...)
  - Biology: Proteins x Pathways or Complexes
- Modeling these complex relationships *faithfully* can provide challenges
- Relationships are typically studied using a graph structure – network science
- But this can be lossy
- Enter: ***Hypergraphs*** and ***Hypernetwork science***



# Graphs and Networks

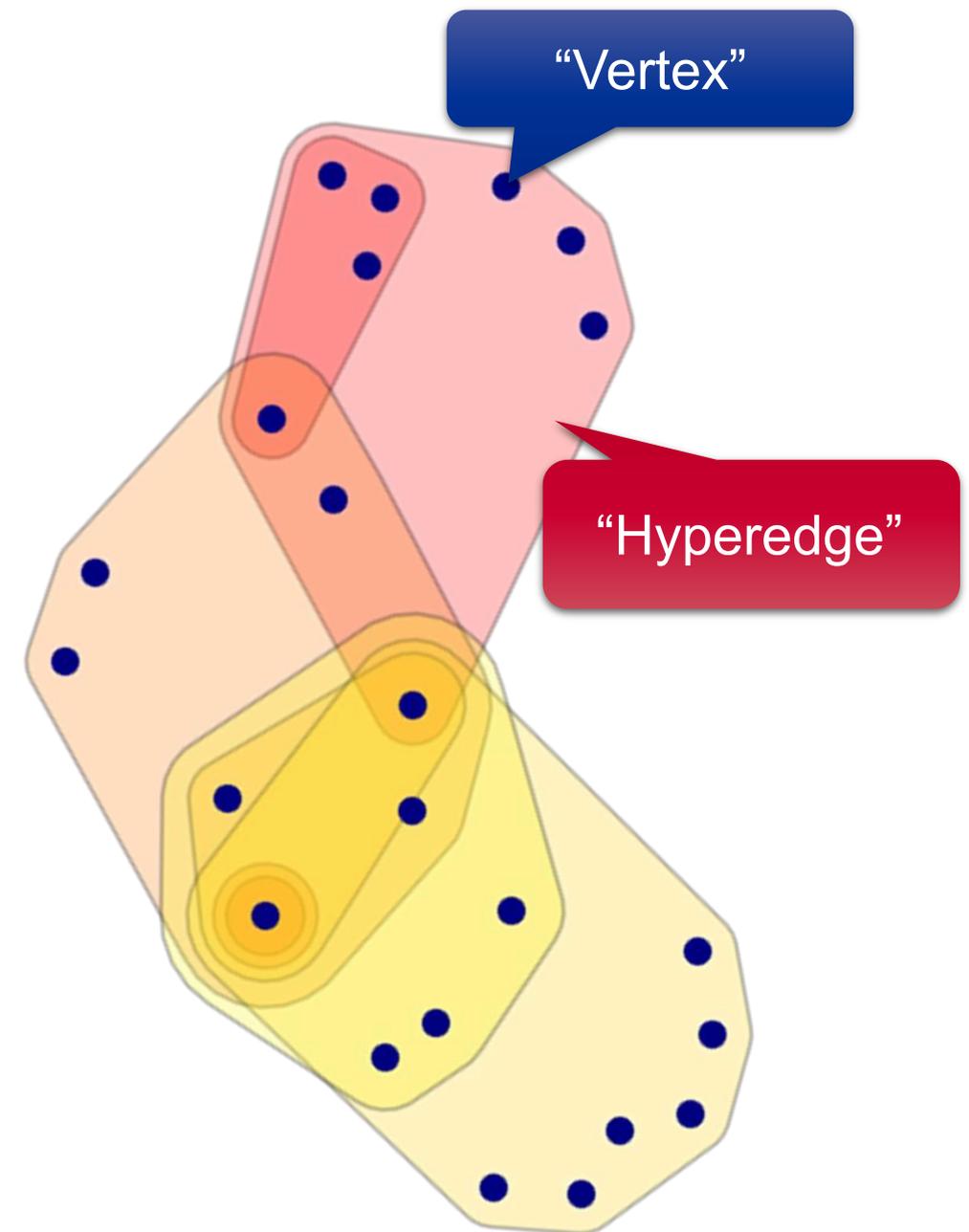
- **Graphs** provide a mathematical model of data focused on **2-way** relationships
  - To *ask* certain kinds of questions
    - ✓ Connectivity of entities
    - ✓ Clustering structure
  - To *model* certain kinds of interactions
    - ✓ Pairwise relationships
- **Applications**
  - Social analysis
  - Cyber
  - Collaboration
  - Proteomics
  - Transportation



<http://genedan.com/category/mathematics/>

# Hypergraphs and Networks

- **Hypergraphs** provide a mathematical model of data focused on **multi-way** relationships
  - To *ask* certain kinds of questions
    - ✓ Connectivity of entities
    - ✓ Clustering structure
  - To *model* certain kinds of interactions
    - ✓ Multi-way relationships
- **Applications**
  - Social analysis
  - Cyber
  - Collaboration
  - Proteomics
  - Transportation



Co-occurrence of characters in Les Miserables, restricted to single character neighborhood. Image generated by HyperNetX.

# What is the use of hypergraphs in data science, given the prevalence of ML / AI?

- Well...



## Hypergraphs

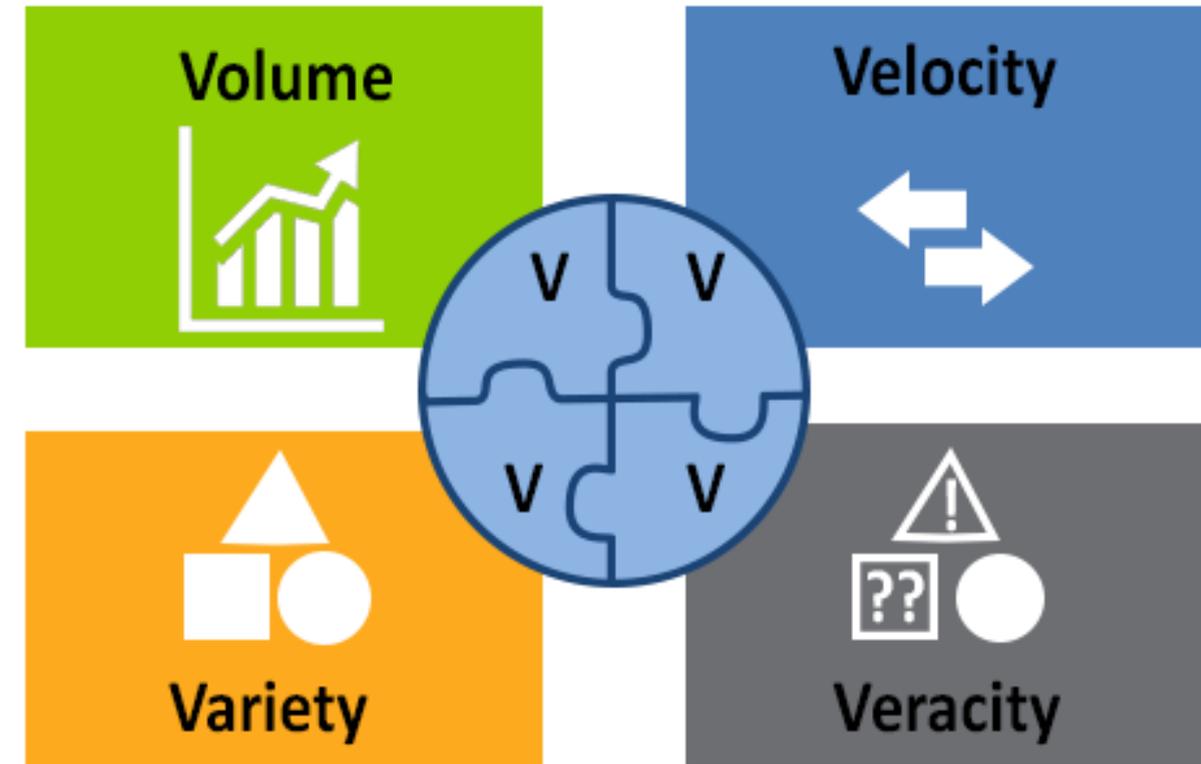
- Provides framework for exploring complex data
- To discover interesting objects, groups
- Combinatorial, deterministic
- To explore and understand data

## ML / AI

- Provides framework for clustering and classifying complex data
- To discover trends, patterns
- Statistical, probabilistic
- To perform a specific task

# What is the use of hypergraphs in data science, given the prevalence of ML / AI?

- Aspects of data that need to be modeled
  - **Volume:** Scaling
  - **Velocity:** Streaming
  - **Veracity:** Uncertainty quantification
  - **Variety:** *Complex data*
- Hypergraphs...
  - Structural model for homogeneous entities in different groups
  - Where homogeneous entities have different attributes
  - *Could* be used to derive features for ML/AI models
  - *Could* be learned from ML/AI methods

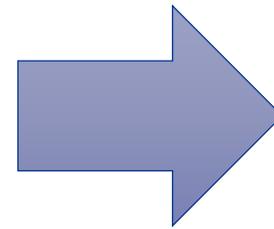


<https://www.zarantech.com/blog/the-4-vs-of-big-data/>

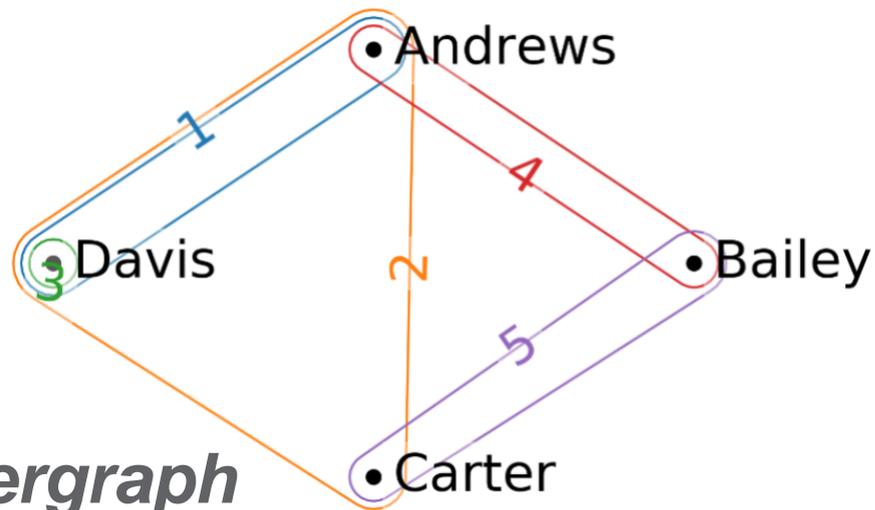
Hypergraphs model the data  
whereas ML / AI reasons about it

# Hypergraphs vs. Graphs

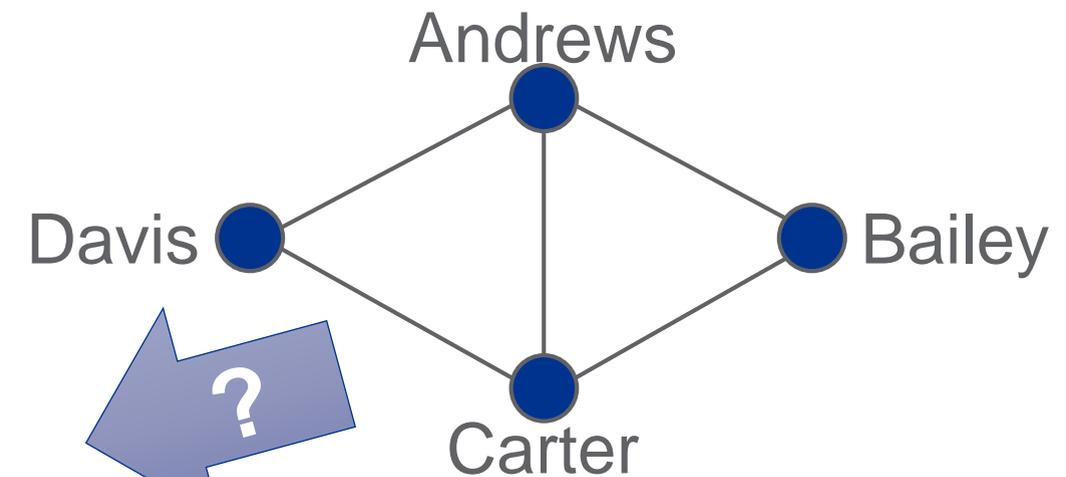
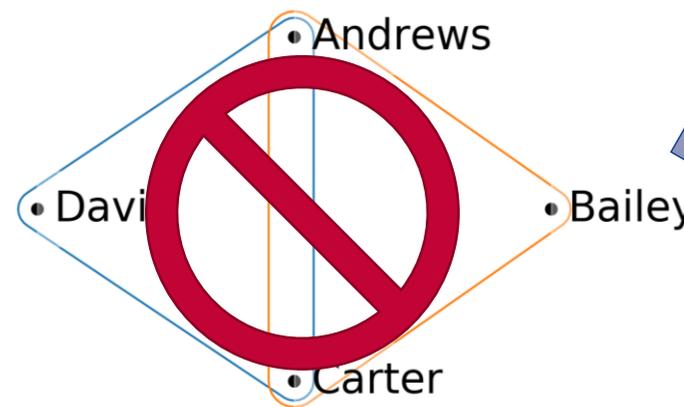
Paper #	Authors
1	Andrews, Davis
2	Andrews, Carter, Davis
3	Davis
4	Andrews, Bailey
5	Bailey, Carter



	Andrews	Bailey	Carter	Davis
Andrews		Y	Y	Y
Bailey	Y		Y	
Carter	Y	Y		Y
Davis	Y		Y	



**Hypergraph**  
representation



**Graph**  
representation

# What kind of Data generate Hypergraphs?

Imagine your tabular data:

- **Attributes:** Entities (rows) are indicated as having specific attributes or properties (columns)

	Supervisor?	Brown Eyes?	Female?	Surgeon?
John	X	X		
Sally	X	X	X	X
Spoe				X
Gertie		X	X	X

- **Joint relationships:** Entities jointly participate in some relationship or activity

	camcountry.com	crowmedicine.com	sonymusicnashville.com	elvisthemusic.com
104.16.236.100	X	X		X
104.16.237.100	X	X		X
104.16.238.100	X	X	X	X
104.16.235.100		X	X	X

Data from <https://activednsproject.org/>

- **Numeric data:** consider thresholding the data

Protein	MERS_WT_0HRS	MERS_WT_12HRS	MERS_WT_24HRS	MERS_WT_36HRS
AAAS	0.089651	-0.081078	-0.136514	-0.268651
AACS	-0.039801	0.344122	-0.970295	0.106712
AADAC	0.017425	-0.018319	0.050566	-0.159475
AAK1	-0.112657	0.235510	0.480794	-0.852893
AAMP	0.021689	0.131453	-0.574124	0.011710

# What kind of Data generate Hypergraphs?

Imagine your tabular data:

- **Attributes:** Entities (rows) are indicated as having specific attributes or properties (columns)

	Supervisor?	Brown Eyes?	Female?	Surgeon?
John	1	1	0	0
Sally	1	1	1	1
Spoe	0	0	0	1
Gertie	0	1	1	1

- **Joint relationships:** Entities jointly participate in some relationship or activity

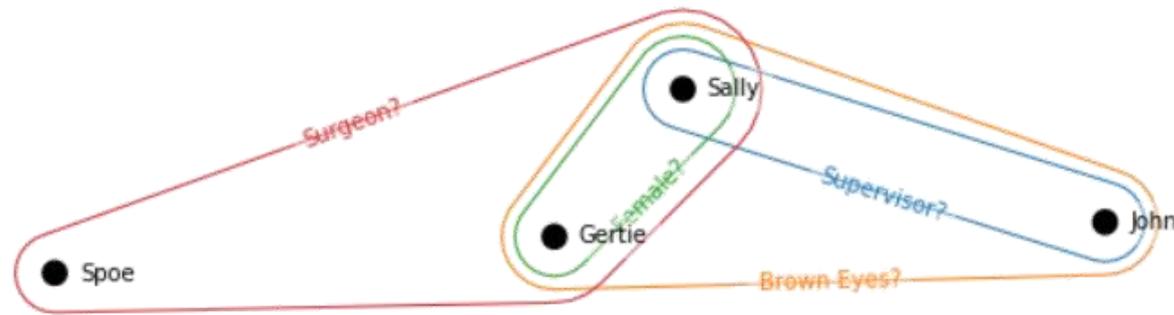
	camcountry.com	crowmedicine.com	sonymusicnashville.com	elvisthemusic.com
104.16.236.100	1	1	1	0
104.16.237.100	1	1	0	1
104.16.238.100	1	1	1	1
104.16.235.100	0	1	1	1

Data from <https://activednsproject.org/>

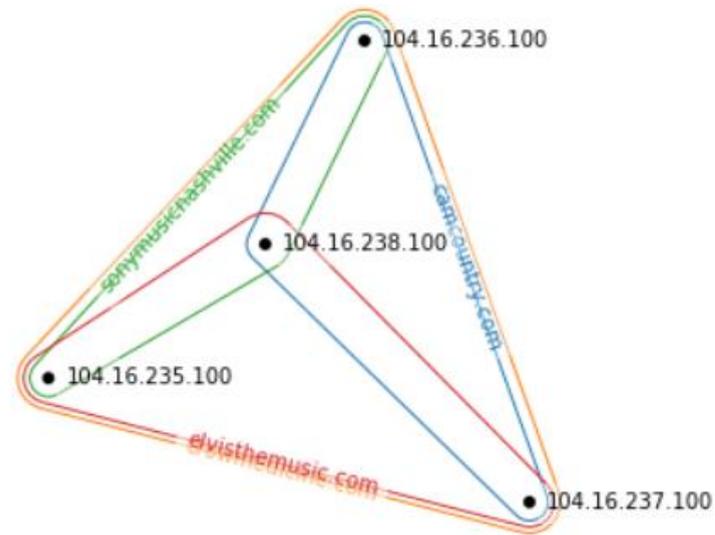
- **Numeric data:** consider thresholding the data  
e.g., cell value > 0

Protein	MERS_WT_0HRS	MERS_WT_12HRS	MERS_WT_24HRS	MERS_WT_36HRS
AAAS	1	0	0	0
AACS	0	1	0	1
AADAC	1	0	1	0
AAK1	0	1	1	0
AAMP	1	1	0	1

