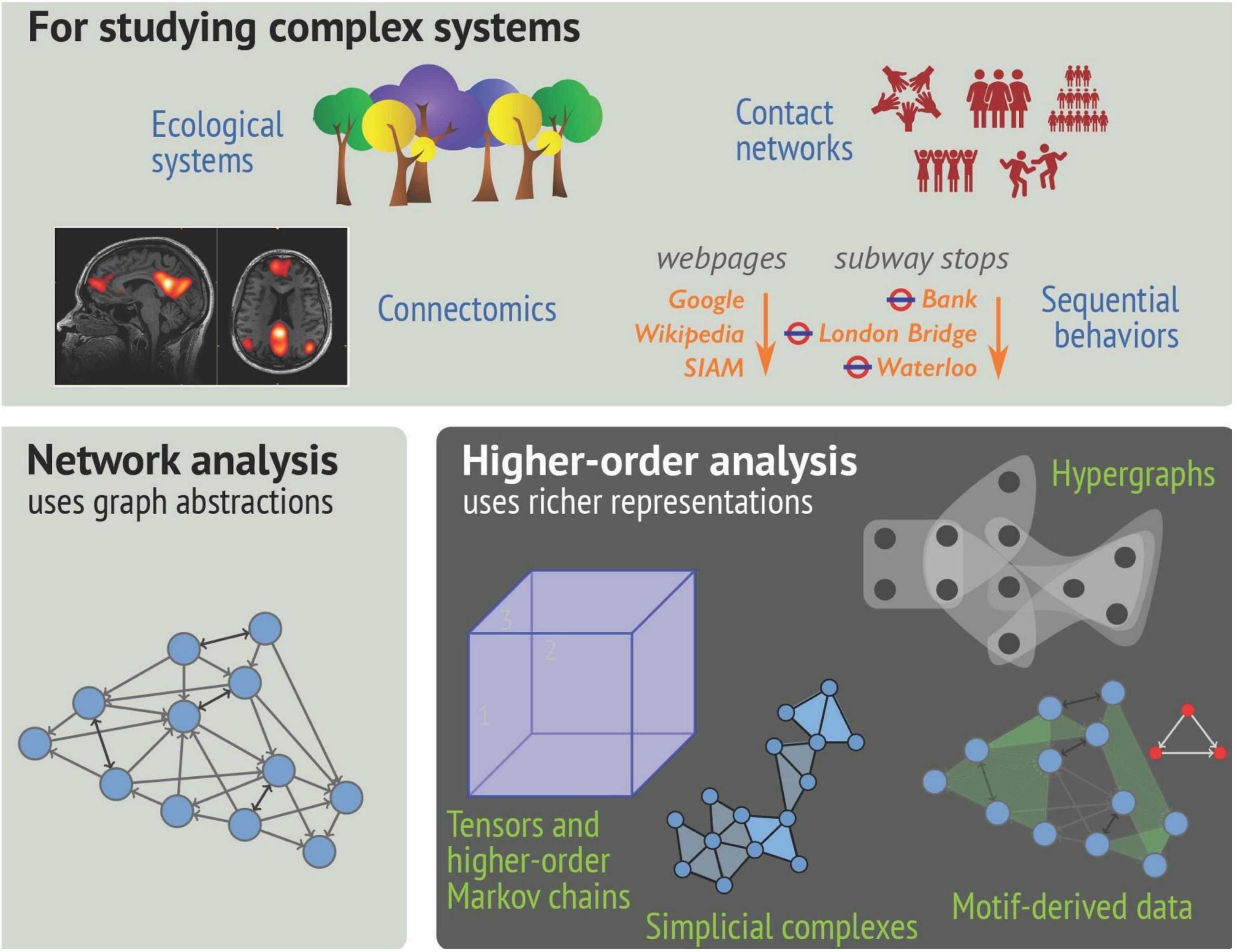
Higher-order networks

An introduction

M. Schaub, RWTH Aachen

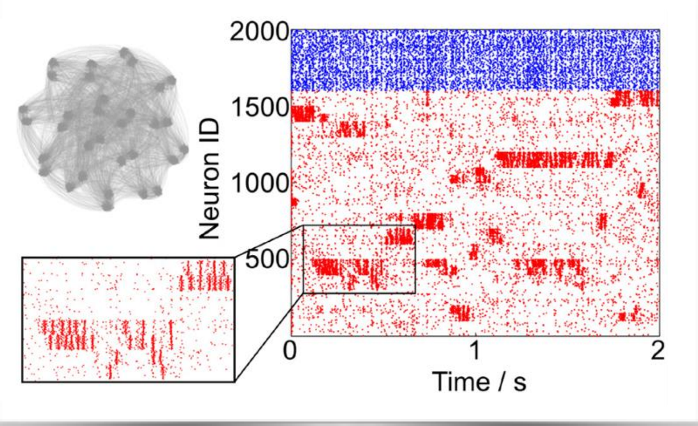
Higher-order networks



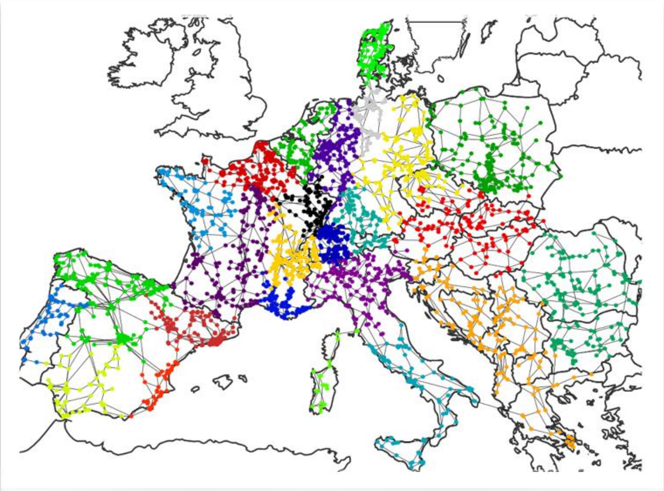
[Benson et al. SIAM News 2021]

Networks Everywhere!?

[Schaub et. al, PloS Comp Bio 2015]



[Schaub et. al, PLoS one 2012]



- Networks / Graphs have become prevalent abstractions across Engineering and Science
- Applications abound from biology to social systems to Engineering etc.
- Network models everywhere, but modeling paradigms potentially very different!

