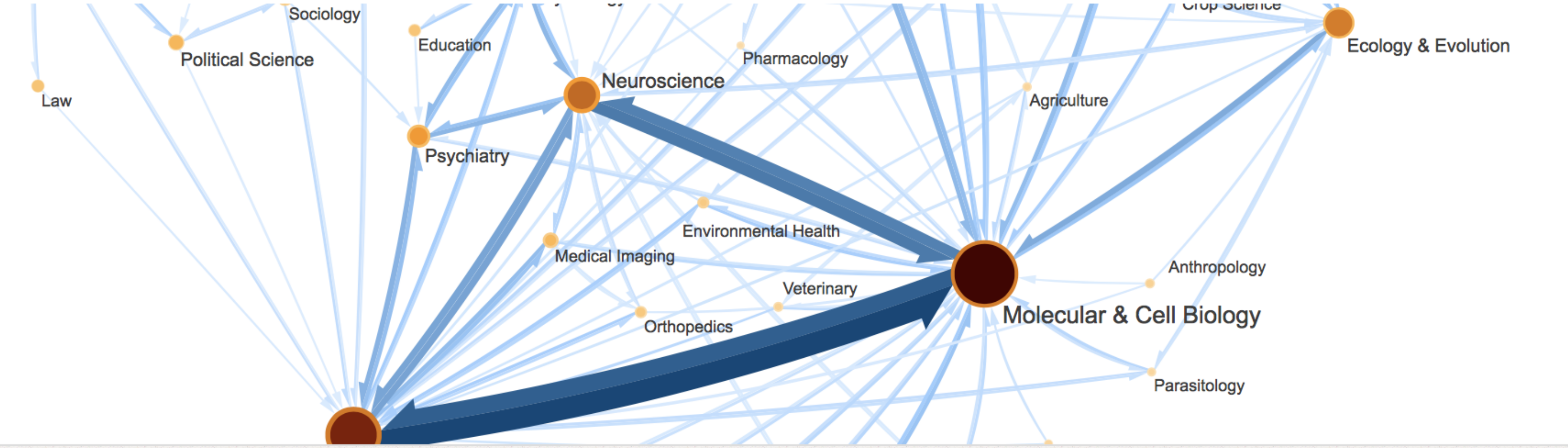
Mapping higher-order network flows

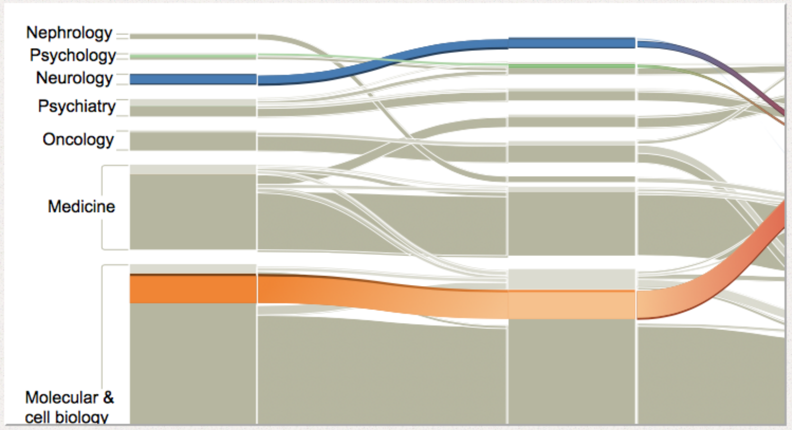
With the minimum description length principle

Anton Eriksson, Daniel Edler, and
Martin Rosvall

Simplify and highlight important structures in complex networks



Apps »



Code »

```
using infomath::plogp;
for (unsigned int i = 0; i < numNodes; ++i)
{
enter_log_enter += plogp(m_moduleFlowData[i].enter
exit_log_exit += plogp(m_moduleFlowData[i].exitFlo
flow_log_flow += plogp(m_moduleFlowData[i].exitFlo
enterFlow += m_moduleFlowData[i].enterFlow;
}
enterFlow += exitNetworkFlow;
enterFlow_log_enterFlow = plogp(enterFlow);
```

Publications »

Maps of information flow reveal community structure in complex networks

Martin Rosvall and Carl T. Bergstrom

PNAS **105**, 1118 (2008). [arXiv:0707.0609]