# What kind of Data generate Hypergraphs?

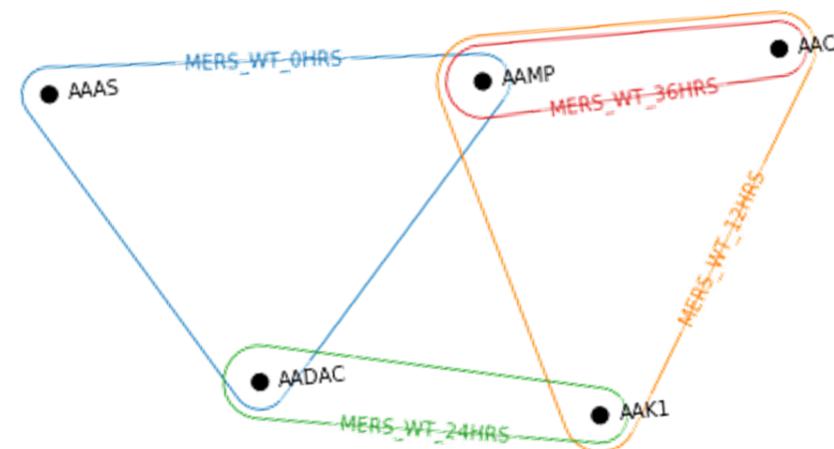


	Supervisor?	Brown Eyes?	Female?	Surgeon?
John	1	1	0	0
Sally	1	1	1	1
Spoe	0	0	0	1
Gertie	0	1	1	1



	camcountry.com	crowmedicine.com	sonymusicnashville.com	elvisthemusic.com
104.16.236.100	1	1	1	0
104.16.237.100	1	1	0	1
104.16.238.100	1	1	1	1
104.16.235.100	0	1	1	1

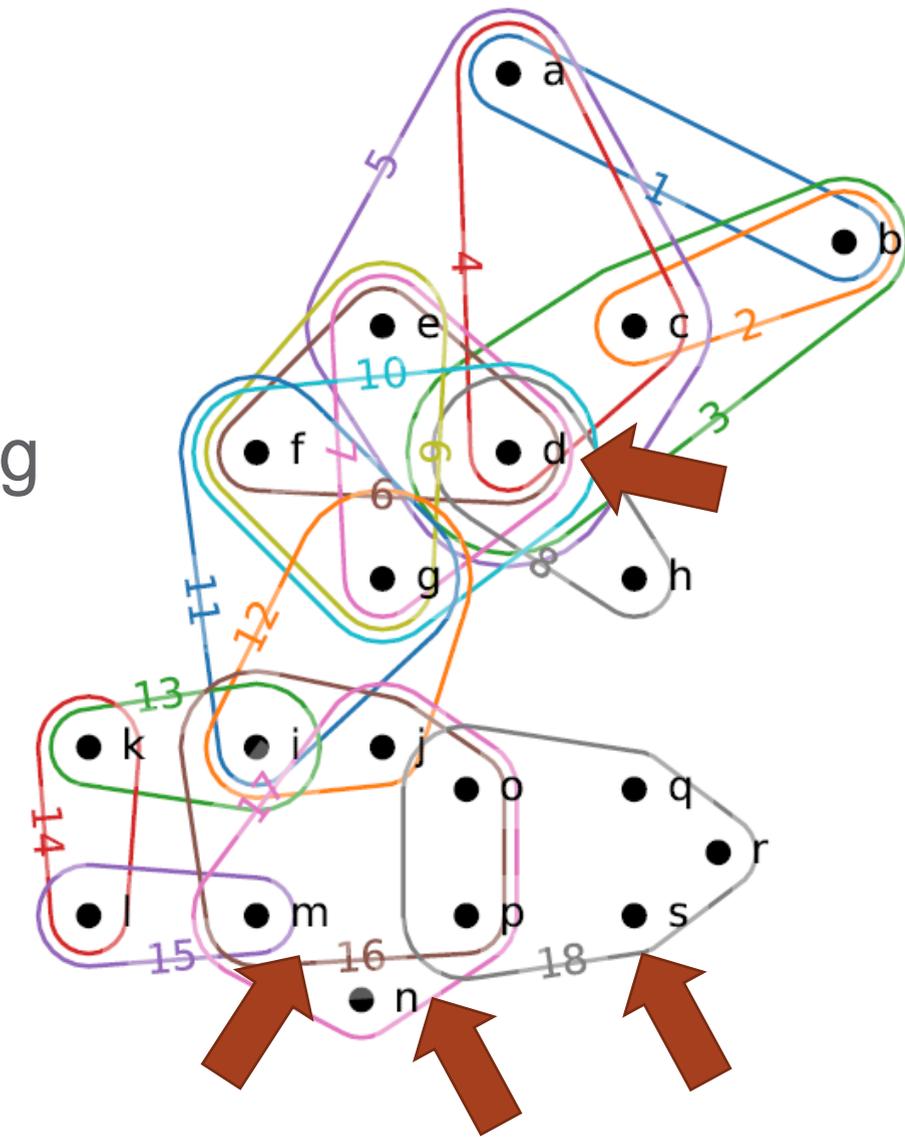
Data from <https://activednsproject.org/>



Protein	MERS_WT_0HRS	MERS_WT_12HRS	MERS_WT_24HRS	MERS_WT_36HRS
AAAS	1	0	0	0
AACS	0	1	0	1
AADAC	1	0	1	0
AAK1	0	1	1	0
AAMP	1	1	0	1

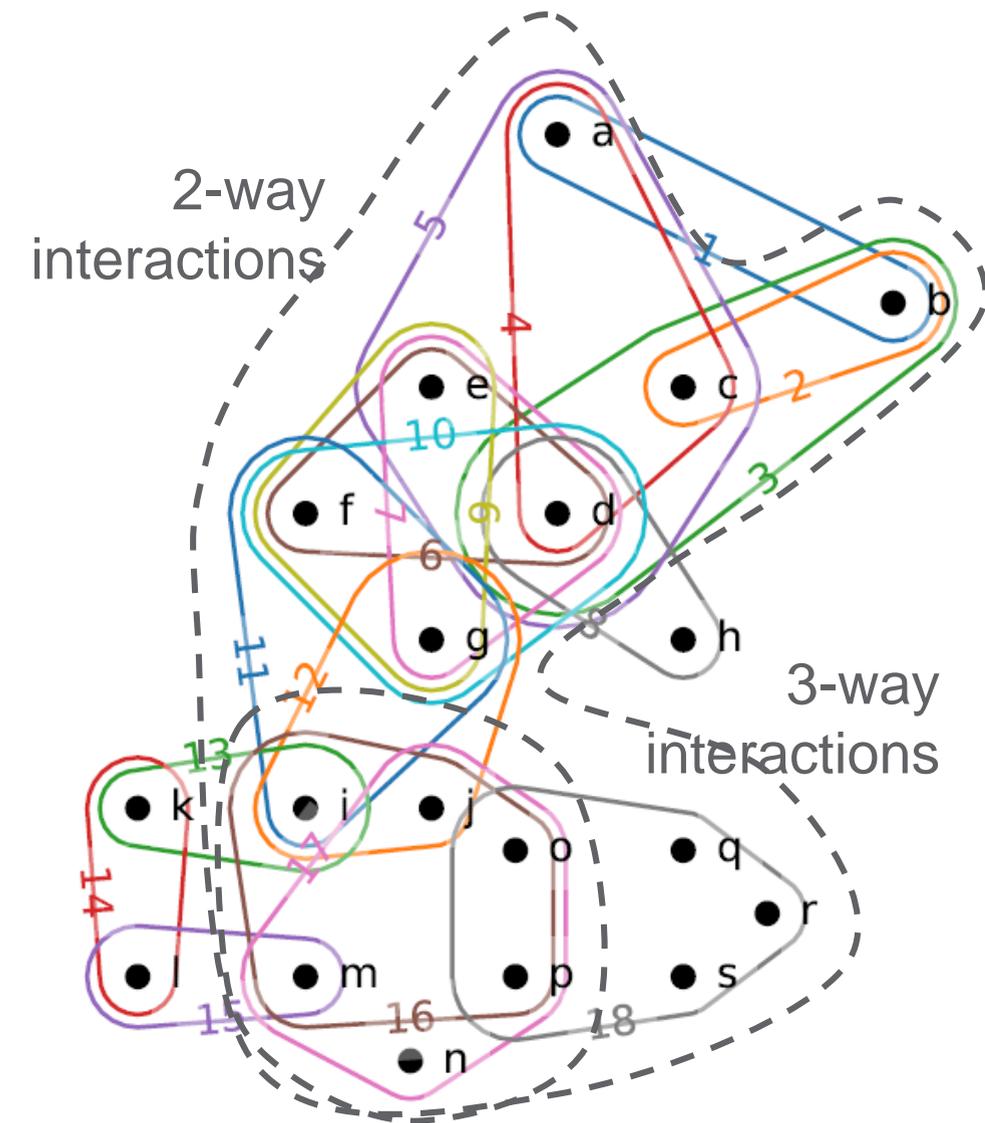
# Towards Hypernetwork Science

- Hypergraphs generalize network science to multi-way relationships
  - For question of *community interaction*
  - *Multidimensional* connectivity, centrality, etc. among *groups* of entities
- Who are most active authors? *Max node degrees*
- Which papers have most authors? *Max edge sizes*



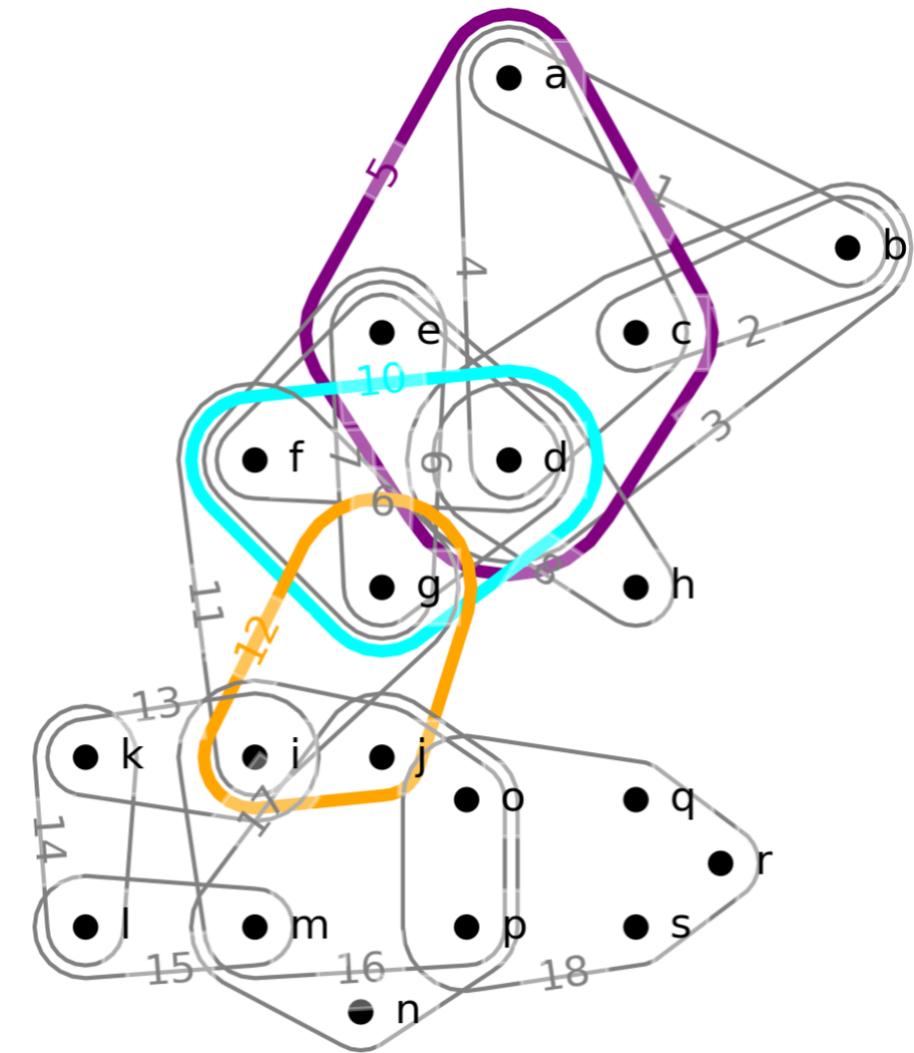
# More Hypernetwork Science

- **What research communities are formed?**  
*Connected components of different strengths*



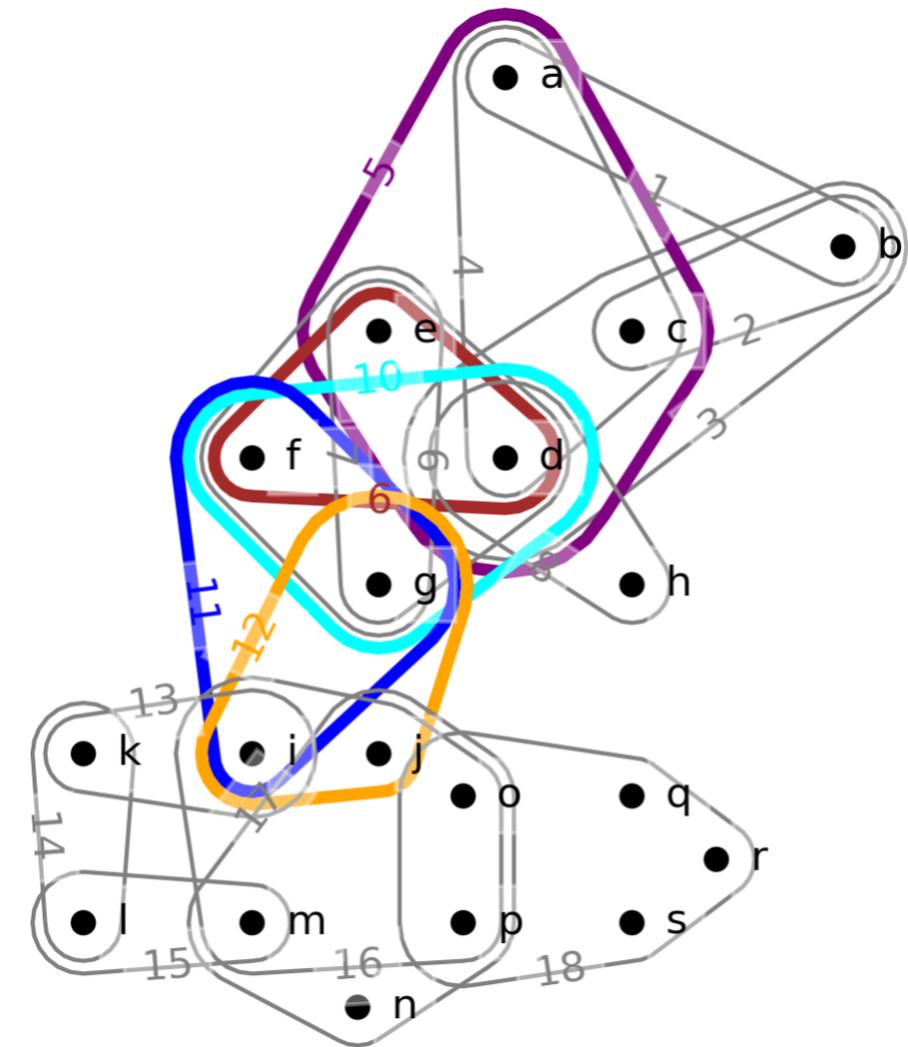
# More Hypernetwork Science

- **What research communities are formed?**  
*Connected components of different strengths*
- **How many collaborations are there between some pair of papers? “s-Distance”**
  - **5** → **10** → **12** (*intersections=1*)



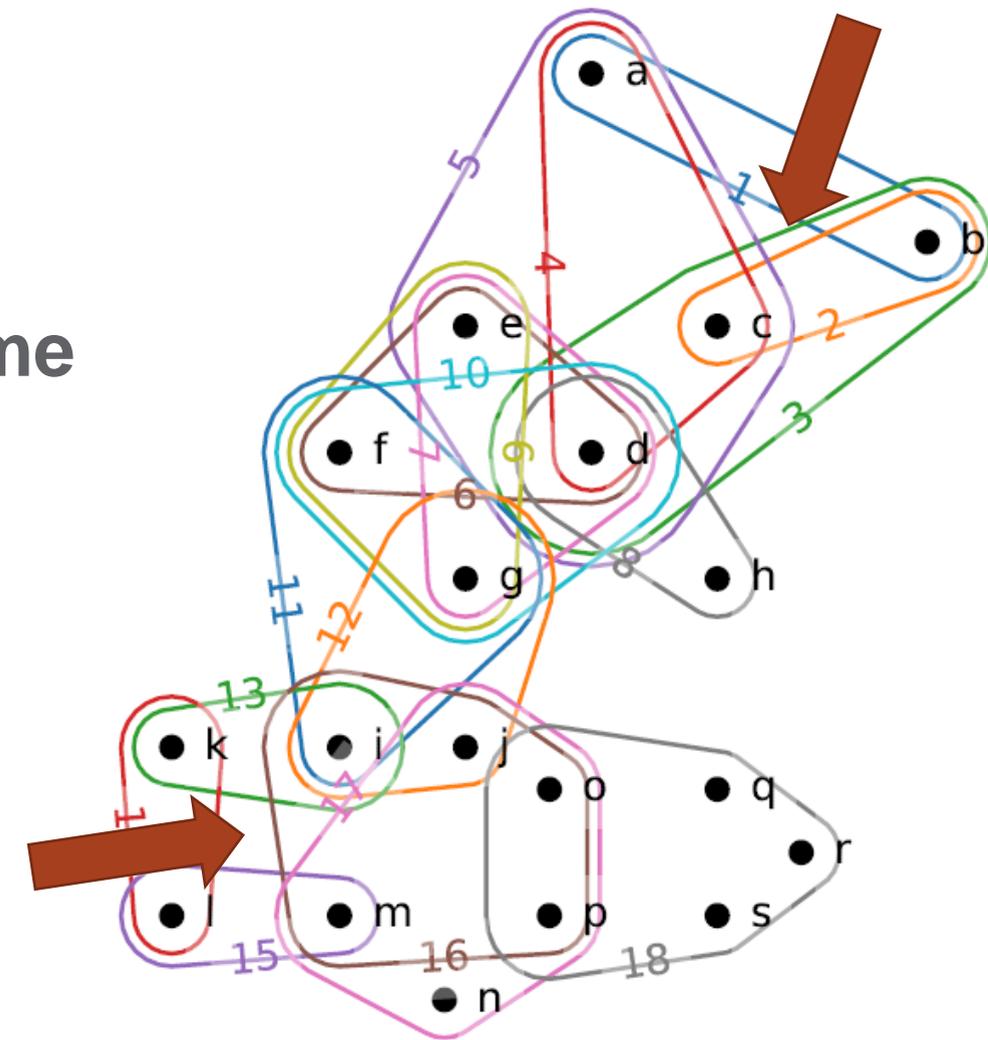
# More Hypernetwork Science

- **What research communities are formed?**  
*Connected components of different strengths*
- **How many collaborations are there between some pair of papers? “s-Distance”**
  - **5** → **10** → **12** (*intersections=1*)
  - **5** → **6** → **10** → **11** → **12** (*intersections=2*)



# More Hypernetwork Science

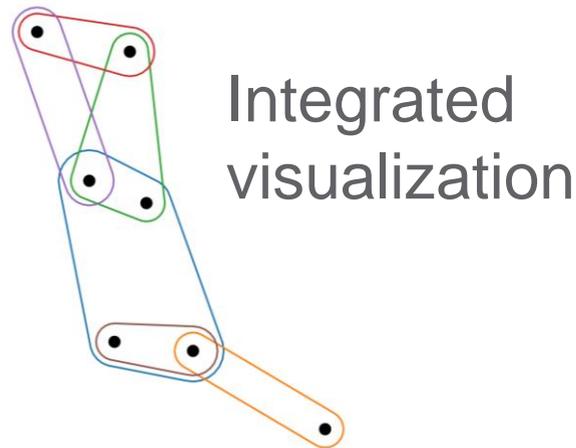
- **What research communities are formed?**  
*Connected components of different strengths*
- **How many collaborations are there between some pair of papers? “s-Distance”**
  - $5 \rightarrow 10 \rightarrow 12$  (*intersections=1*)
  - $5 \rightarrow 6 \rightarrow 10 \rightarrow 11 \rightarrow 12$  (*intersections=2*)
- **What is the most distant pair of papers?**  
*“s-Diameter”*
- **Are there groups of authors who aren’t working together but should?**  
*Homology, Betti numbers*
  - “Holes as hypotheses”



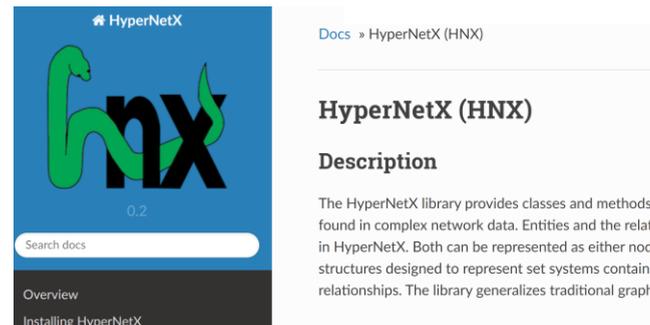
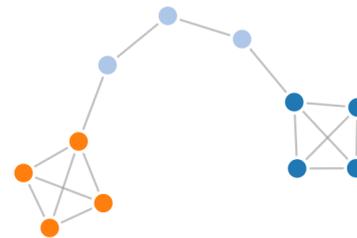
# The HyperNetX (HNX) Library



- HNX has various hypergraph constructors for: dictionaries, lists of lists, bipartite graphs...