Model paradigm A – Networks to model relational data

Example 1

- Observed Data: relations between people in social network
- Task: Find communities

Example 2

- Observed Data: Predator-Prey relations
- Task: Analyse foodweb for central entities

“Generic Goal”

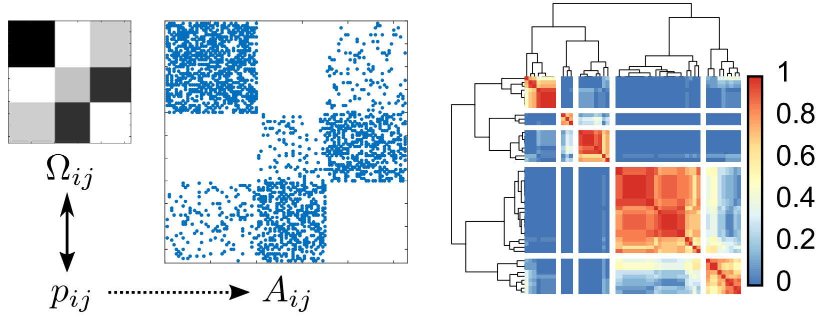
Analyse, model and understand connectivity patterns in complex systems

Model / data perspective A

data = network (edges/relations)

$$p(A) = \prod_{(ij)} \omega_{g(ij)}^{A_{ij}} (1 - \omega_{g(ij)})^{1-A_{ij}}$$

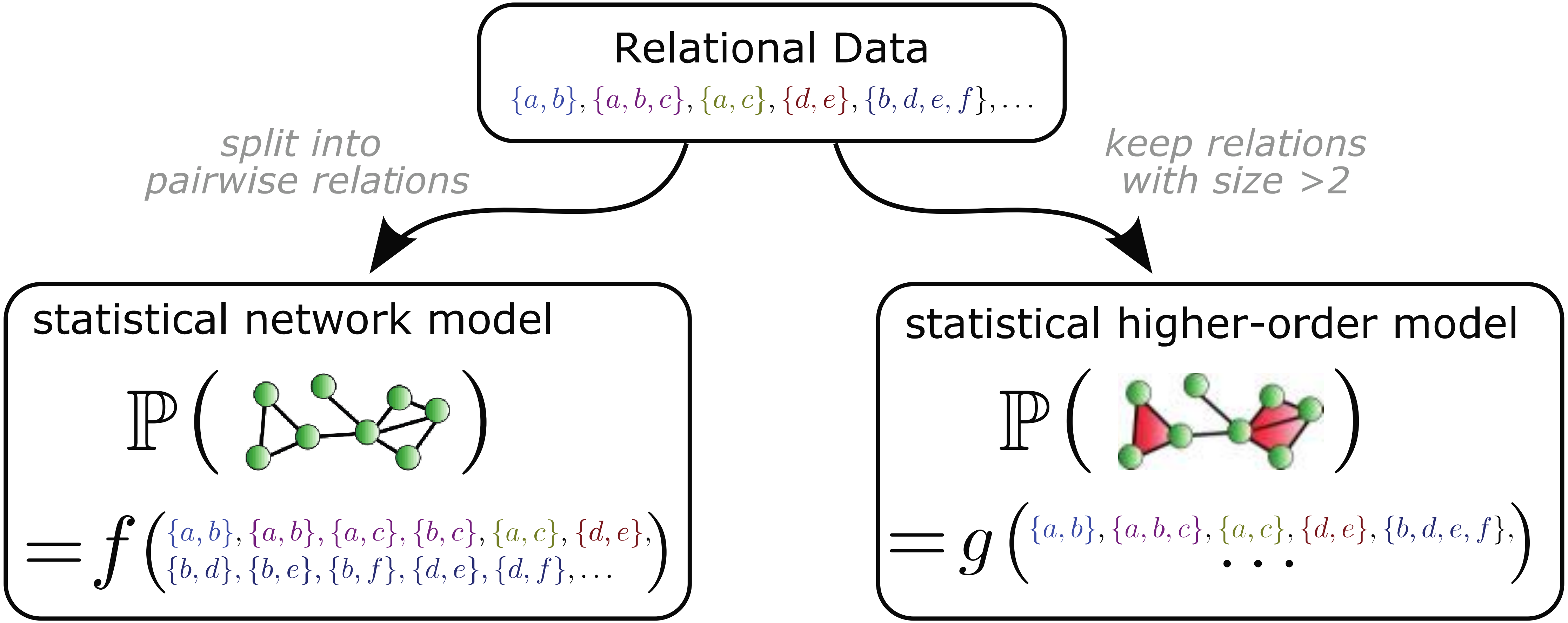
Example: clustering expression data / infer a generative network model



Mathematical tools

Statistical Inference,
Machine Learning

Model paradigm A – Higher-order networks to model relational data



Model paradigm B – Extract relations from (point-cloud) data to analyze systems

Example 1

- Observed Data: Point Cloud Data
- Task: Find manifold of data (dimensionality reduction)

Example 2

- Observed Data: Point Cloud Data,
- Task: Understand topological shape of point cloud, e.g., via persistent homology

“Generic Goal”

Leverage (geometrical) information to understand overall composition of observed data

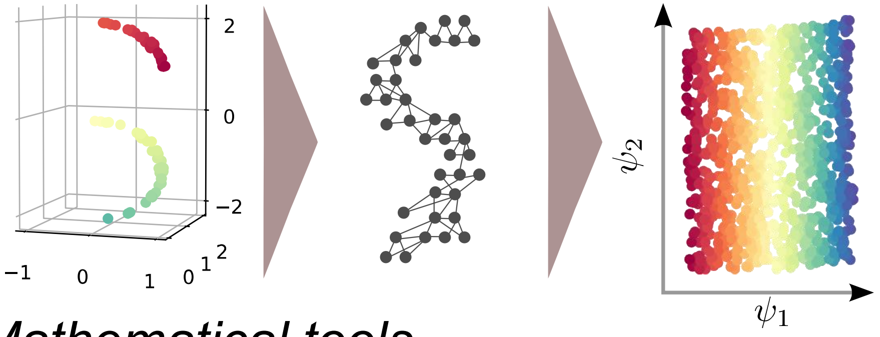
Model / data perspective

learn data manifold with graphs

$$\{X_i\}_{i=1}^{N} \rightarrow \mathcal{G} \rightarrow \mathcal{D}$$

raw data — graph (e.g., k-nn graph) — new data parametrization

Example: parametrizing single cell RNA-seq data via graphs



Mathematical tools

Manifold learning

Topological Data analysis

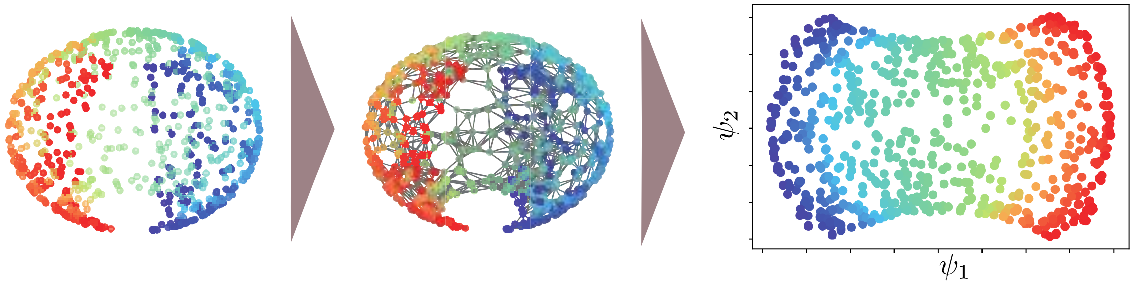
Model paradigm B – Extract higher-order relations from (point-cloud) data to analyze systems

A *Graph-based*

$\{X_i\}_{i=1}^N \rightarrow \mathcal{G} \rightarrow \mathcal{D}$

raw data — graph — data description

Example: Isomap (dim. reduction)