To comprehend the multipartite organization of large-scale biological and social systems, we introduce a new information-theoretic approach to reveal community structure in

News

Nov 22, 2017 Research paper — Mapping intermittent communities — efficiently reveal intermittent communities in temporal networks
Sep 30, 2017 Research paper — Mapping higher-order network flows — in memory and multilayer networks with Infomap
April 11, 2017 Source code — Infomap on Windows — run Infomap in bash on ubuntu on Windows
July 4, 2016 Reserach paper — Maps of sparse Markov chains — efficiently reveal community structure in network flows with memory
December 5, 2015 Mapping tool — Infomap Bioregions — interactive mapping of biogeographical regions from species distributions
September 5, 2015 Source code — GossipMap — a distributed implementation of infomap by Seung-Hee Bae and Bill Howe
August 13, 2015 Interactive storyboard — Multilevel network sampling — infer network modes from multiple samples
August 13, 2015 Interactive storyboard — Higher-order Markov models — identify flows on memory and multilayer networks
July 23, 2015 Source code — Infomap — updates to memory and multilayer algorithms
April 1, 2015 Source code — Infomap with other languages — use Infomap as a library wity Python, iGraph, or R
March 6, 2015 Paper — Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems — Phys Rev X 5, 011027 (2015)

Coding theory

The minimum description length principle

Claude Shannon, David Huffmann, Jorma Rissanen, …

Network flows

The map equation

Carl Bergstrom, Daniel Edler, Alcides Viamontes Esquivel, Renaud Lambiotte, Martin Rosvall, …

Higher-order flows

Sparse Markov models

Andrea Lancichinetti, Renaud Lambiotte, Manlio de Domenico, Alex Arenas, Christian Persson, Ludvig Bohlin, Daniel Edler, Martin Rosvall, …

Coding theory

The minimum description length principle

Claude Shannon, David Huffmann, Jorma Rissanen, …

Network flows

The map equation

Carl Bergstrom, Daniel Edler, Alcides Viamontes Esquivel, Renaud Lambiotte, Martin Rosvall, …

Higher-order flows

Sparse Markov models

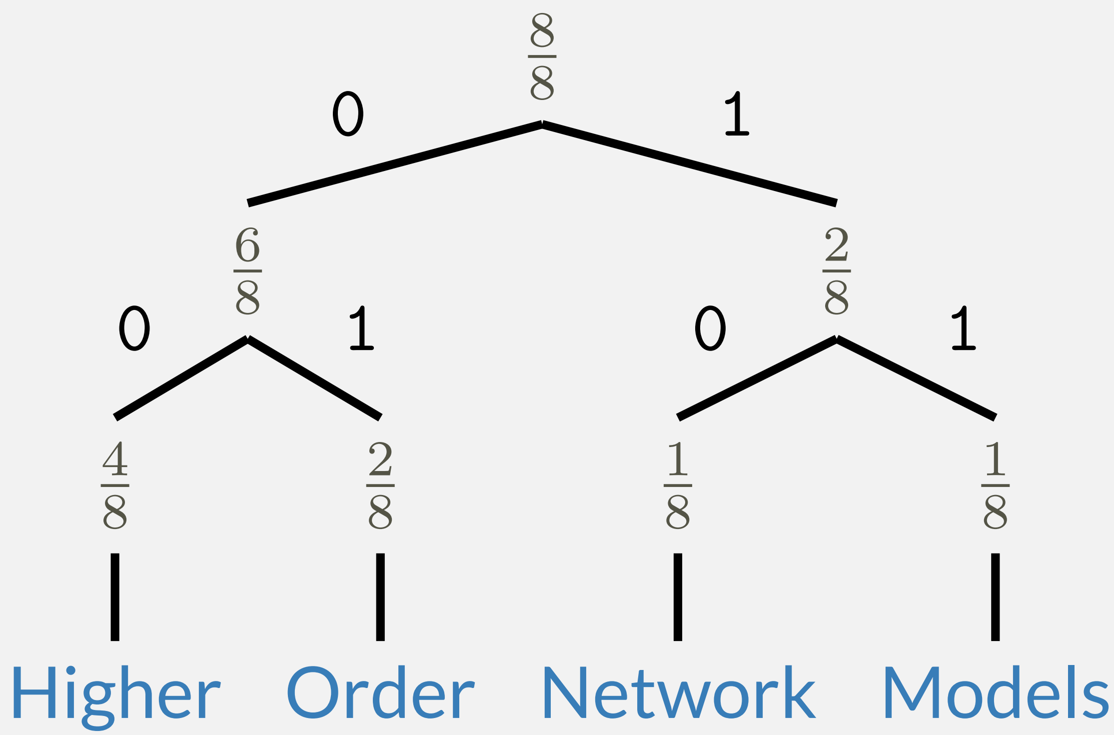
Andrea Lancichinetti, Renaud Lambiotte, Manlio de Domenico, Alex Arenas, Christian Persson, Ludvig Bohlin, Daniel Edler, Martin Rosvall, …