Builds on  
NetworkX



Install  
with PyPI

Interactive tutorials  
 jupyter

- Core Requirements:

Python  $\geq 3.6$   
NetworkX  
Numpy

SciPy  
Matplotlib  
Jupyter (for tutorials)

- Current scale:

- Hypergraph exploration for  $O(10K)$  vertices and hyperedges
- Experimenting with CuPy for scaling up

- Open Sourced on Github:

<https://github.com/pnnl/HyperNetX>

- Developers: Brenda Praggastis (lead), Dustin Arendt (visualizations)

# Chapel Hypergraph Library



<https://github.com/pnnl/chgl/>

- *Scalable parallel computation methodologies for complex high-dimensional data objects*
- **Chapel:** Experimental data-parallel HPC language
- **Hypergraph Analytics:**
  - Abstract hypergraph data structures (concepts)
  - Abstract hypergraph algorithms (based on concepts)
  - Mapping to families of efficient asynchronous parallel implementations
- **HPC-Grade Chapel Implementation**
  - Use the high-level features of Chapel to implement an HPC library
  - Based on three principles: Genericity, Performance, Usability
  - Current state: Many working algorithms, privatized distributed adjacency list data structure, aggregation library (CAL), and more



The Overview

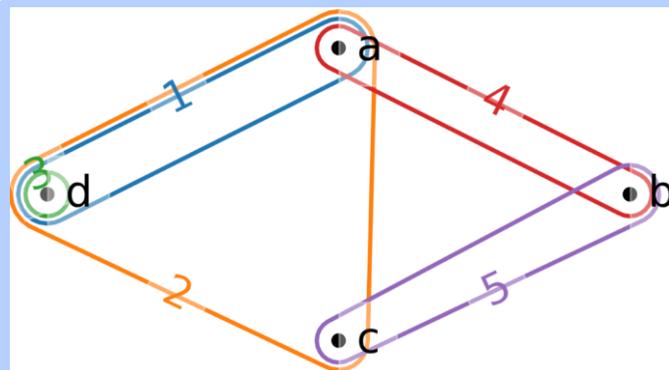
# The Math

The Applications

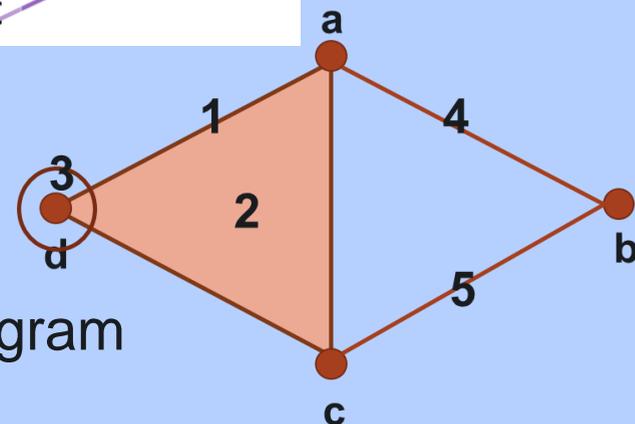
Future Directions

# Hypergraph definition

- **Hypergraph ( $H$ ):** vertices  $V$ , family of edges  $E = (e_1, \dots, e_m)$ ,  $e_i \subseteq V$ .
  - A **graph  $G$**  is a hypergraph where all edges have size 2
- Special cases: empty and multi-edges, isolated and redundant vertices

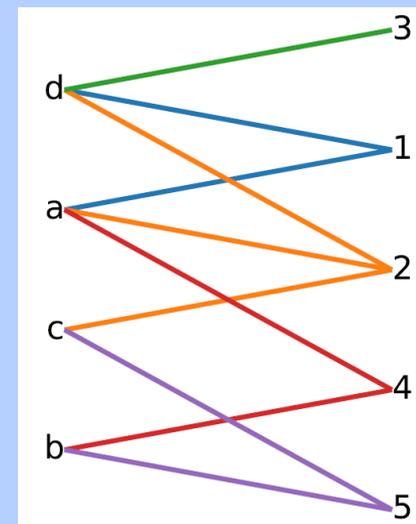


Euler diagram



Simplicial diagram

Bipartite diagram



Incidence matrix

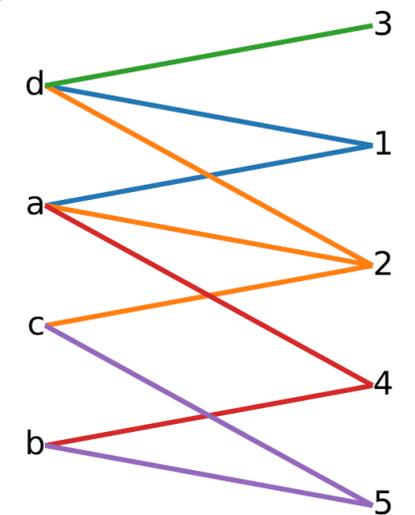
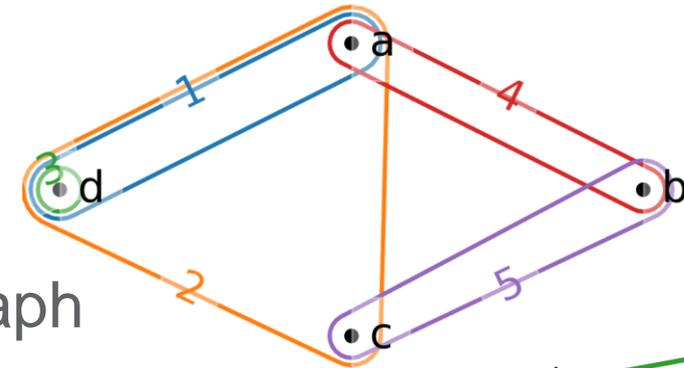
$$\begin{matrix} a \\ b \\ c \\ d \end{matrix} \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

$$H = \{acd, ab, bc, ad, d\}$$

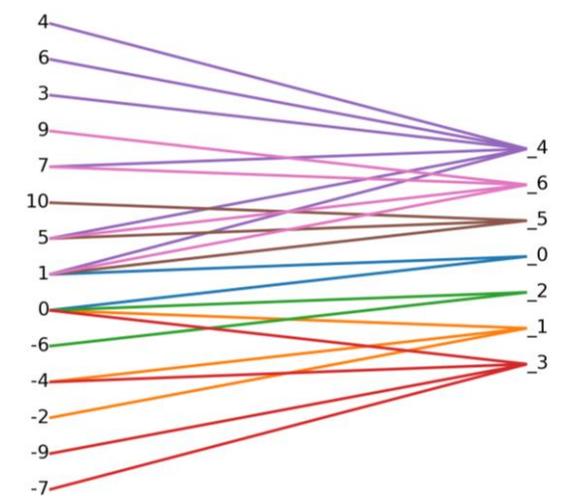
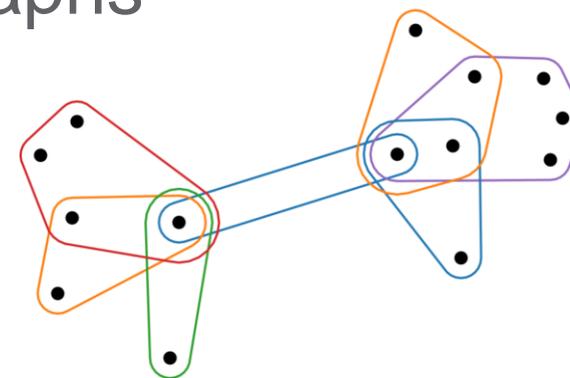
Compact set notation

# Digression: Hypergraphs vs Bipartite graphs

- Hypergraphs  $\neq$  Bipartite graphs!
  - Q: What goes wrong?
  - A: There are at least two colorings of each bipartite graph  
( $2^{\#}$  connected components)



- Bipartite graph + coloring = *Bicolored graph*
- Hypergraphs  $\cong$  Bicolored graphs
- But, complex interaction amongst hyperedges not as easily understood in the language of bicolored graphs



# Walks on edges or vertices?

– For graphs, essentially the same.

- Each pair of vertices in  $G$  belong to at most 1 edge, so:

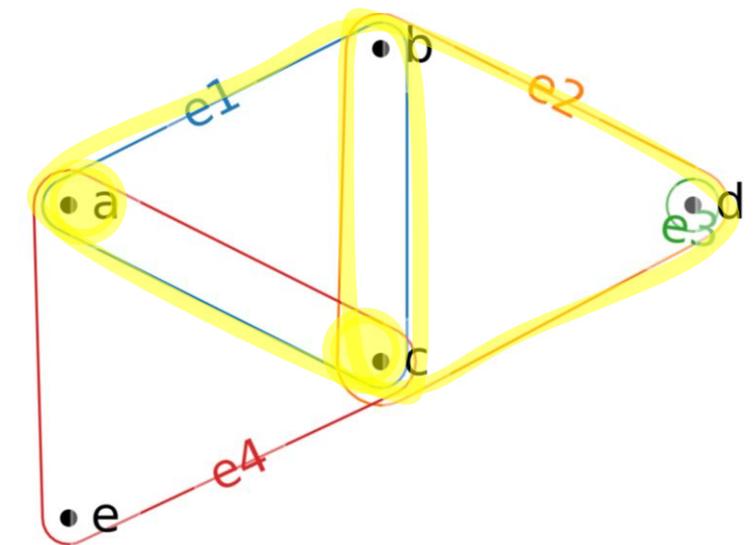
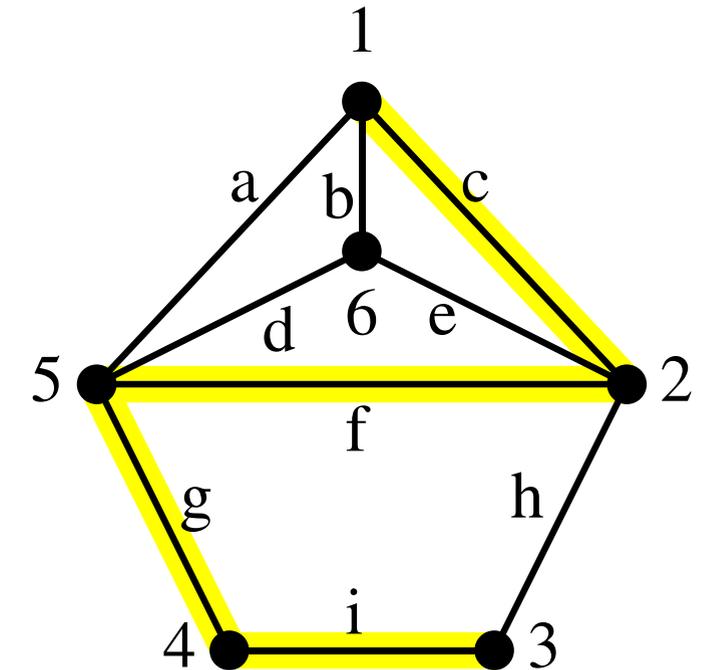
$$\underbrace{v_0, v_1}_{\text{adjacent}}, \dots, \underbrace{v_{k-1}, v_k}_{\text{adjacent}} \rightarrow \underbrace{e_1}_{\{v_0, v_1\}}, \dots, \underbrace{e_k}_{\{v_{k-1}, v_k\}}$$

- Each pair of edges in  $G$  intersect in at most 1 vertex, so:

$$\underbrace{e_1, e_2}_{\text{incident}}, \dots, \underbrace{e_{k-1}, e_k}_{\text{incident}} \rightarrow \underbrace{v_0}_{e_1 \setminus e_2}, \underbrace{v_1}_{e_1 \cap e_2}, \dots, \underbrace{v_{k-1}}_{e_{k-1} \cap e_k}, \underbrace{v_k}_{e_k \setminus e_{k-1}}$$

– For hypergraphs, not the same.

- Each pair of vertices can belong to many edges.
- Each pair of edges can intersect at many vertices.



**Walks between edges:** sequence of successively intersecting edges ← Our focus  
**Walks between vertices:** sequence of successively adjacent vertices

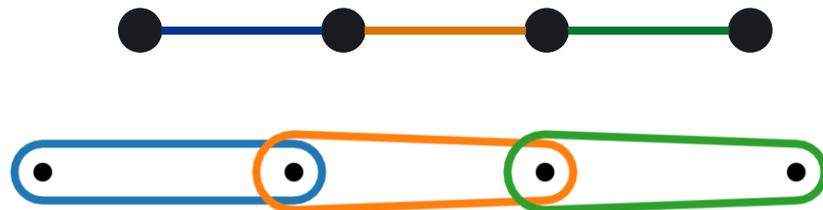
# Hypergraph paths have width

- **s-Walk:** sequence of edges  $e_1, \dots, e_k$  such that  $|e_i \cap e_{i+1}| \geq s$
- Paths in hypergraphs have width in addition to length:

## A 2-Uniform Hypergraph Path:

(Edgewise) Length = 2

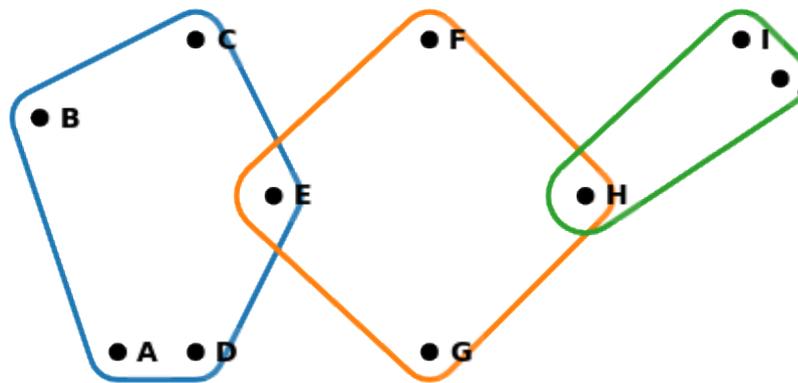
Width = 1



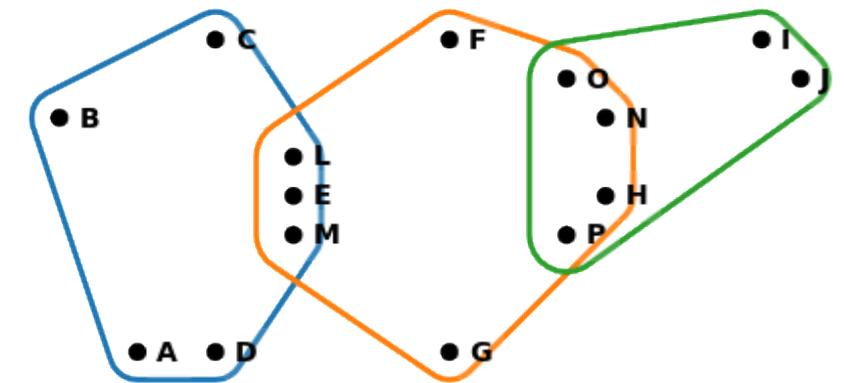
As a 2-uniform HG

## Two Hypergraph Paths:

Same length = 2



**Weak interactions:** Width=1



**Strong interactions:** Width=3

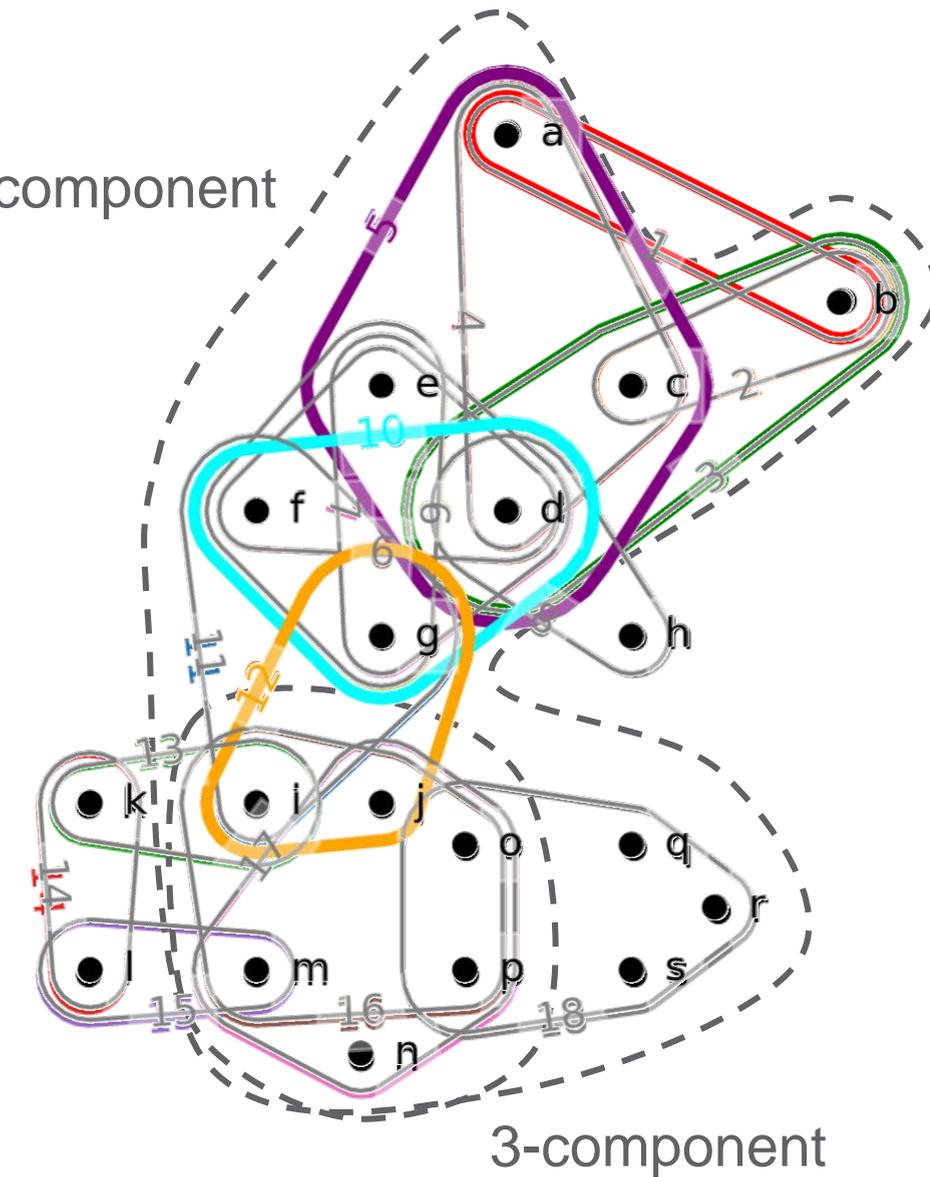
- From s-walks we can define...
  - s-Path
  - s-Distance, s-Diameter