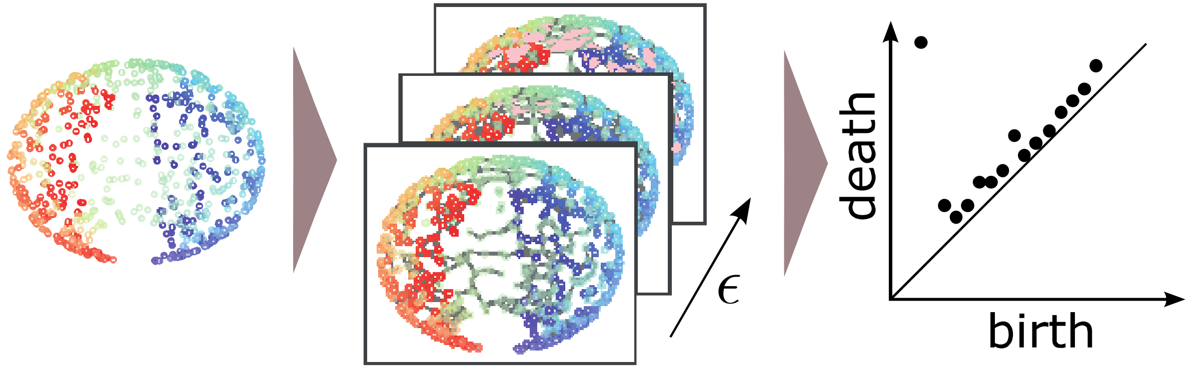


B *Higher-order*

$\{X_i\}_{i=1}^N \rightarrow \mathcal{X} \rightarrow \mathcal{D}$

raw data — (simplicial) complex — data description

Example: persistent homology



Model paradigm C – Understand data supported on top of network

Example 1

- Start: ODE model for opinion formation on graphs
- Task: understand long-term behavior of dynamics, pattern formation

Example 2

- Starting point: social network with node attributes
- Task: Predict unobserved attributes

“Generic Goal”

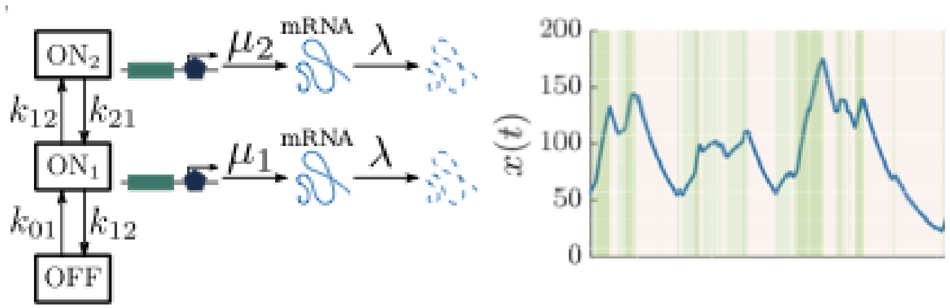
Leverage, model and analyze interplay between “structure” and data supported on ”structure” to understand observed system

Model / data perspective

data / dynamics on 'fixed' network

$$\dot{x}_i = f(x_i) + \sum_j A_{ij} g(x_i, x_j) + u_i$$

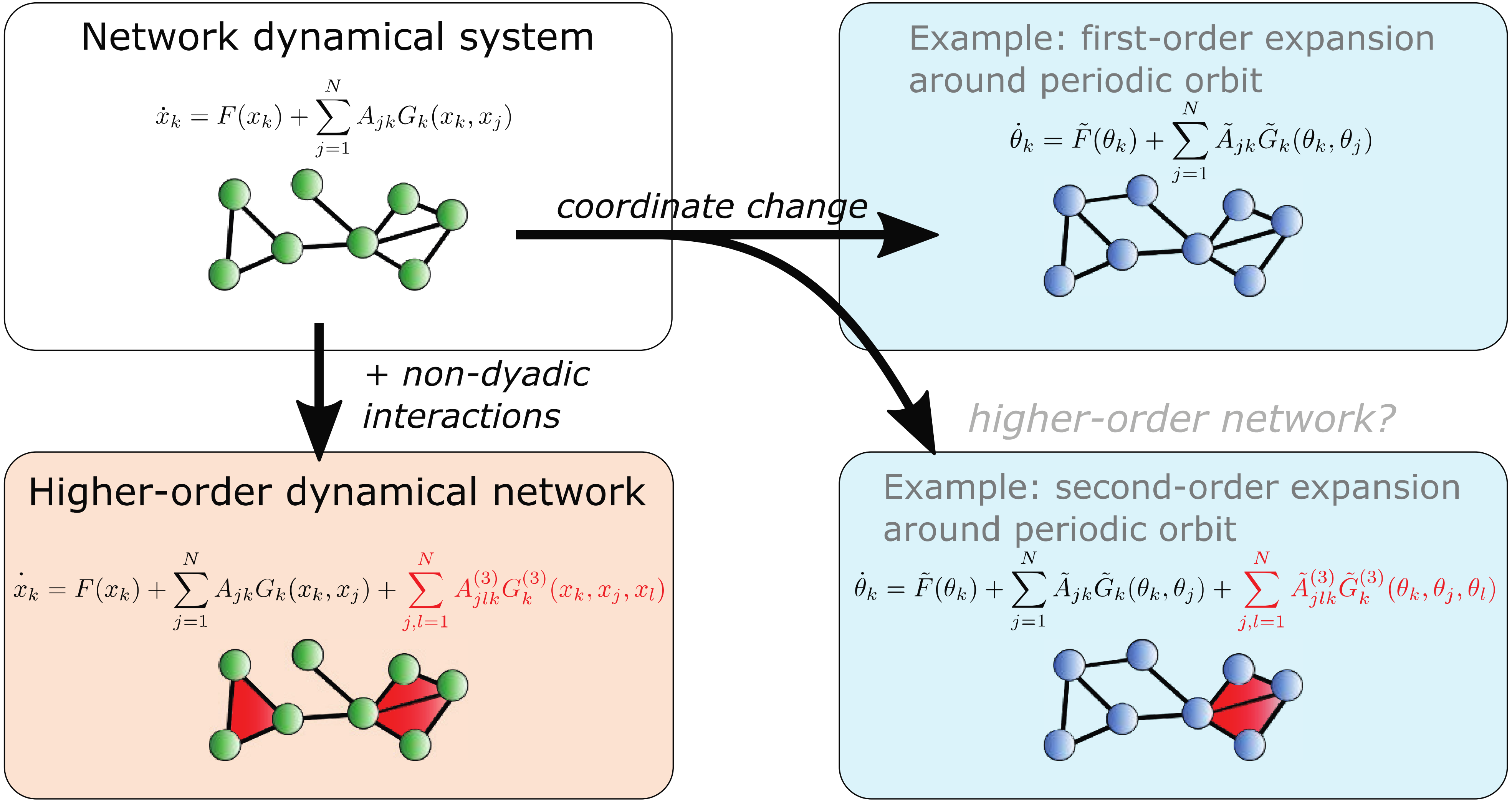
Example: ODE models, Master equations



Mathematical tools

Dynamical Systems & Control Theory
Graph Signal Processing

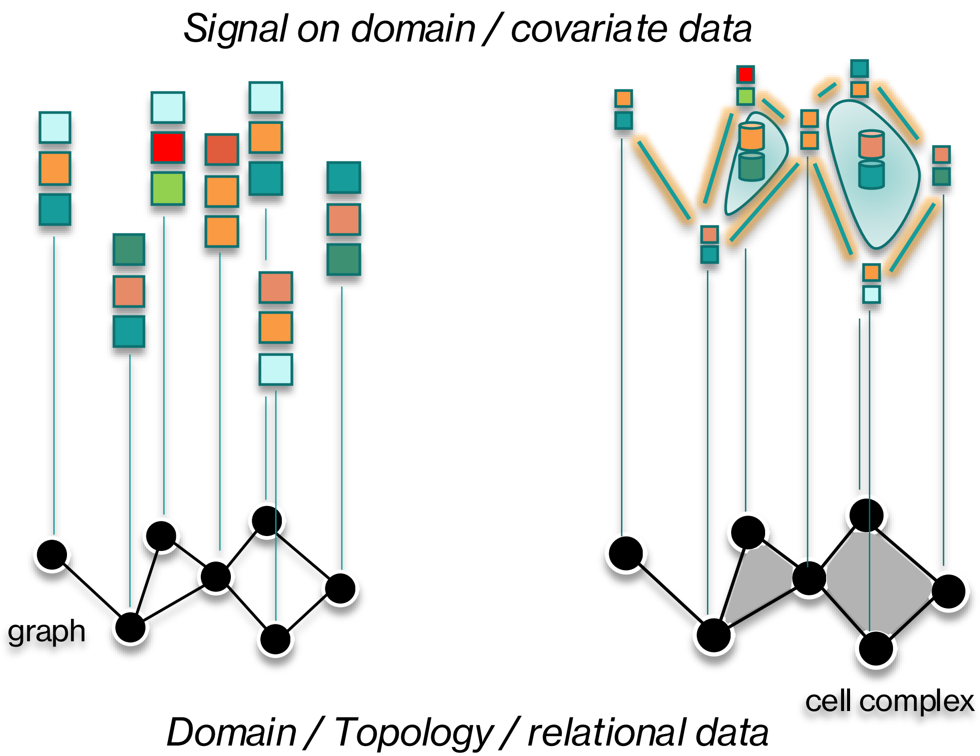
Model paradigm C – Understand data supported on top of higher-order network



Today’s Menu

Overview of higher-order networks from 3 perspectives

- Basic Concepts and misconceptions
- Models and methods for higher-order relational data
- Geometry/Topology with higher-order relations
- Data and Dynamics on top of higher-order networks