- s-Connected components
- s-Centrality
- s-Clustering coefficient

# s-Distance and s-Components

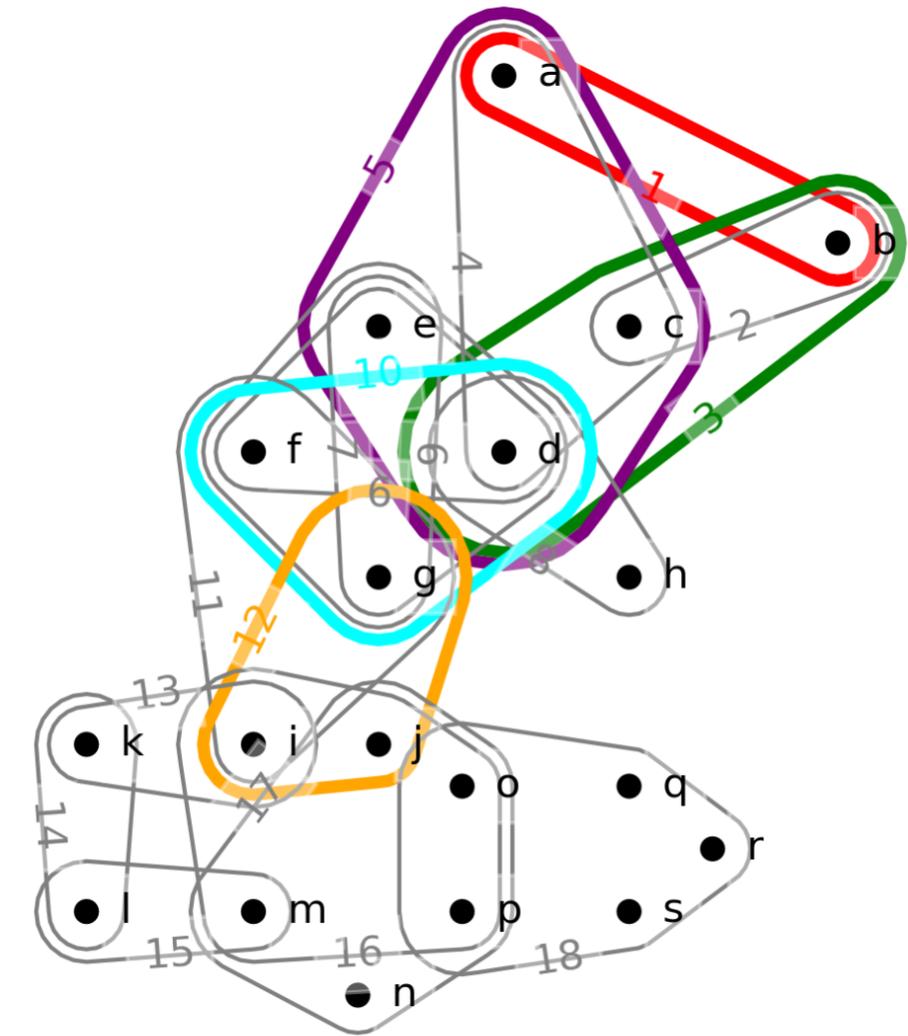
- s-Walk  $\rightarrow$  s-Path  $\rightarrow$  s-Distance and s-Components
- s-Path = s-Walk where edges are not repeated
- s-Distance = shortest s-path between two edges
- s-Component = collection of edges so that any pair are connected via an s-Path

2-component



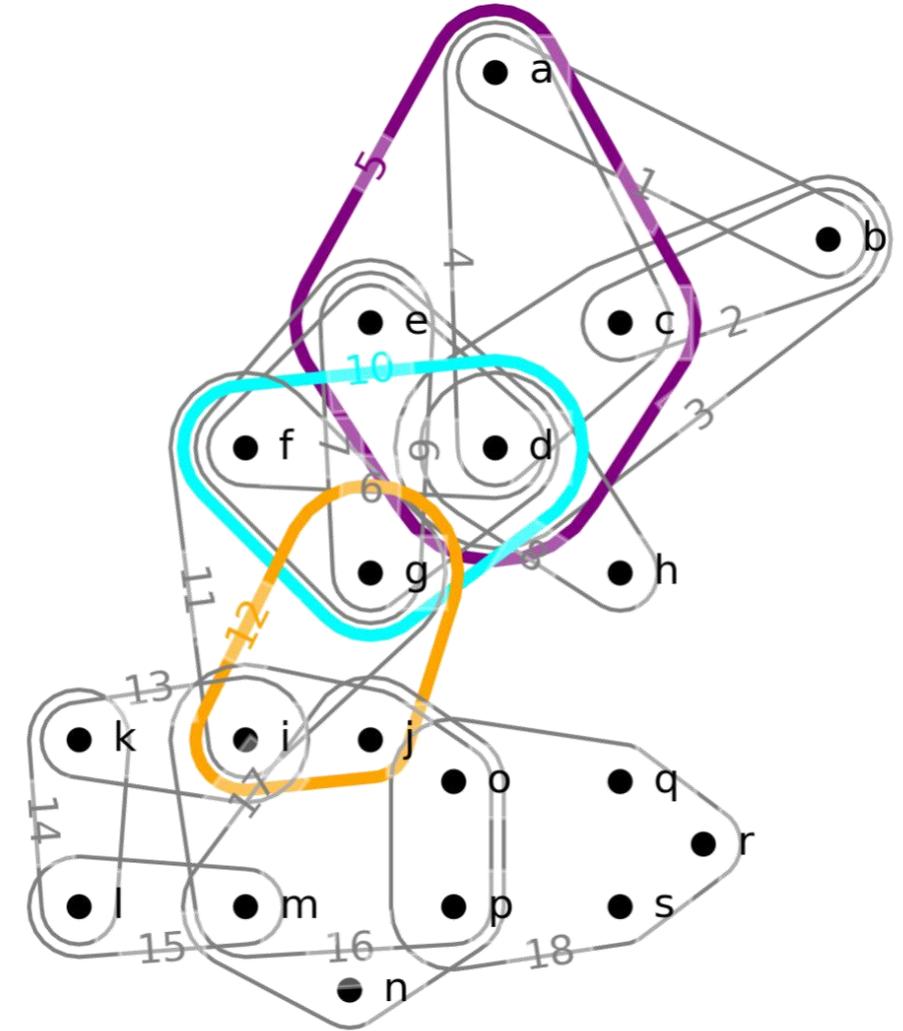
# s-Distance and s-Components

- s-Walk  $\rightarrow$  s-Path  $\rightarrow$  s-Distance and s-Components
- s-Path = s-Walk where edges are not repeated



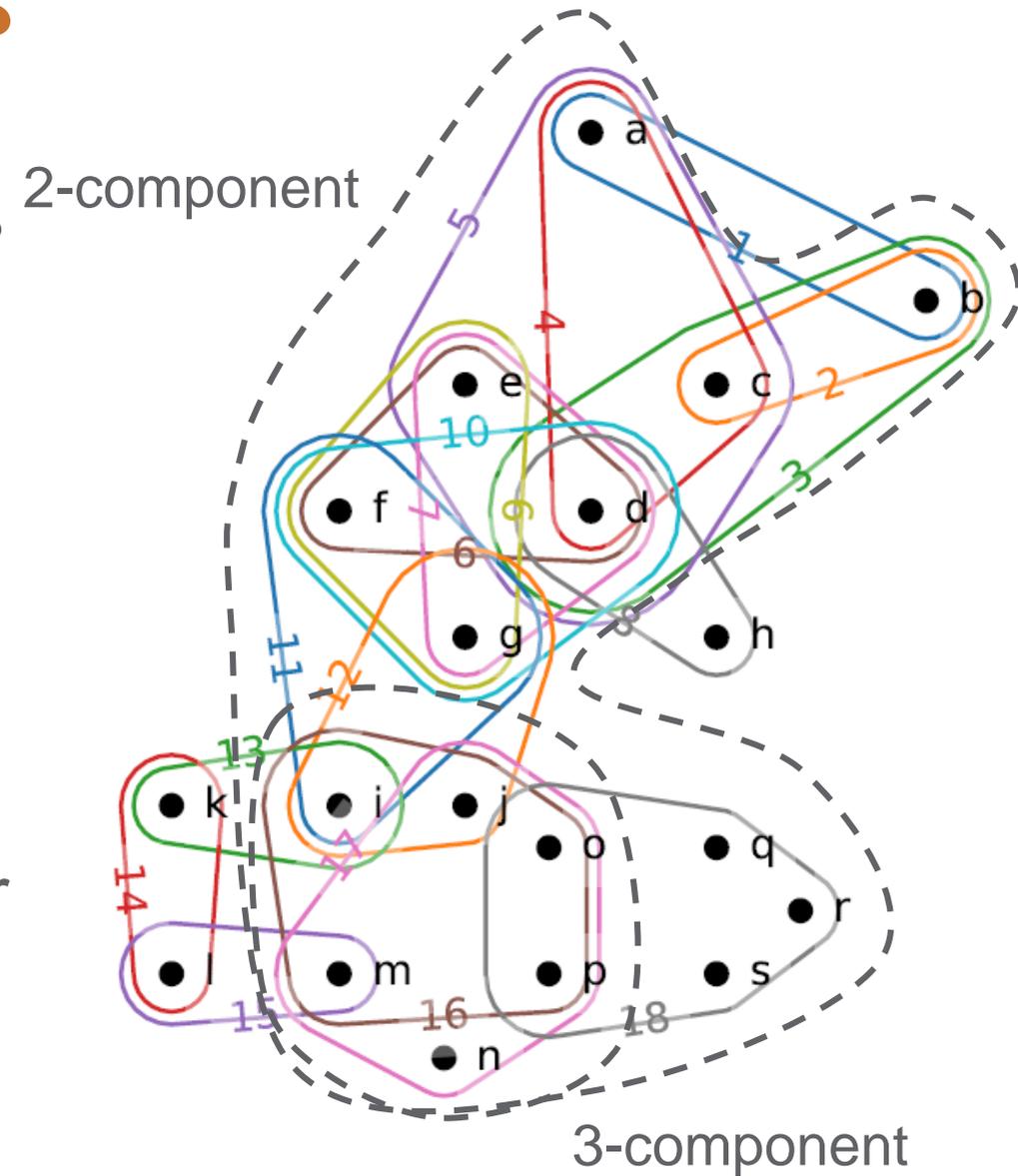
# s-Distance and s-Components

- s-Walk  $\rightarrow$  s-Path  $\rightarrow$  s-Distance and s-Components
- s-Path = s-Walk where edges are not repeated
- s-Distance = shortest s-path between two edges



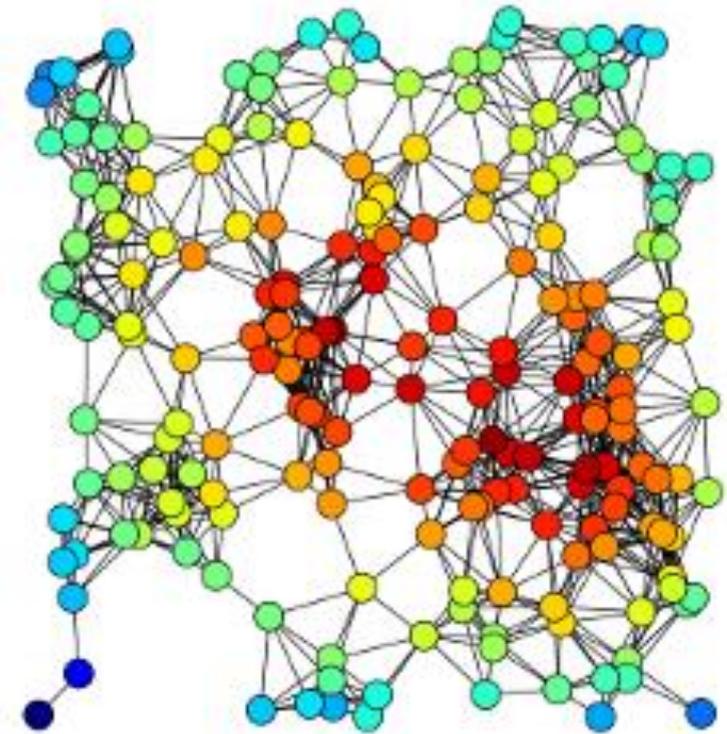
# s-Distance and s-Components

- s-Walk  $\rightarrow$  s-Path  $\rightarrow$  s-Distance and s-Components
- s-Path = s-Walk where edges are not repeated
- s-Distance = shortest s-path between two edges
- s-Component = collection of edges so that any pair are connected via an s-Path



# s-Closeness centrality

- **Question:** Which nodes or edges are “close” to everything?



## Graphs

### Closeness Centrality

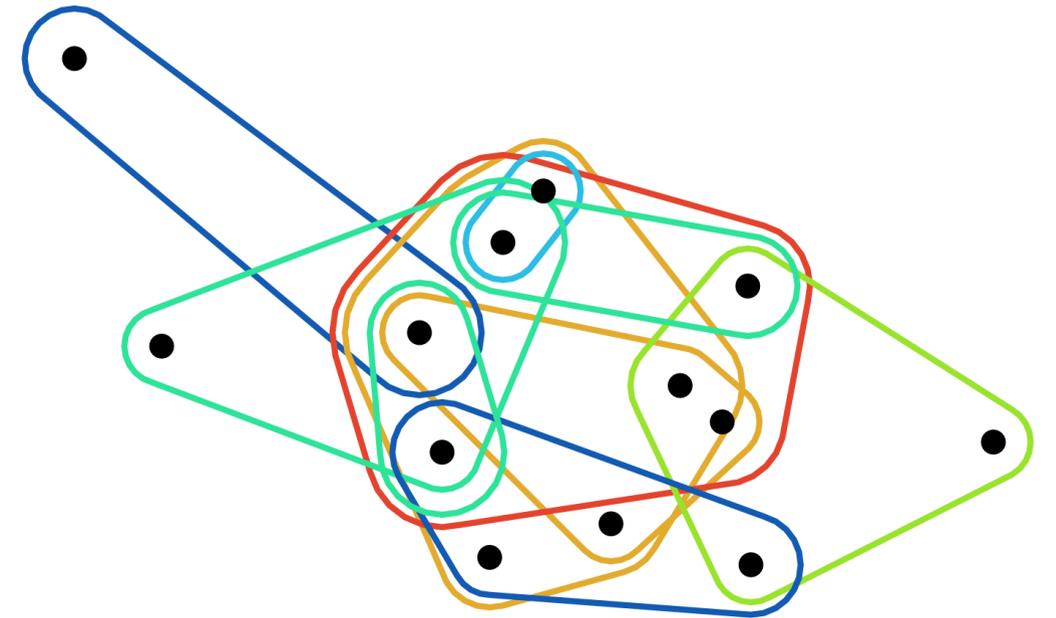
$$C(v) = \frac{|V| - 1}{\sum_{u \in V} d(v, u)}$$

### Harmonic Closeness Centrality

$$HC(v) = \frac{1}{|V| - 1} \sum_{u \in V} \frac{1}{d(v, u)}$$

# s-Closeness centrality

- **Question:** Which nodes or edges are “close” to everything?