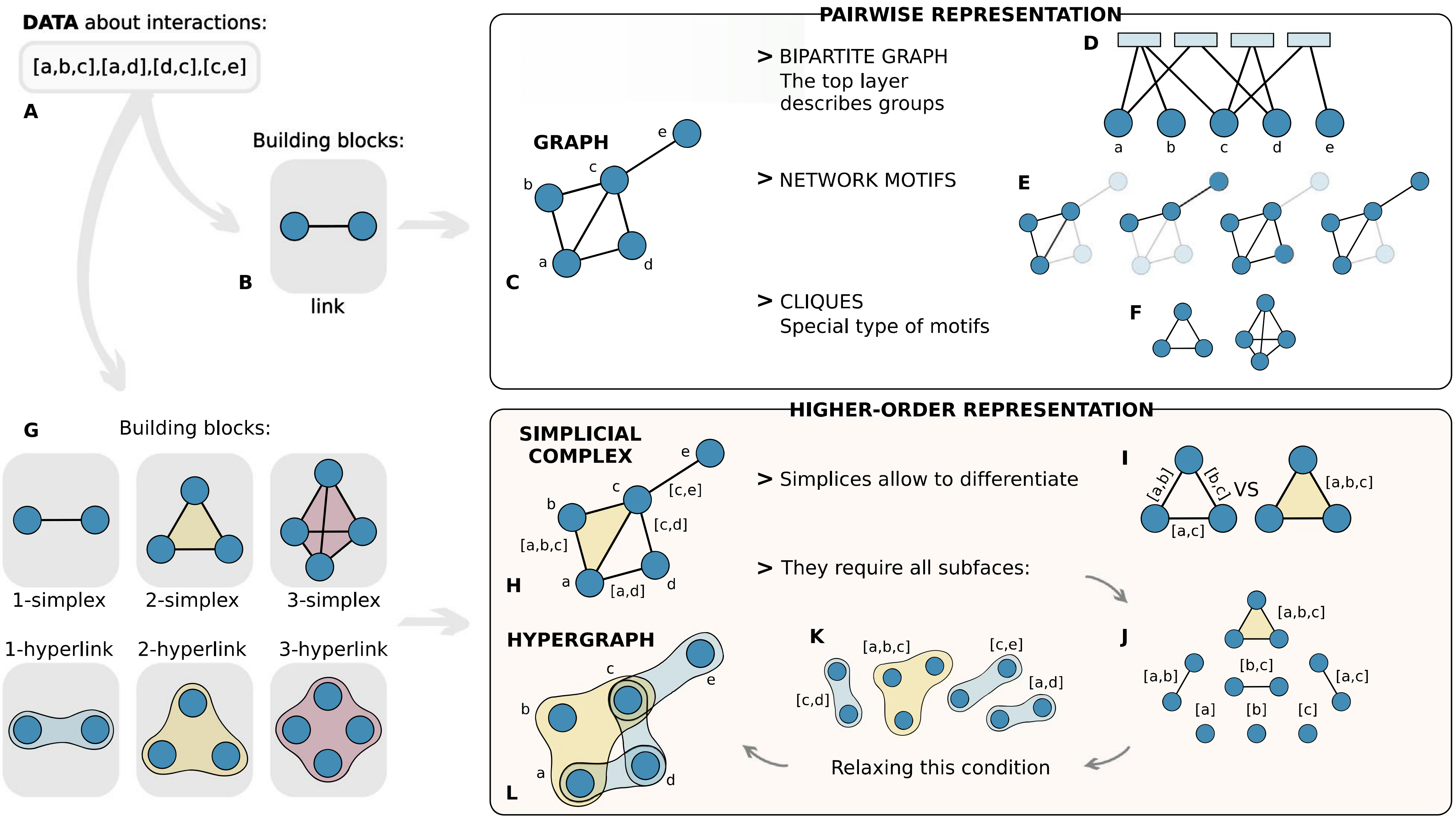


Some useful references

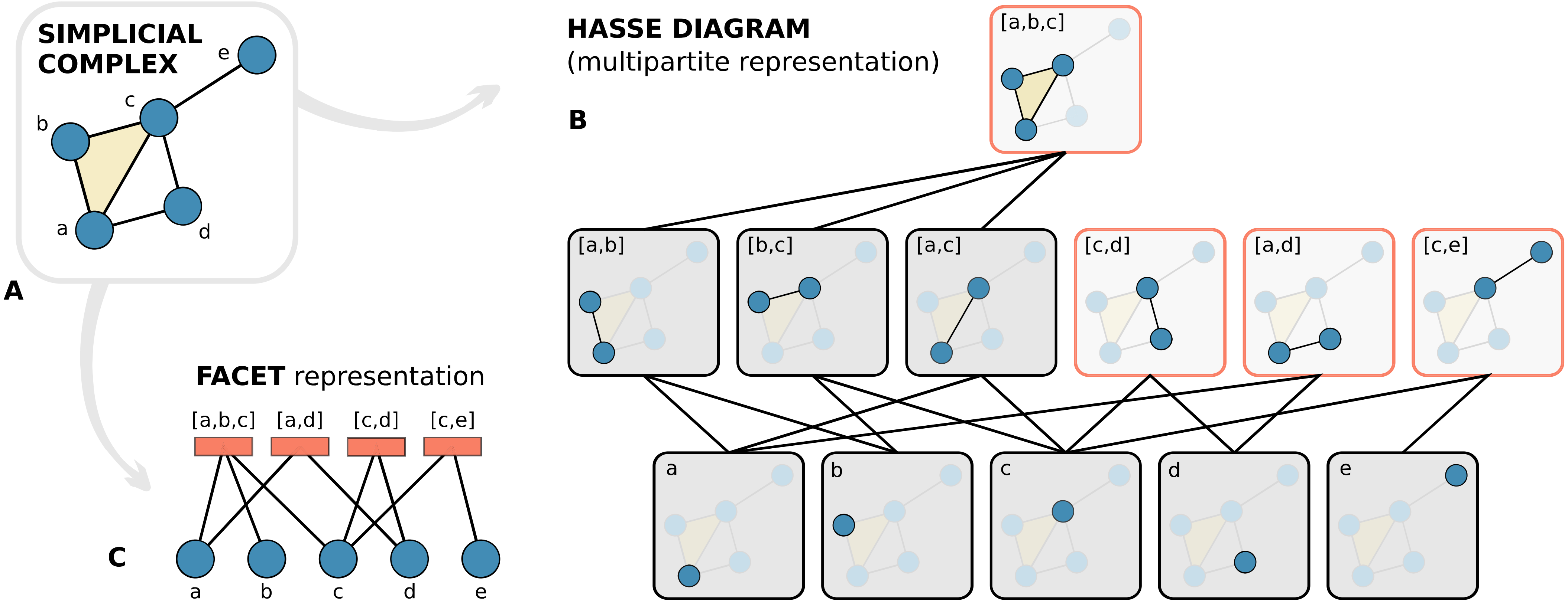
- Bick C, Gross E, Harrington HA, Schaub MT. What are higher-order networks?. SIAM review. 2023;65(3):686-731.
- Battiston F, Cencetti G, Iacopini I, Latora V, Lucas M, Patania A, Young JG, Petri G. Networks beyond pairwise interactions: Structure and dynamics. Physics reports. 2020 Aug 25;874:1-92.
- Torres L, Blevins AS, Bassett D, Eliassi-Rad T. The why, how, and when of representations for complex systems. SIAM Review. 2021;63(3):435-85.

Chapter 1: Basic concepts and misconceptions

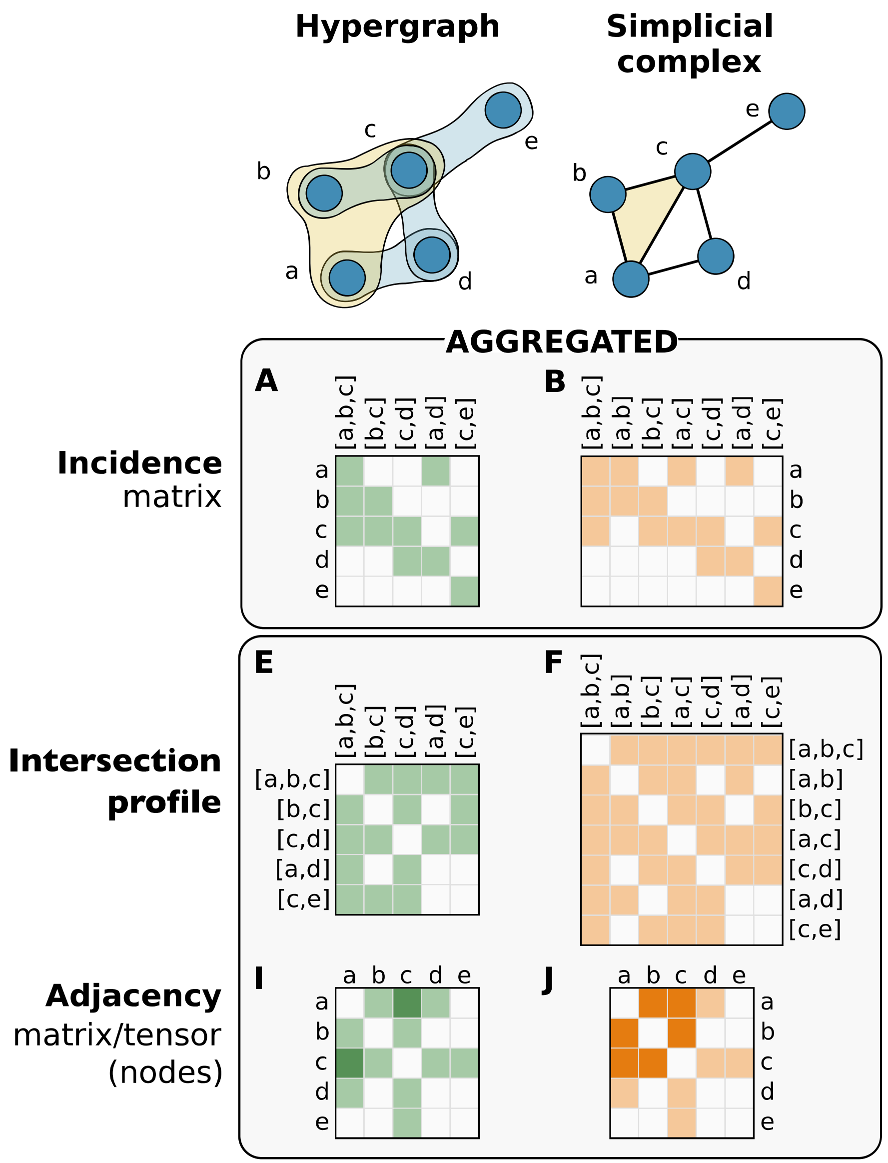
Representations of relational data



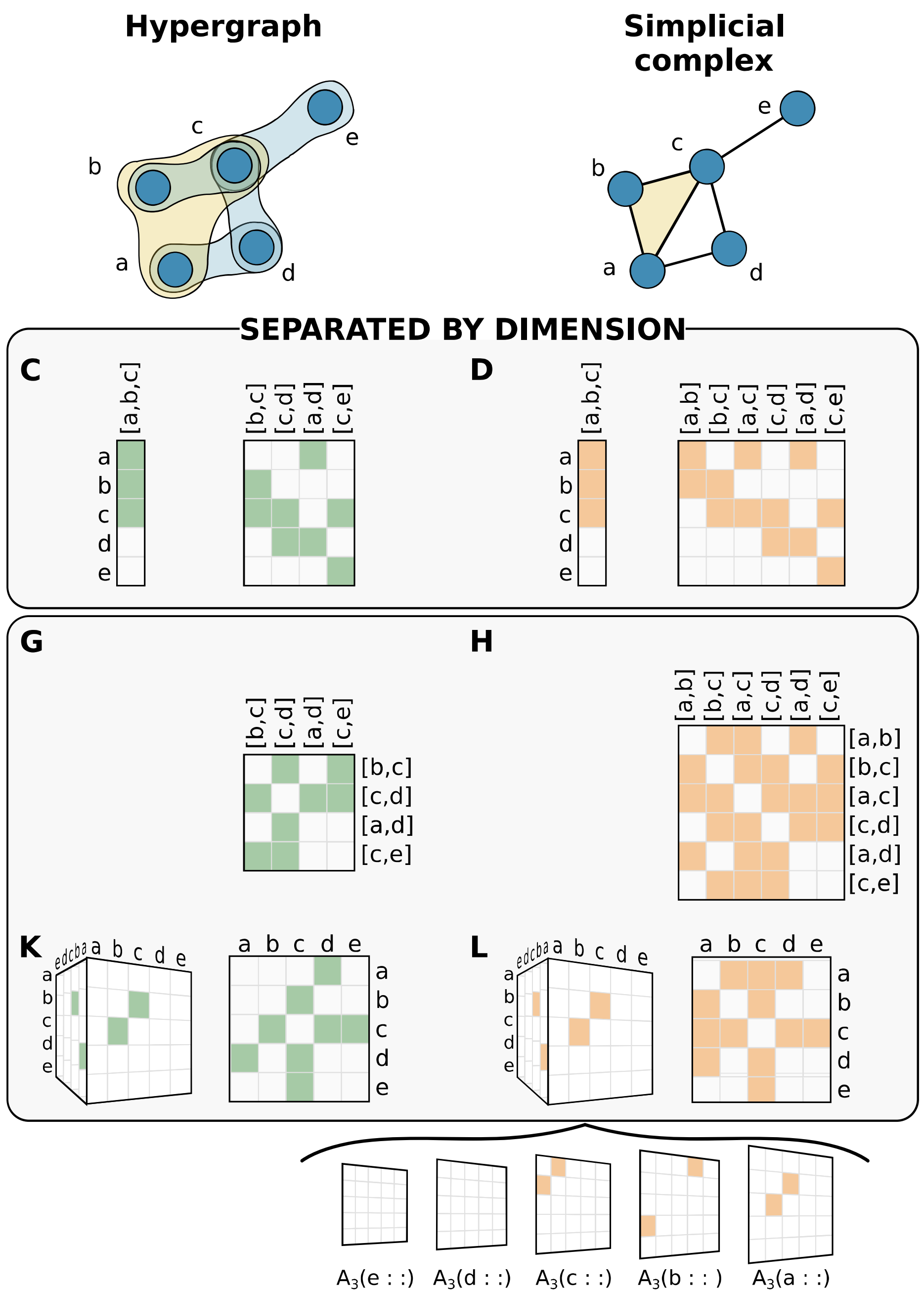
Data structures for SCs



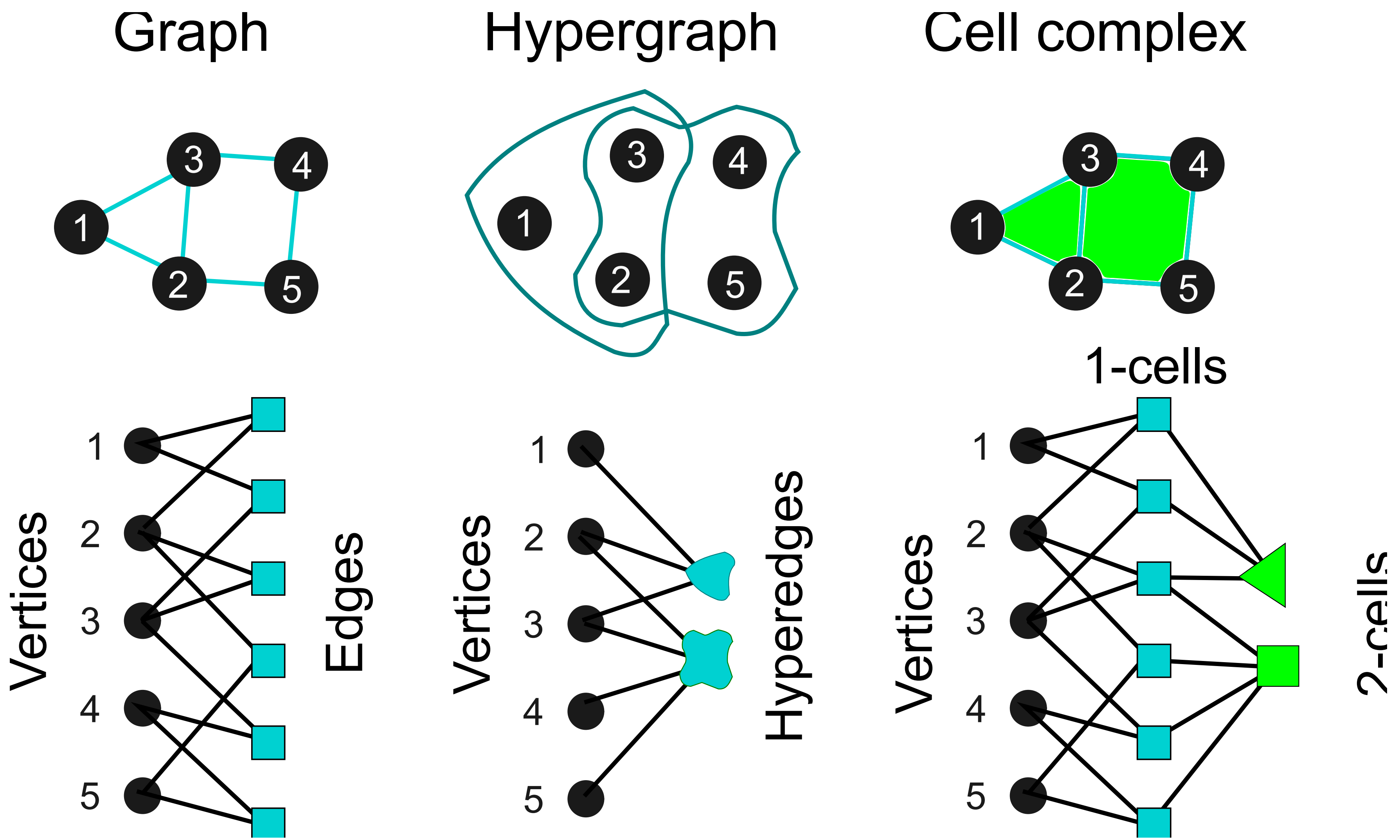
SCs vs Hypergraphs



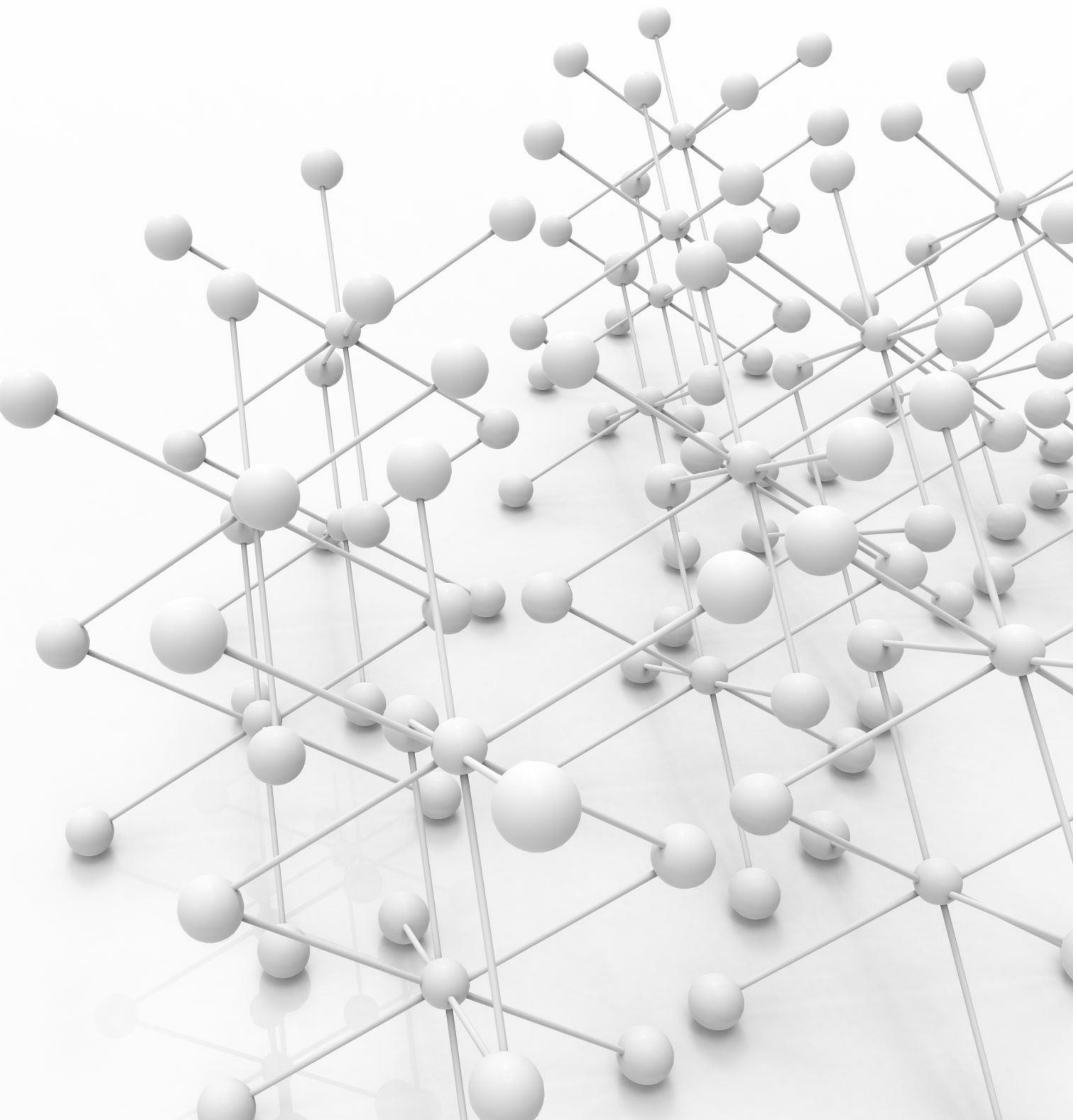
SCs vs Hypergraphs II



Confusion: isn't everything representable as a graph?



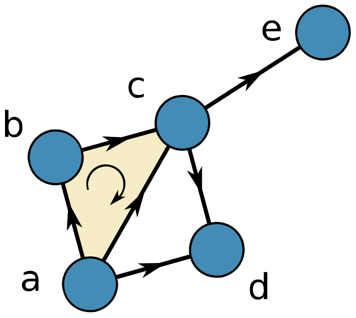
Chapter 2: Higher-order relational data



Chapter 3: Geometry and Topology of Data with Higher-order networks

SCs, Hodge Laplacians etc.

A



Boundary matrices

B

$$B_1 = \begin{array}{c|cccccc|c} & [a,b] & [b,c] & [a,c] & [c,d] & [a,d] & [c,e] & \\ \hline & -1 & & -1 & & -1 & & a \\ & 1 & -1 & & & & & b \\ & & 1 & 1 & -1 & & -1 & c \\ & & & & 1 & 1 & & d \\ & & & & & & 1 & e \end{array}$$

C