## Graphs

### Closeness Centrality

$$C(v) = \frac{|V| - 1}{\sum_{u \in V} d(v, u)}$$

### Harmonic Closeness Centrality

$$HC(v) = \frac{1}{|V| - 1} \sum_{u \in V} \frac{1}{d(v, u)}$$

## Hypergraphs

$$E_s = \{e \in E : |e| \geq s\}$$

$$C_s(e) = \frac{|E_s| - 1}{\sum_{f \in E_s} d_s(e, f)}$$

$$HC_s(e) = \frac{1}{|E_s| - 1} \sum_{f \in E_s} \frac{1}{d_s(e, f)}$$



The Overview

The Math

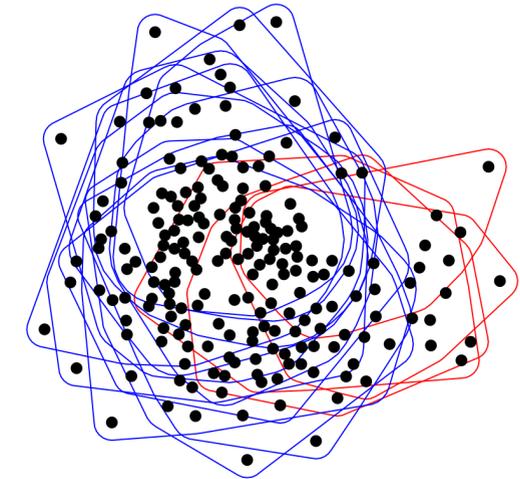
# The Applications

Future Directions

# Three hypergraph data science vignettes

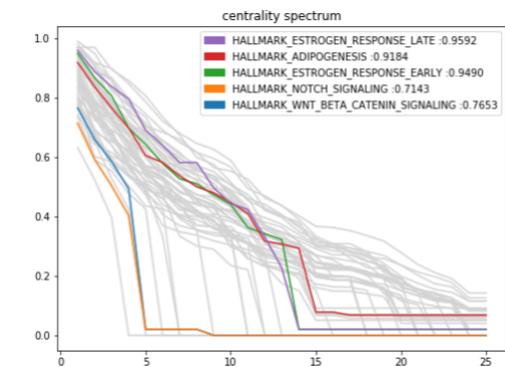
## 1. Pathogen host response data:

- s-Components to discover interconnectedness within and across pathogens
- s-Centrality to discover important genes



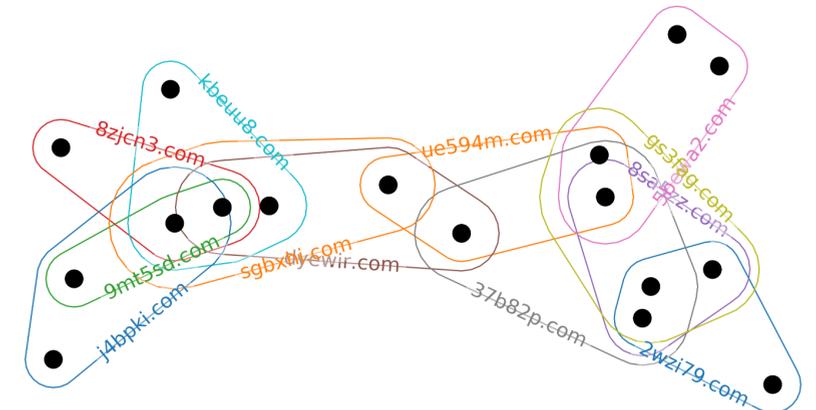
## 2. RNA Transcription factor data:

- s-Centrality to find behavior changes



## 3. Computer network defense:

- Modeling DNS: domain  $\leftrightarrow$  IP relationships
- Discovering motifs

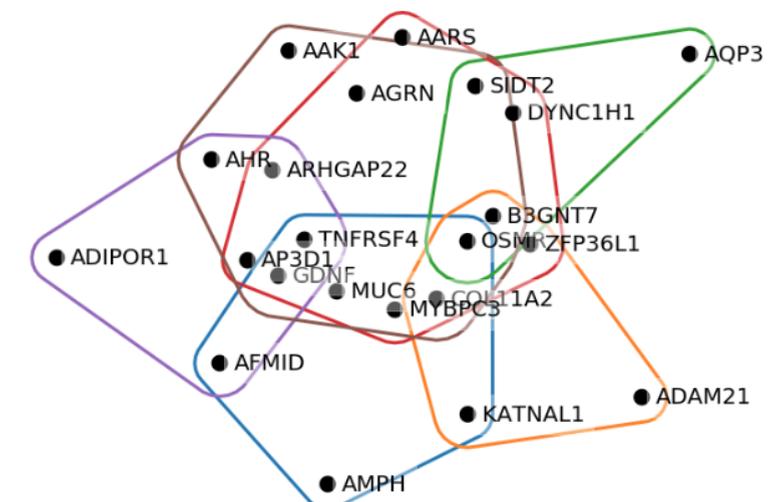


# Example Target Questions

- Do different viruses (e.g. SARS and MERS) perturb shared sets of genes or proteins?
- What genes or proteins are most central in a host response network?
- How are genes shared across multiple pathways?
- Are there patterns of protein activity that vary with degree of viral pathogenicity?
- What are unique patterns elicited by different viruses?
- Are signaling pathways activated in similar ways by different viruses?

Protein	EB1_WT_0h_b	EB1_WT_00h_b	EB1_WT_8h_b
AAAS	-0.053539	-0.021629	0.069450
AACS	0.031504	0.131252	0.309998
AADAC	-0.041660	0.031732	-0.106712
AAK1	0.139425	0.148185	0.251240
AAMP	0.139837	0.005684	0.077773
AARS	0.011162	0.106871	0.202427
AARS2	0.120218	0.040178	0.193941
		⋮	

```
In [25]: 1 Hdf = hnx.from_dataframe(df)
2 noborder(12,12)
3 hnx.draw(Hdf,with_edge_labels=False,
4 with_node_labels=True,
5 node_labels_kwargs={'fontsize':20},
6 edges_kwargs={'edgecolors': plt.cm.tab10(np.arange(len(Hdf.edges))%10),
7 'linewidths': 3},
8 )
```



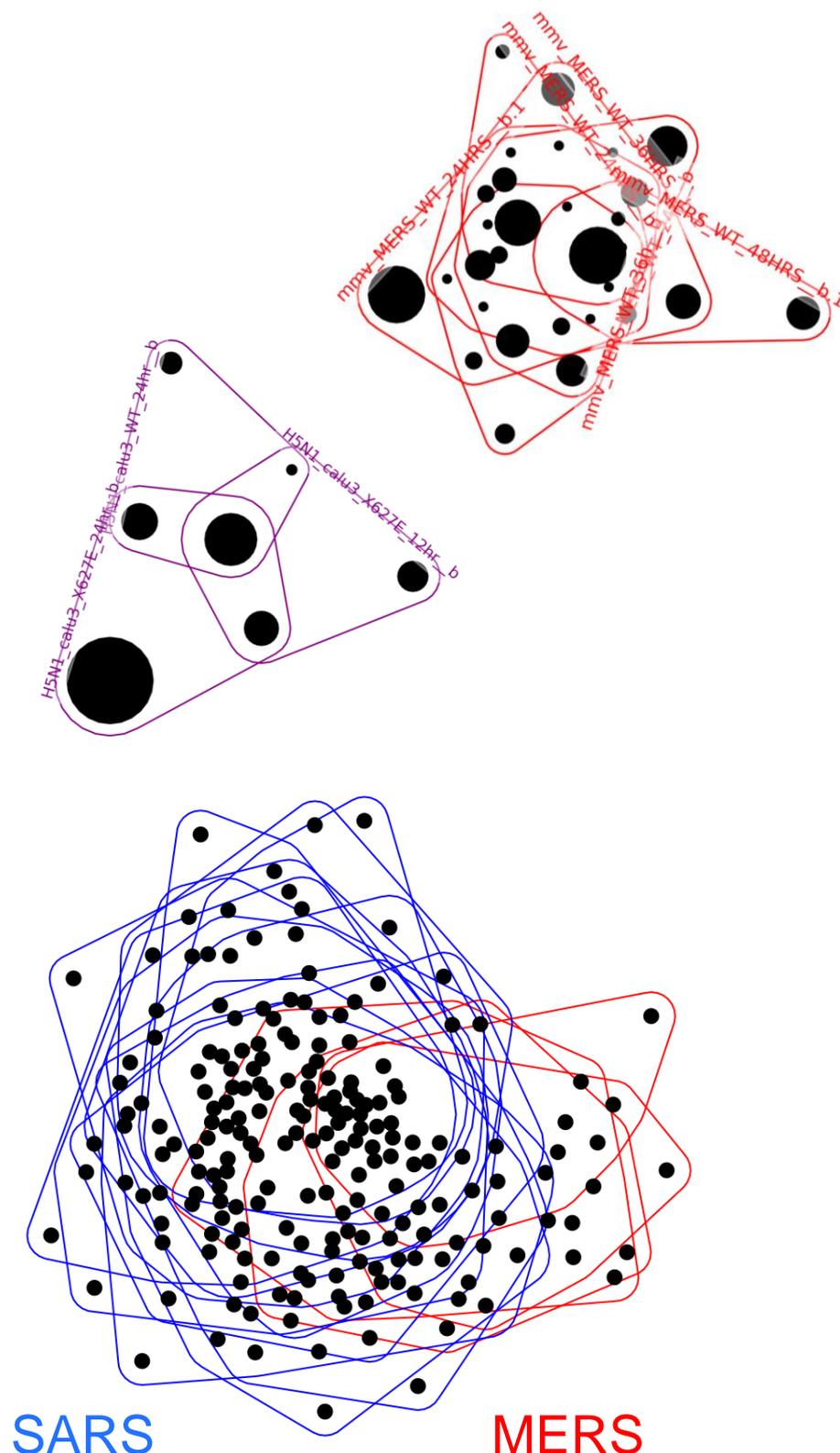
# Host Response Data

- Mouse and human cells infected with viral strains
  - Ebola, Influenza, MERS, SARS, West Nile
- Samples analyzed at time points post infection
- Proteomics and transcriptomics data: Log(fold change) for each [sample, gene] pair vs control
- **Hypergraph:**
  - Nodes = genes
  - Edges = samples
  - Node/edge containment = genes with log(fold change) z-score  $\geq 3$  for a given sample

Protein	EB1_WT_0h_b	EB1_WT_00h_b	EB1_WT_8h_b
AAAS	-0.053539	-0.021629	0.069450
AACS	0.031504	0.131252	0.309998
AADAC	-0.041660	0.031732	-0.106712
AAK1	0.139425	0.148185	0.251240
AAMP	0.139837	0.005684	0.077773
AARS	0.011162	0.106871	0.202427
AARS2	0.120218	0.040178	0.193941

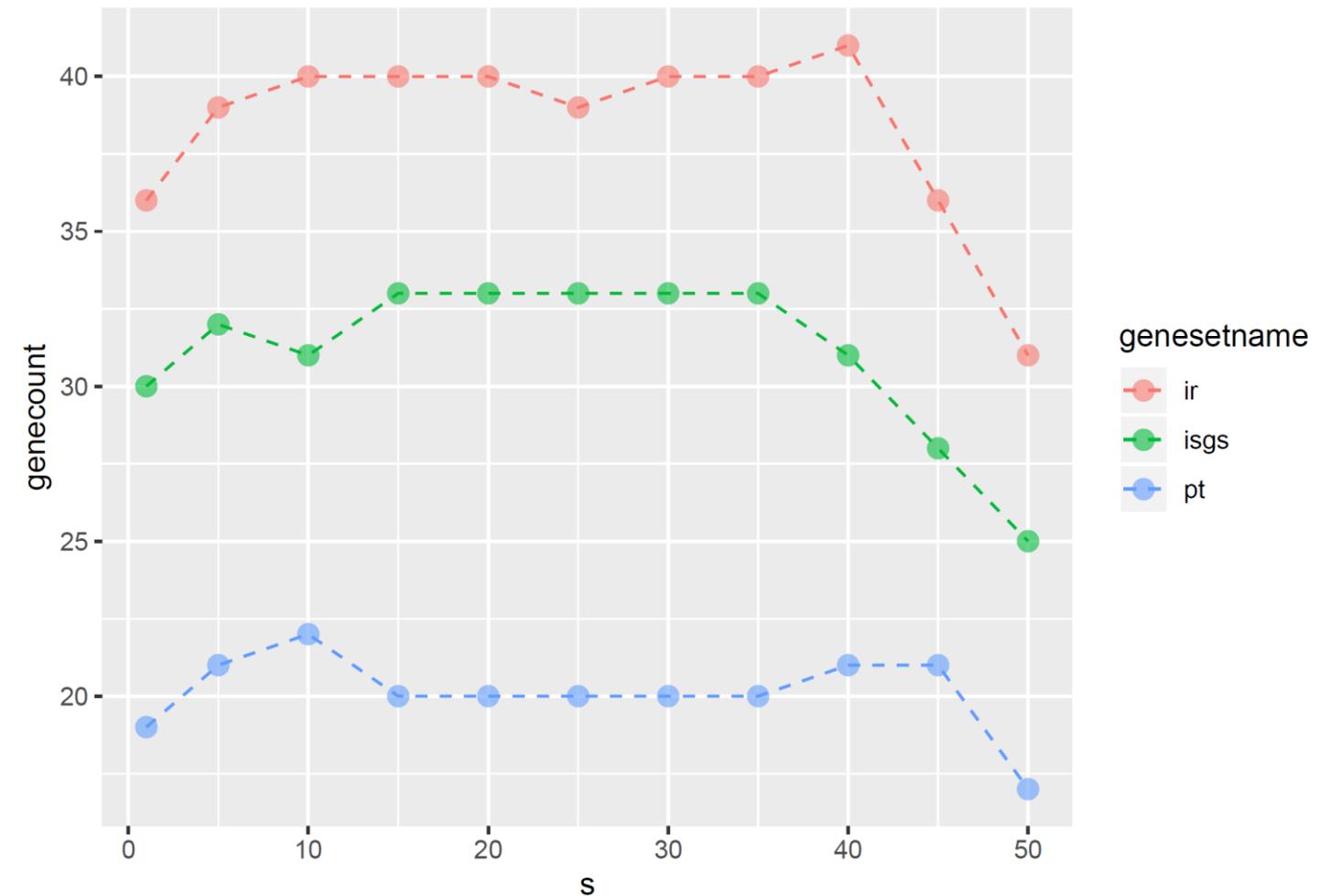
# s-Components in Host Response Data

- **Compute s-components for a range of s values**
  - $s = 1$  (“graph case”): 1 giant component, 4 single edge
  - $s = 50$ : 1 giant component, 5 small single-pathogen components, many single edge components
  - **Conclusion:** Similarities within pathogen stronger than between pathogens
  
- **One high width ( $s=100$ ) component:**
  - Late time point SARS and MERS, same tissue type
  - All MERS mutants at final time point
  - Sole SARS WT and one SARS mutant for  $> 36h$
  - **Conclusion:** SARS and MERS eventually perturb similar genes, with some exceptions

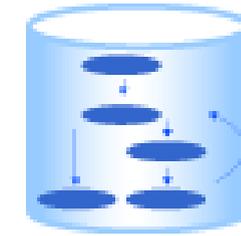


# s-Closeness in Host Response Data

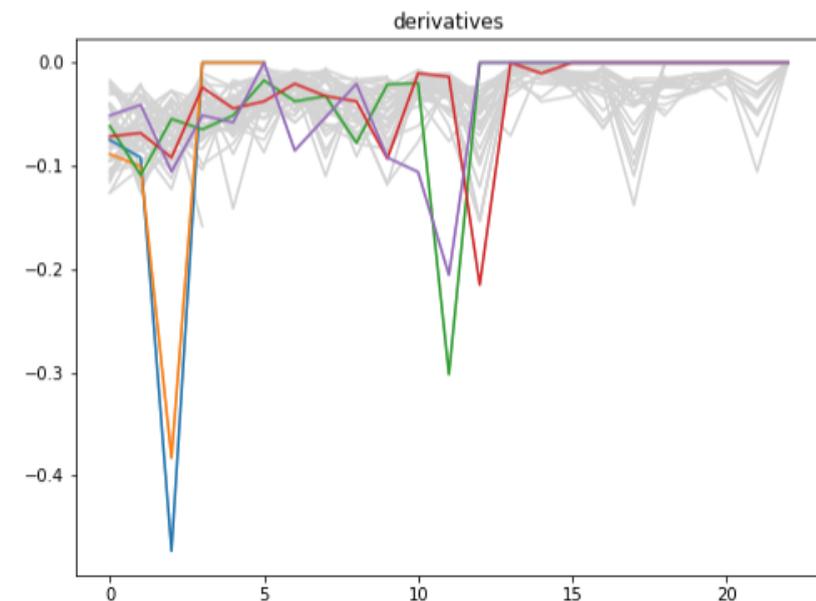
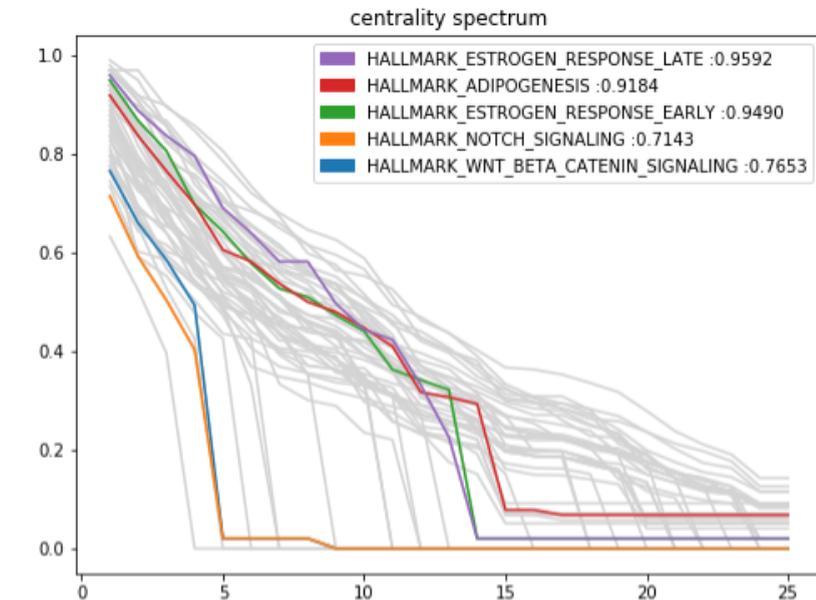
- Compute s-harmonic closeness to find important genes
  - $s = 1$  is like graph case
  - $s > 1$  shows added value of hypergraph model
- Compare top 50 genes at each s value to list of known immune response and pathogen targeted genes
  - Generally  $s > 1$  sees more of the known important genes than  $s=1$



# Transcription Factors from



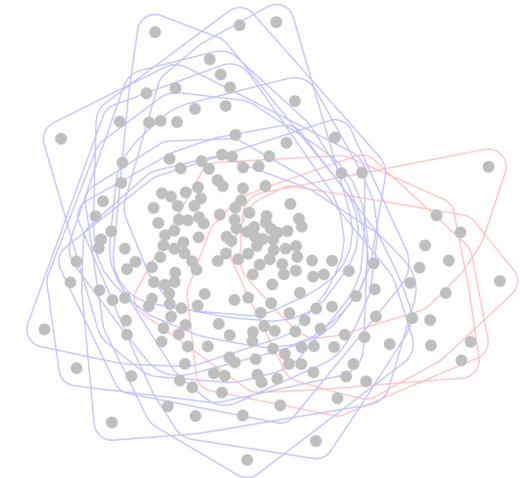
- **Transcription factor binding sites as groups of genes**
  - Gene sets representing potential targets of regulation by transcription factors or microRNAs
  - Each set (hyperedge) consists of genes (nodes) sharing short sequence in non-protein coding regions
- Compute s-centrality for sequence of s values
- Some edges are very central for small s but become less central as s increases
  - Consider slope of s-centrality curve to find outliers
- **ESTROGEN RESPONSE LATE** starts very central (0.9592) but quickly falls
  - **Conclusion:** Estrogen Response Late includes a small group of ubiquitous signaling components. The rest of the pathway is quite specific



# Three hypergraph data science vignettes

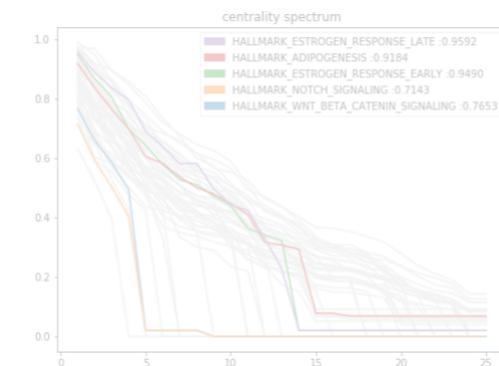
## 1. Pathogen host response data:

- s-Components to discover interconnectedness within and across pathogens
- s-Centrality to discover important genes



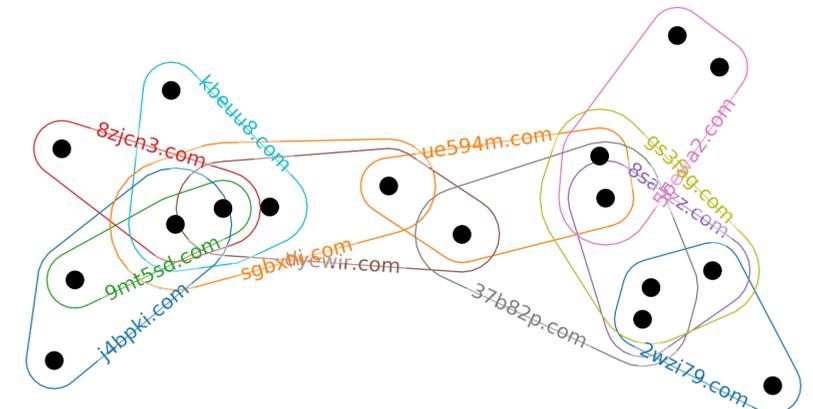
## 2. RNA Transcription factor data:

- s-Centrality to find behavior changes



## 3. Computer network defense:

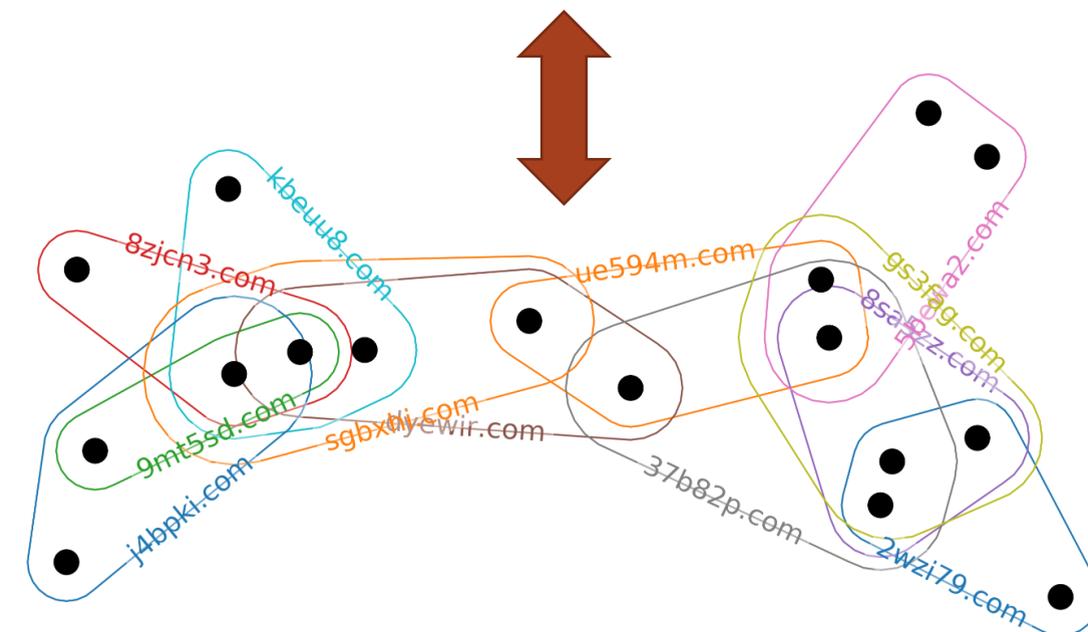
- Modeling DNS: domain  $\leftrightarrow$  IP relationships
- Discovering motifs



# DNS Use Case

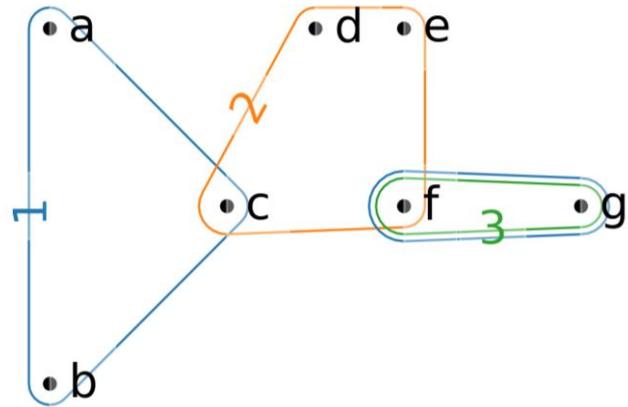
- **Hypergraph: IP X Domain**
  - Nodes = IP addresses
  - Hyperedges = domains
- **When DNS is not one-to-one:**
  - Domain aliases
  - Hosting services to multiple web sites
  - Site management across IPs
  - Random IP assignment
- **ActiveDNS: GA Tech**  
<https://activednsproject.org/>
- **Analytical Questions:**
  - *General Exploration:* Abnormal IPs and domains
  - *Targeted Exploration:* Neighborhoods of known bad IPs or domains

	2wzi79.com	37b82p.com	5bewa2.com	8sa5zz.com	8zjen3.com	9mt5sd.com	dyewir.com	gs3fag.com	j4bpki.com	kbeuu8.com	sgbxhi.com	ue594m.com
103.86.122.130						X			X	X		
103.86.122.148	X	X	X				X				X	
103.86.122.149			X				X				X	
103.86.122.152	X	X		X								
103.86.122.154					X	X	X			X	X	
103.86.122.160							X				X	X
103.86.122.169			X									
103.86.122.173	X											
103.86.122.181	X			X			X					
103.86.122.192					X							
103.86.122.195						X		X				
103.86.122.220			X									
103.86.122.222		X					X					X
103.86.122.223					X	X		X	X	X		
103.86.122.225	X	X		X			X					
103.86.122.238								X				
103.86.122.242									X			

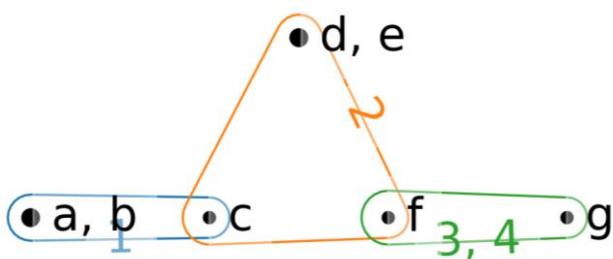


# One Day of Data

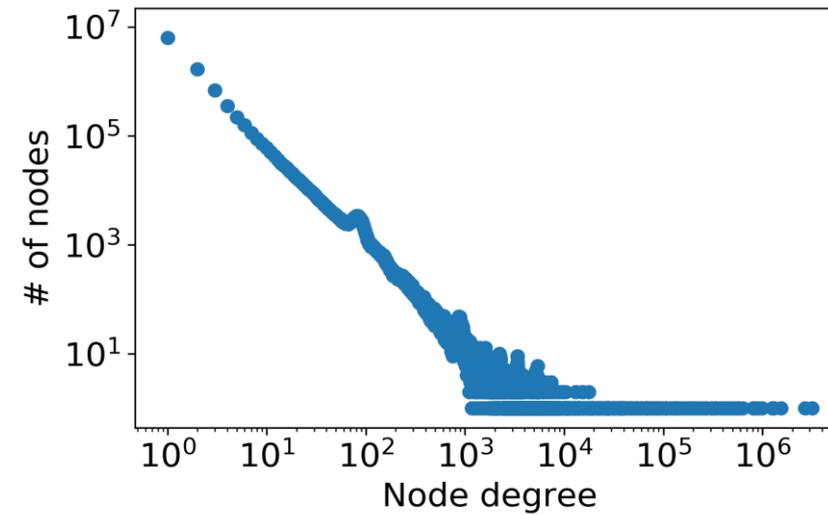
	Initial	Collapsed	Non-Singleton Components
$ V $	10.6M	10.3M	557K
$ E $	131.2M	11.0M	1.2M
Aspect ratio	0.081	0.941	0.460
# Cells	157.4M	25.7M	15.9M
Density	1.14 E-7	2.26 E-7	2.35 E-5



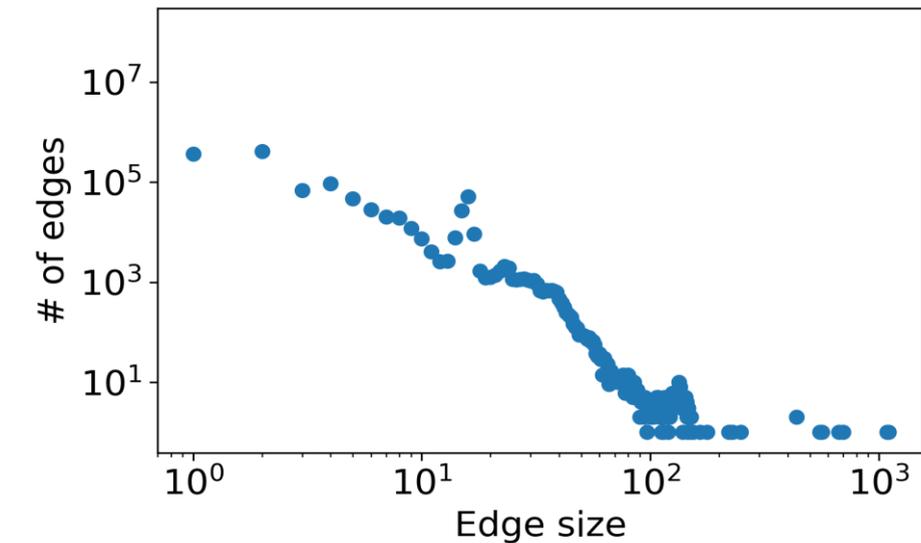
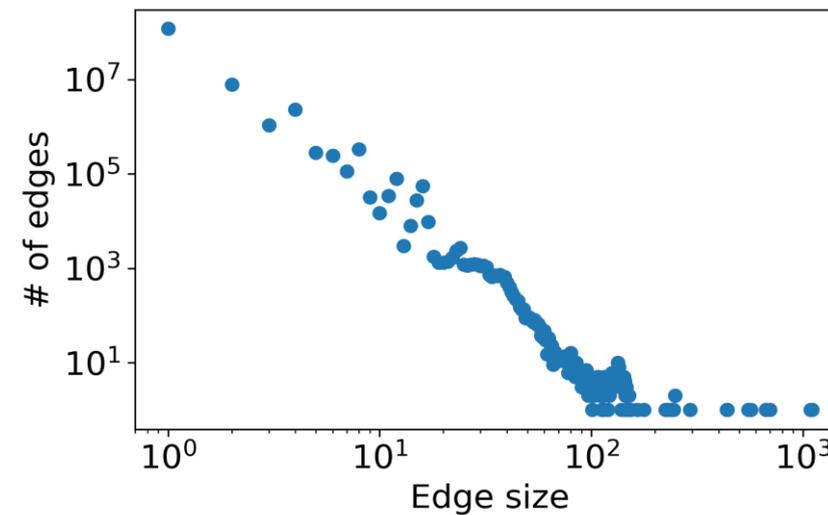
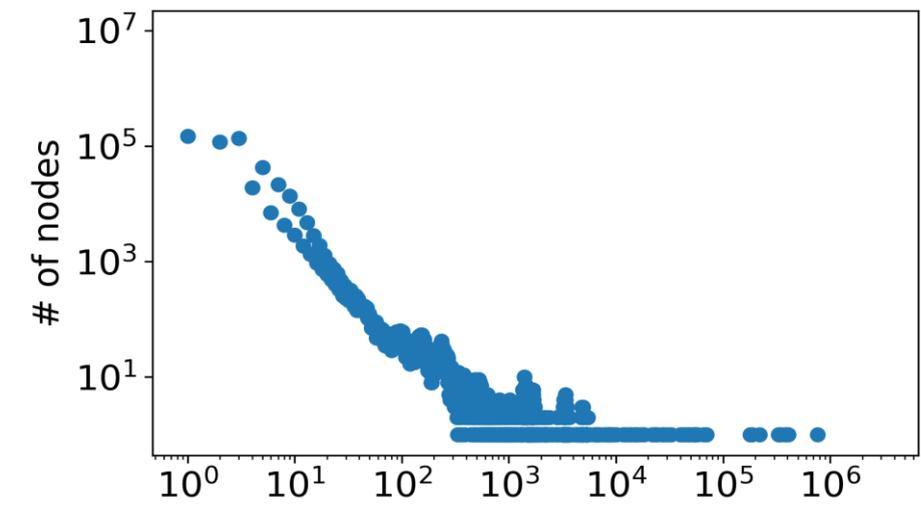
“Collapse”



Original



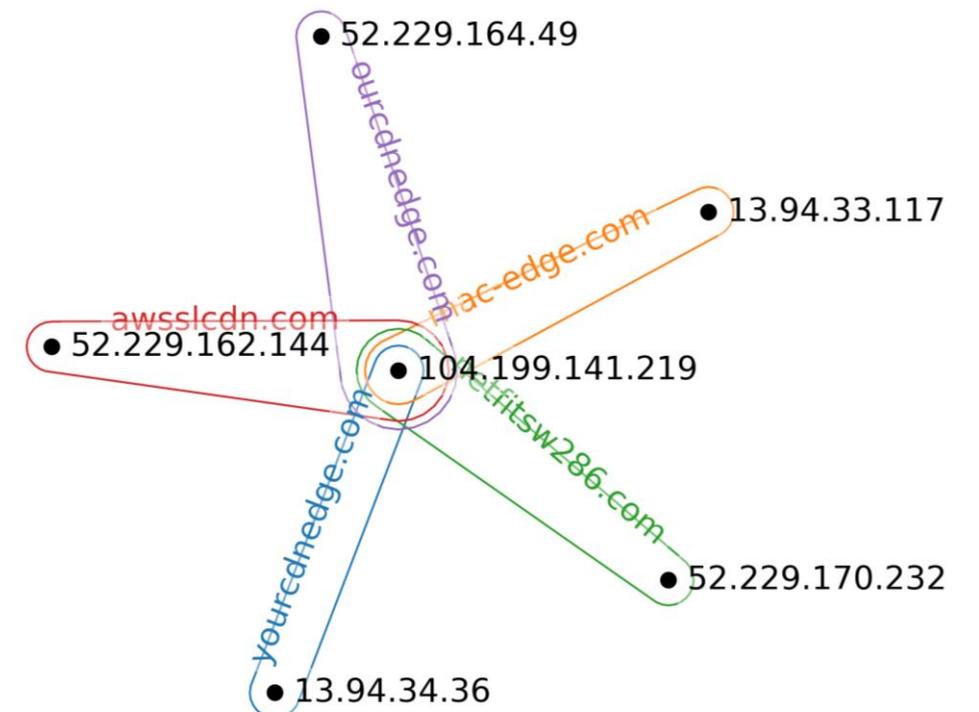
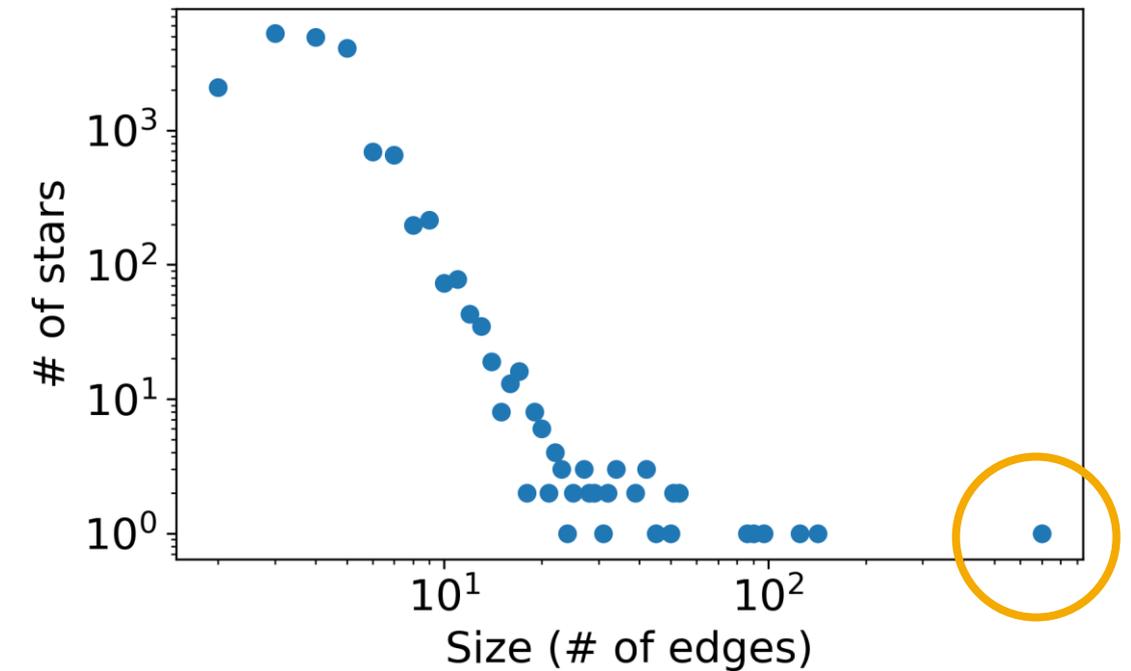
Fully Collapsed



# Star Motifs

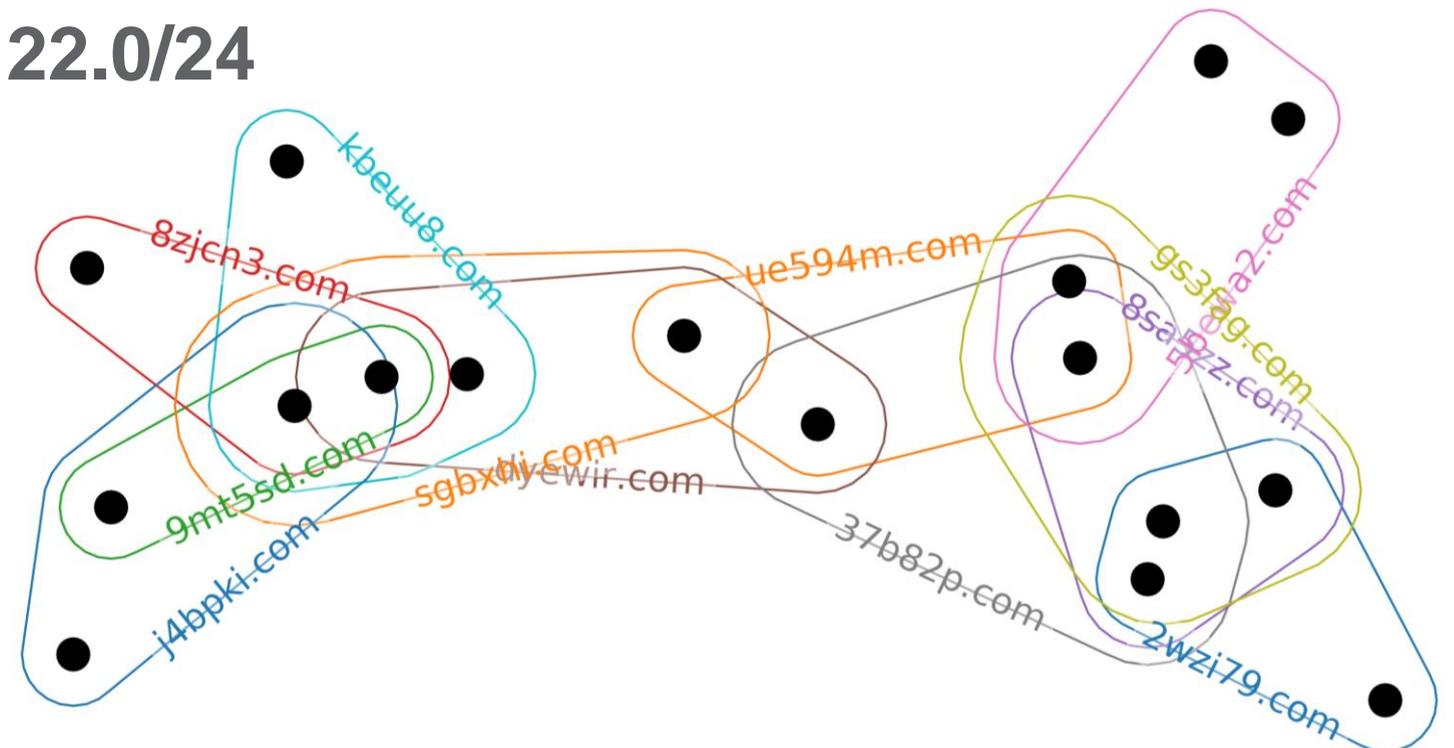
- Searched all 1-components for stars and computed their sizes (# edges)
- Largest star is outlier with 642 leaves, consistent with DNS sinkhole behavior
  - Central node 17.17.17.17 with start of authority (SOA) record proclaiming “sinkhole root@sinkhole”
  - Leaf nodes come from 640 distinct /16 (first two octets) IP ranges
- Smaller stars more consistent with content delivery networks (CDNs)
  - All IPs and domains within the same, or a relatively small set of, ranges and organizations
  - Example: Central IP address registered to Google Cloud, leaves registered to Microsoft Corporation. All five domains are registered through the hosting site GoDaddy.com.