$$B_2 = \begin{array}{c|c|c} & [a,b,c] & \\ \hline & 1 & a \\ & 1 & b \\ & -1 & c \\ & & d \\ & & e \end{array}$$

Laplacians

D

$$L_0 = B_1 B_1^T = \begin{bmatrix} -1 & & -1 & & -1 & \\ 1 & -1 & & & & \\ & 1 & 1 & -1 & & -1 \\ & & & 1 & 1 & \\ & & & & & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ -1 & & 1 & & \\ & & -1 & 1 & \\ -1 & & & 1 & \\ & & -1 & & 1 \end{bmatrix} = \begin{array}{ccccc|c} a & b & c & d & e & \\ \hline 3 & -1 & -1 & -1 & & a \\ -1 & 2 & -1 & & & b \\ -1 & -1 & 4 & -1 & -1 & c \\ -1 & & -1 & 2 & & d \\ & & -1 & & 1 & e \end{array}$$

E

$$L_1 = B_1^T B_1 + B_2 B_2^T = \begin{array}{cccccc|c} [a,b] & [b,c] & [a,c] & [c,d] & [a,d] & [c,e] & \\ \hline 3 & & & & 1 & & [a,b] \\ & 3 & & -1 & & -1 & [b,c] \\ & & 3 & -1 & 1 & -1 & [a,c] \\ & -1 & -1 & 2 & 1 & 1 & [c,d] \\ 1 & & 1 & 1 & 2 & & [a,d] \\ & -1 & -1 & 1 & & 2 & [c,e] \end{array}$$

F

$$L_2 = B_2^T B_2 = 3$$

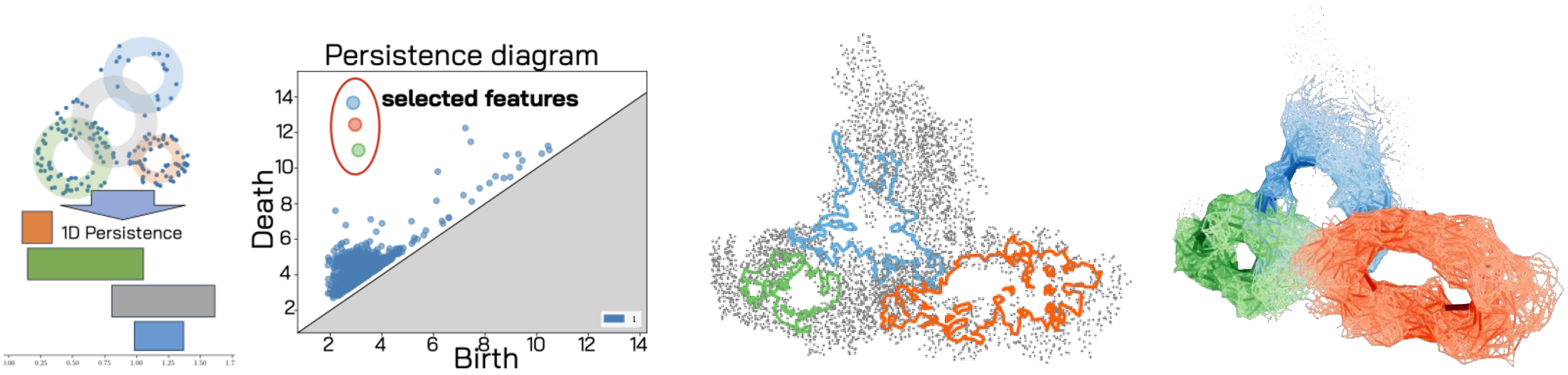
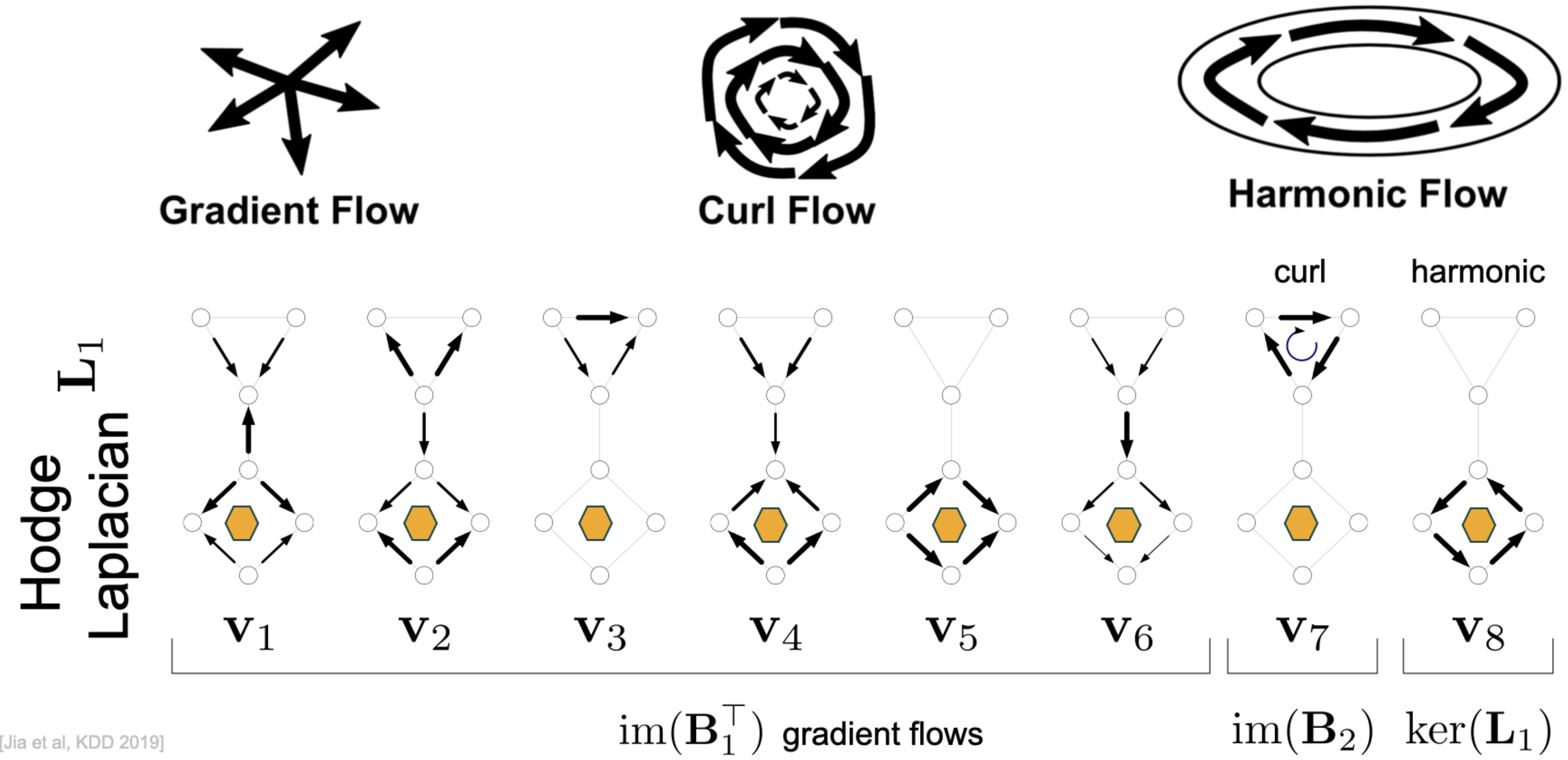


Figure 2: ph sketch and topf pipeline applied to nalcn channelosome, a membrane protein [47]. *Left:* Bars represent life times of features [36]. *Centre left:* Steps 1&2a, when computing persistent 1-homology, three classes are more prominent than the rest. *Centre right:* Step **2b**: The selected homology generators. *Right:* Step **3**: The projections of the generators into harmonic space are now each supported on one of the rings.

The Hodge decomposition for Flows

$$\mathbb{R}^E = \mathrm{im}(\mathbf{B}_1^\top) \oplus \mathrm{im}(\mathbf{B}_2) \oplus \ker(\mathbf{L}_1)$$



[Jia et al, KDD 2019]

Chapter 4: Dynamics and Data on top of higher-order networks

Example: linear diffusion dynamics on hypergraphs

- Consider a dynamics defined on a 3-uniform hypergraph of the form

$$\dot{x}_i = \sum_{j,k=1}^{N} A_{ijk} f_i^{(jk)}(x_i, x_j, x_k)$$

- The function f_i^{jk} describes how the states of nodes *j* and *k* affect node *i*.
- The hypergraph adjacency tensor A_{ijk} is 1 if there is a connection and 0 if not.

Observation

- If the function is *f(x)* is linear in *x,* then we can always write this is a “network dynamical system”

$$\dot{x}_i = \sum_j \mathcal{A}_{ij} x_j$$

Example continued

Let's set interaction function to:

$$f_i^{\{jk\}}(x_i, x_j, x_k) = c((x_j - x_i) + (x_k - x_i))$$

Then

$$\dot{x}_i = \sum_{jk} A_{ijk} c((x_j - x_i) + (x_k - x_i))$$

$$= 2c \sum_{jk} A_{ijk}(x_j - x_i) = -2c \sum_{j} (L_T)_{ij} x_j,$$

where

$$(L_T)_{ij} = (D_T - W_T)_{ij} \qquad (D_T)_{ii} = \sum_{kj} A_{ijk} \qquad (W_T)_{ij} = \sum_{k} A_{ijk}.$$

What have we learned?

- A “network dynamics” with linear interaction function can always be written in terms of a dynamical system on a graph.

- Nonlinear behavior *necessary* for “true” higher-order effects
- Nonlinearity enough? No!

Example

Kuramoto like dynamics on hypergraph with vertex set V and edges $E_a \subseteq V$

$$\dot{x}_i = \sum_{\alpha : i \in E_\alpha} \sum_{j \in E_\alpha} \sin(x_j - x_i) = \sum_{j=1}^{N} (B^\top B)_{ij} \sin(x_j - x_i)$$

Confusion II: The Koopman operator / lifting argument

- Any nonlinear system can be represented via a lifted linear system on an *extended* (infinite dimensional) state space via the Koopman operator
- Hence, everything can be represented as a network dynamics??!

Example

$$\dot{x}_1 = \mu x_1,$$
$$\dot{x}_2 = \lambda\left(x_2 - x_1^2\right).$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_1^2 \end{bmatrix} \Rightarrow \frac{d}{dt}\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \underbrace{\begin{bmatrix} \mu & 0 & 0 \\ 0 & \lambda & -\lambda \\ 0 & 0 & 2\mu \end{bmatrix}}_{\mathrm{K}} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}.$$

- However, in general the state space is *infinite dimensional* and new state variables are typically not *localized*!

Why are localized state variables important for network interpretation?

Example

Consider a linear dynamics on an undirected graph

$$\dot{x}_i = \sum_j \mathcal{A}_{ij} x_j$$

This can always be diagonalized using spectral coordinates, in which we have:

$$\dot{y}_i = \lambda_i y_i$$

But

$$y_i = v_i^\top x$$

depends on all node states!