All observations about DNS records and IP or domain registration were found using publicly available services like WHOIS and BGP routing.



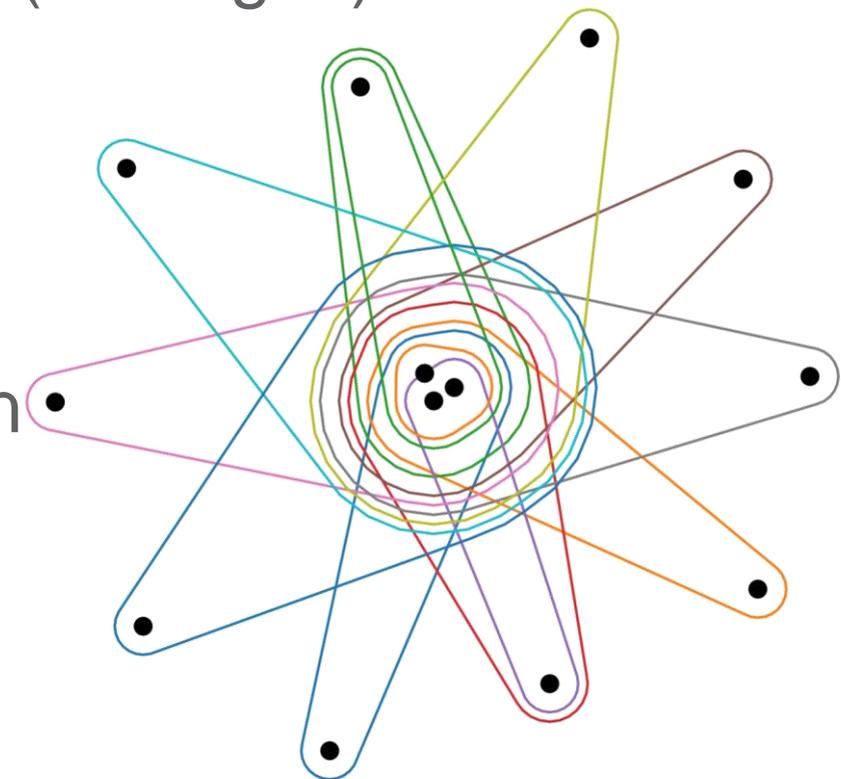
## Large diameter components

- Computed diameter of all 2- and 3-components
- Max diameter (6) 2-Component: Consistent with *fast flux* behavior
  - Relationship between IP and domain is very short-lived
  - Used by botnets to hide malicious content delivery sites and make malware networks more difficult to discover
- All domains with IPs in 103.86.122.0/24
  - In late NOV 2018 the IPs were 103.86.123.0/24 with time to live (TTL) of 120 seconds
  - Now many of the domains have no associated IP addresses



# Targeted exploration

- FireEye Threat Research Blog
  - “On the Hunt for FIN7: Pursuing an Enigmatic and Evasive Global Criminal Operation”
  - Contains list of ‘blacklisted’ IP addresses and DNS names
- Set of ten domains that follow the blacklisted pattern all contained within the same small 2-component (16 edges) and 3-component (13 edges)
  - No common intersection among all domains
  - Two central IPs with each domain containing at least one of these two IPs
  - All domains registered by the same organization
- Targeted analysis could be used to discover how known TTP signatures manifest





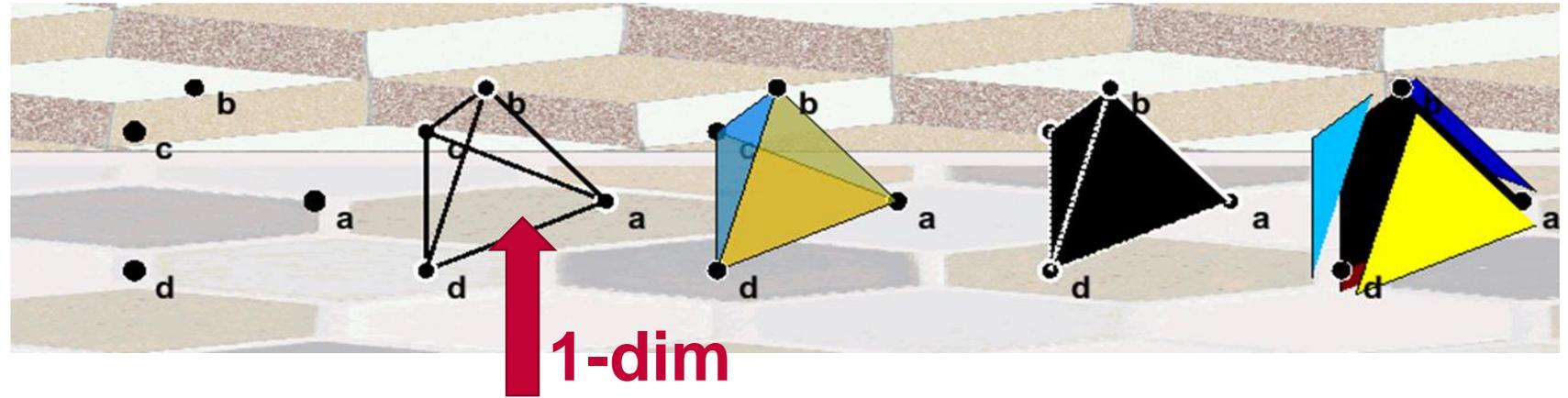
The Overview

The Math

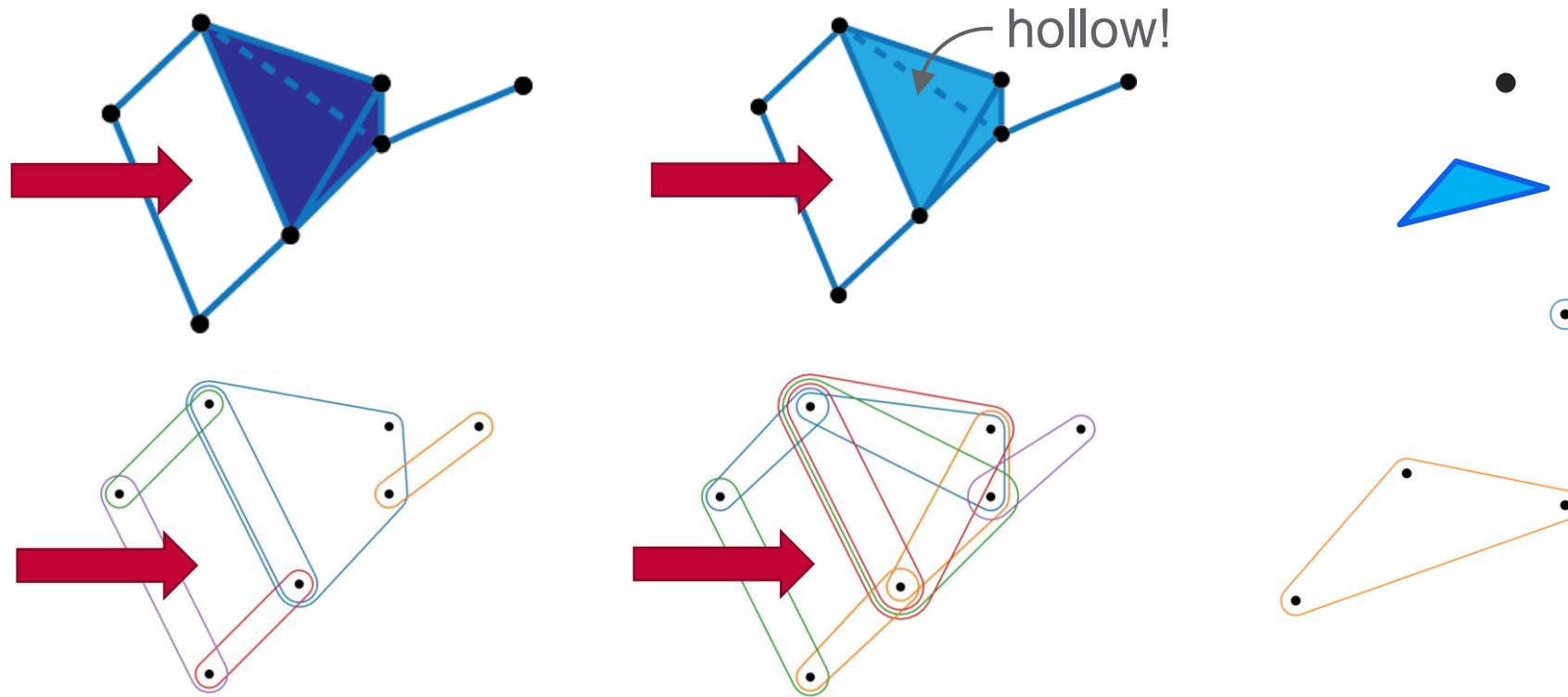
The Applications

# Future Directions

# Hypergraphs as Topological Objects

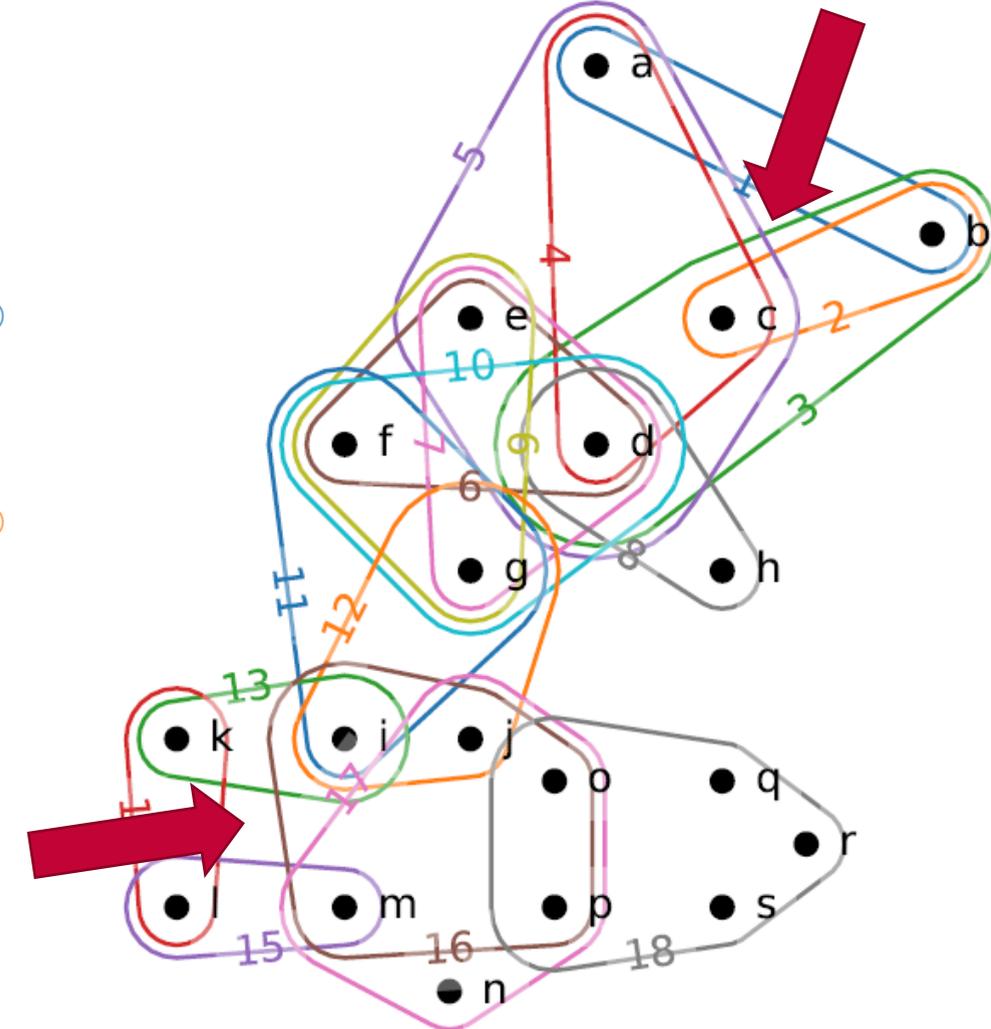


- Hypergraphs as multidimensional objects have topological properties

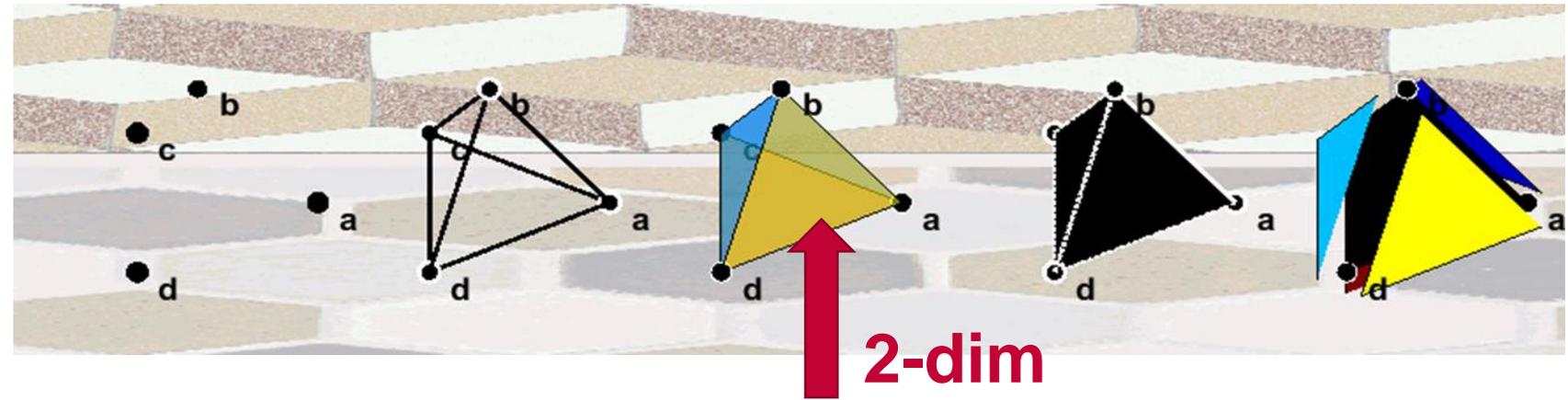


- **Homology to identify multidimensional holes**

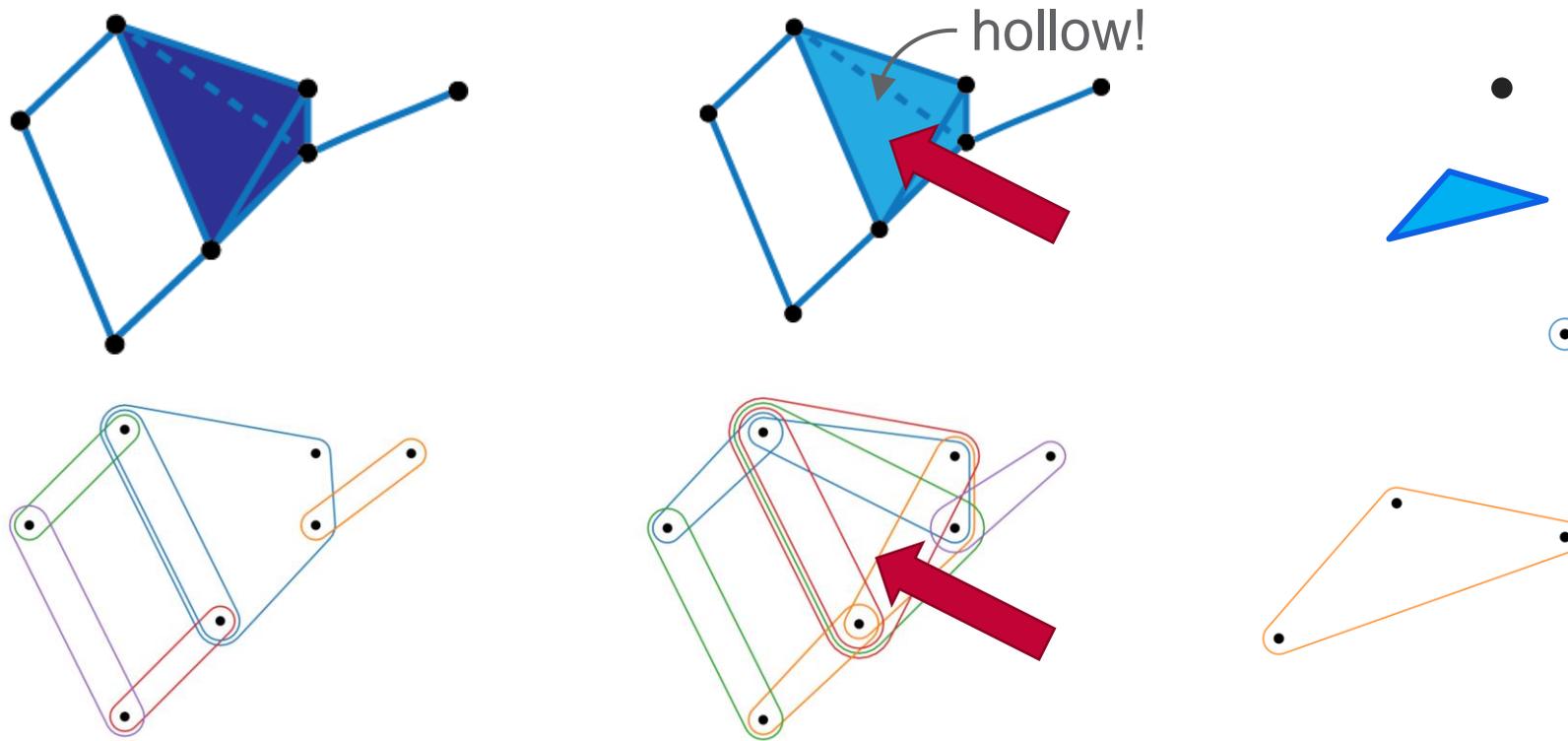
- As hypotheses for missing data
- Need for bridging metadata



# Hypergraphs as Topological Objects

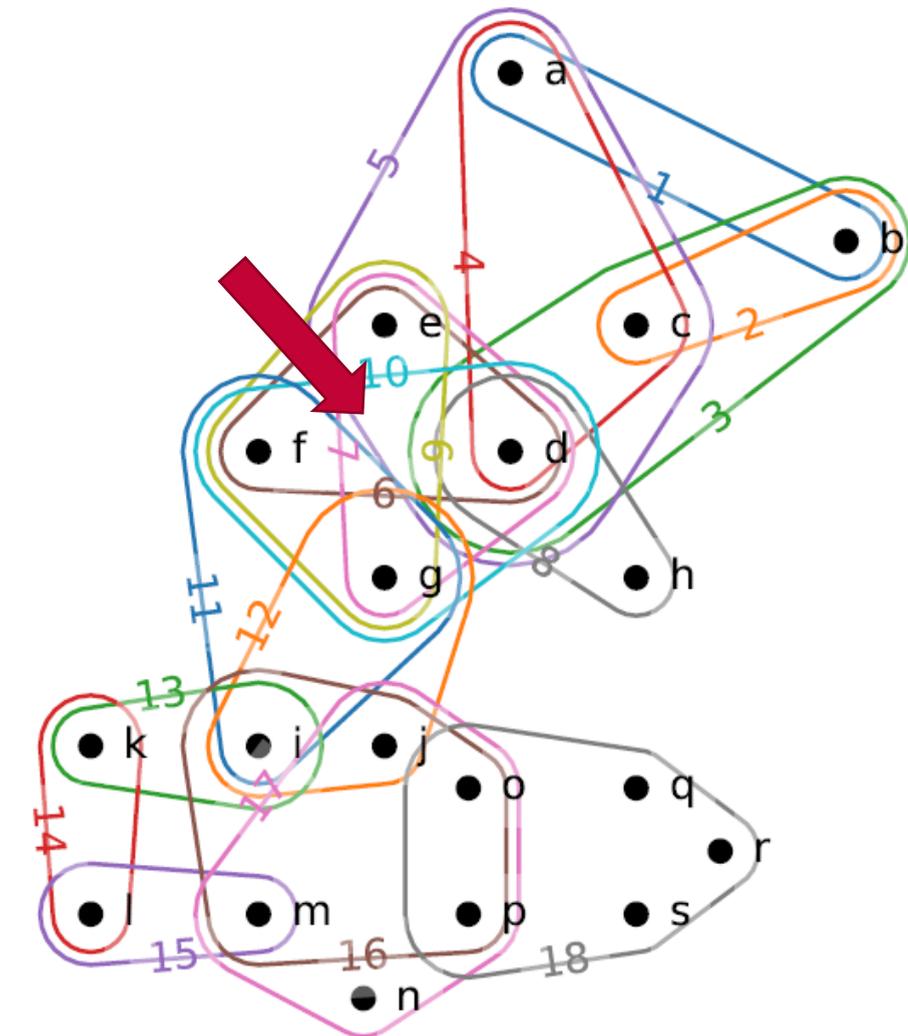


- Hypergraphs as multidimensional objects have topological properties

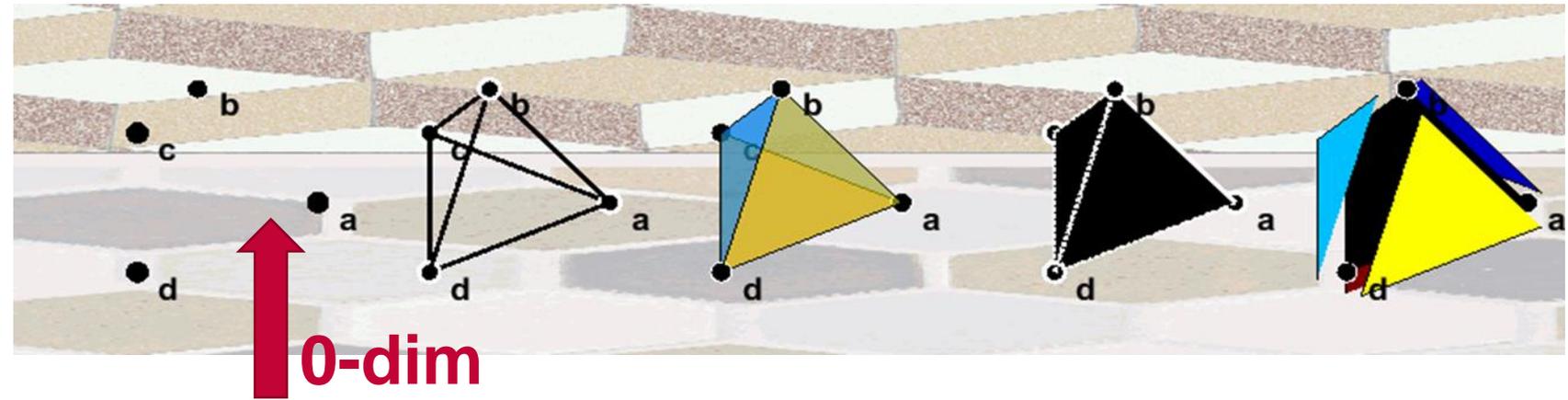


- **Homology to identify multidimensional holes**

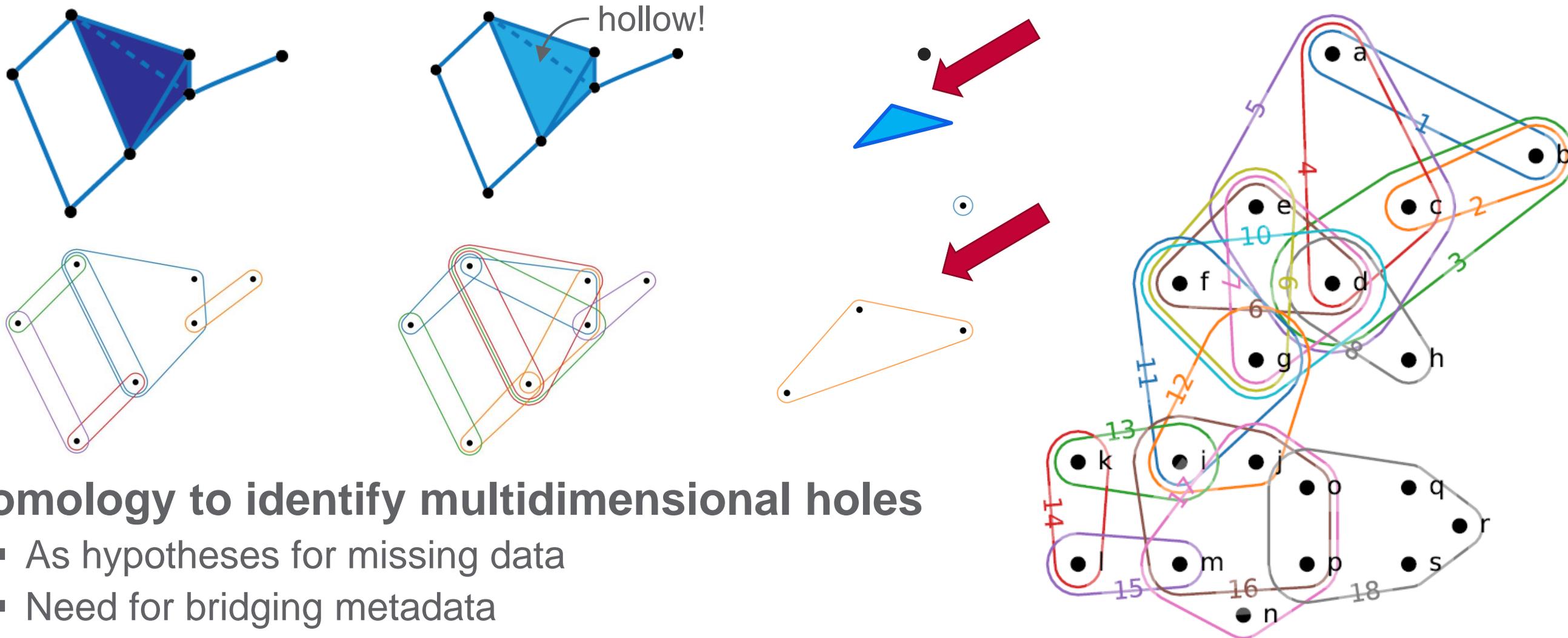
- As hypotheses for missing data
- Need for bridging metadata



# Hypergraphs as Topological Objects



- Hypergraphs as multidimensional objects have topological properties

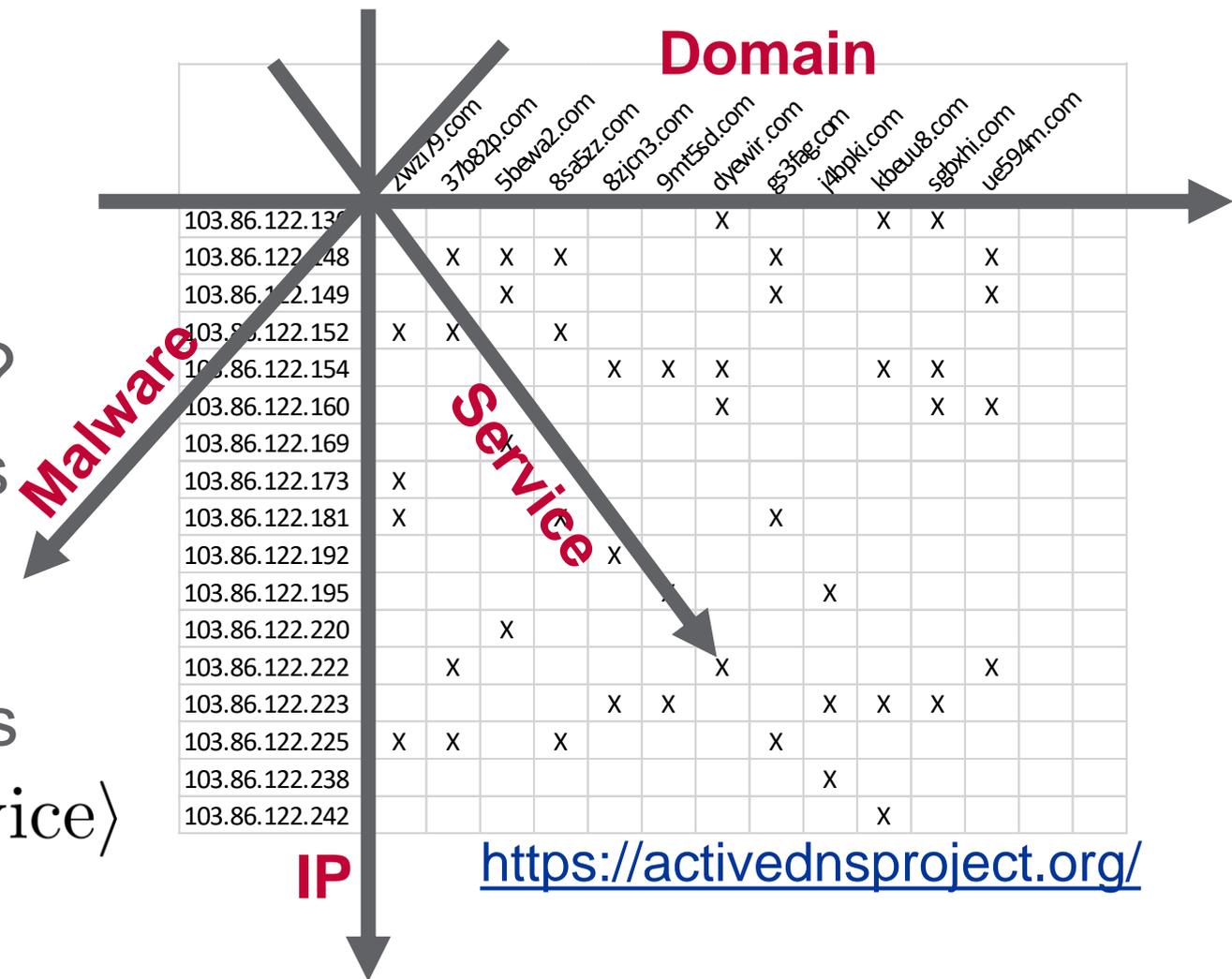


- **Homology to identify multidimensional holes**

- As hypotheses for missing data
- Need for bridging metadata

# Towards Multi-Attribute Hypergraph Systems

- Current focus on Dom x IP, but:
  - What about other identifiers?
- What is the right generalization of hypergraphs for multiple dimensions?
  - **Pivoting:** amongst pairs of dimensions  
 $\text{Dom} \times \text{IP} \rightarrow \text{IP} \times \text{Service} \rightarrow \dots$
  - **Properties:** “unused” dims to attributes  
 $\text{Dom} \times \text{IP}$  properties  $\langle \text{malware}, \text{service} \rangle$
  - **Nesting:** recursive structure  
 $\text{Dom} = \{ \text{IP} = \{ \text{Malware} = \{ \text{Service} \} \} \}$

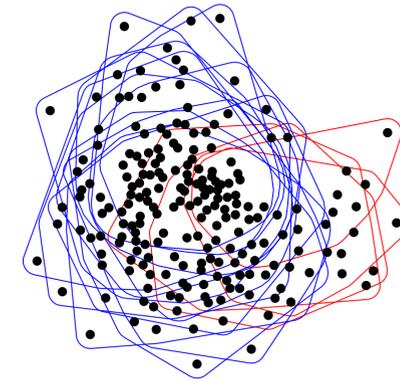


	Domain											
	zwz179.com	37b82p.com	5bewa2.com	8sa5z.com	8zjcn3.com	9mt5sd.com	dyewir.com	gs5fag.com	j4bpki.com	kbeu8.com	sgbxhi.com	ue594m.com
103.86.122.130						X				X	X	
103.86.122.148	X	X	X				X					X
103.86.122.149		X					X					X
103.86.122.152	X	X		X								
103.86.122.154					X	X	X			X	X	
103.86.122.160							X				X	X
103.86.122.169												
103.86.122.173	X											
103.86.122.181	X							X				
103.86.122.192					X							
103.86.122.195									X			
103.86.122.220			X									
103.86.122.222		X					X					X
103.86.122.223					X	X			X	X	X	
103.86.122.225	X	X		X				X				
103.86.122.238									X			
103.86.122.242										X		

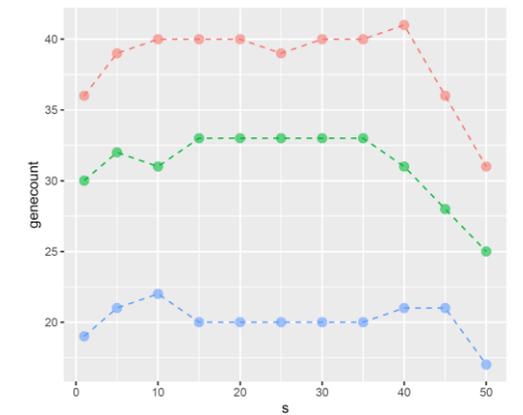
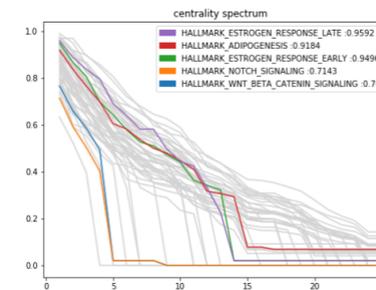
<https://activednsproject.org/>

# Hypergraphs for... your complex data

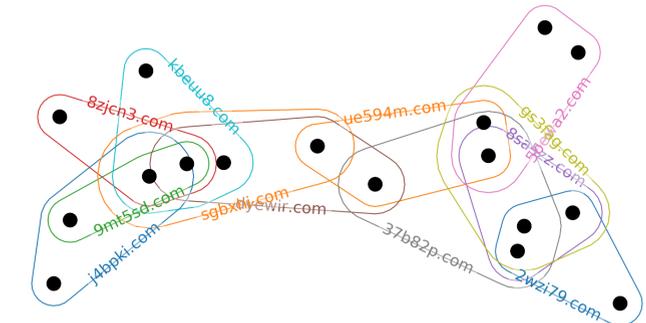
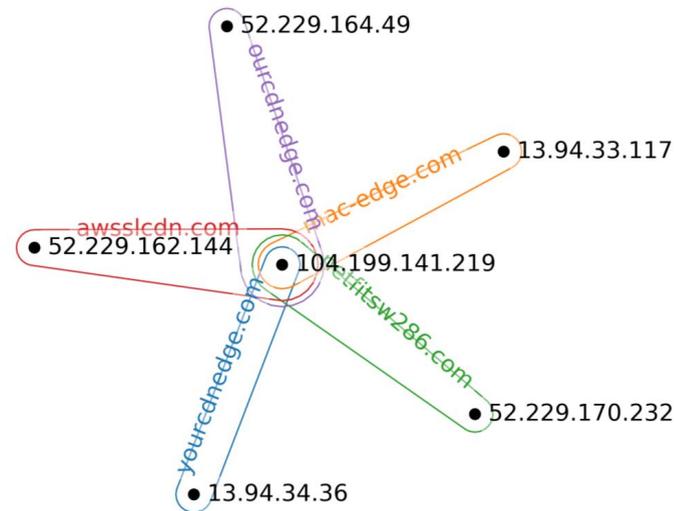
1. Hypergraphs for biology



2. Hypergraphs for cyber



3. What's next?





**Thank you**