$X = \{$Higher, Order, Network, Models$\}$
$\mathcal{P} = \{P(\text{Higher}) = \frac{1}{2}, P(\text{Order}) = \frac{1}{4}, P(\text{Network}) = \frac{1}{8}, P(\text{Models}) = \frac{1}{8}\}$

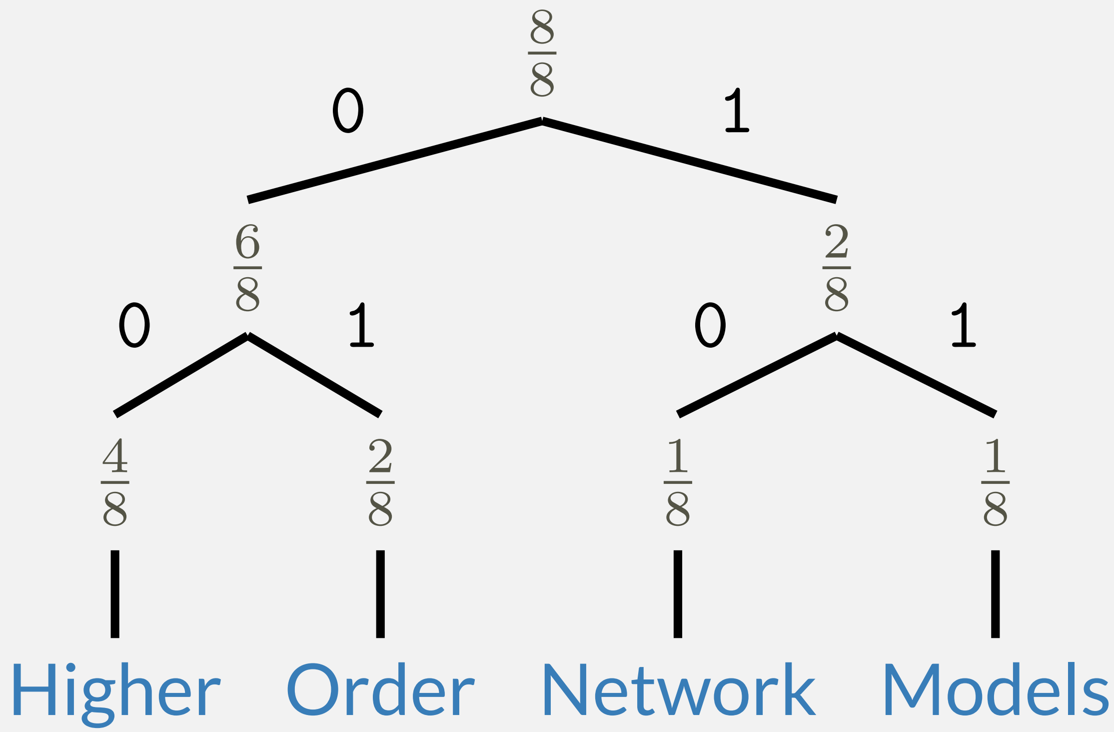
Higher Network Network Higher Higher Higher Order Network Higher Higher Network Higher Network Higher Higher Order Higher Models Models Higher Higher Order Models Higher Higher Higher Higher Higher Higher Higher Higher Higher Higher Higher Higher Higher Network Order Order Higher Models Network Higher Higher Order Order Models Higher Network Order Higher Order Models Network Order Order Higher Higher Network Higher Higher Order Order Order Higher Higher Order Network Order Higher Higher Higher Higher Order Models Order Higher Higher Order Models Network Order Network Higher Order Models Order Higher Higher Order Models Higher Network Network Models Order Network Order Order

Higher = 00, Order = 01, Network = 10, Models = 11

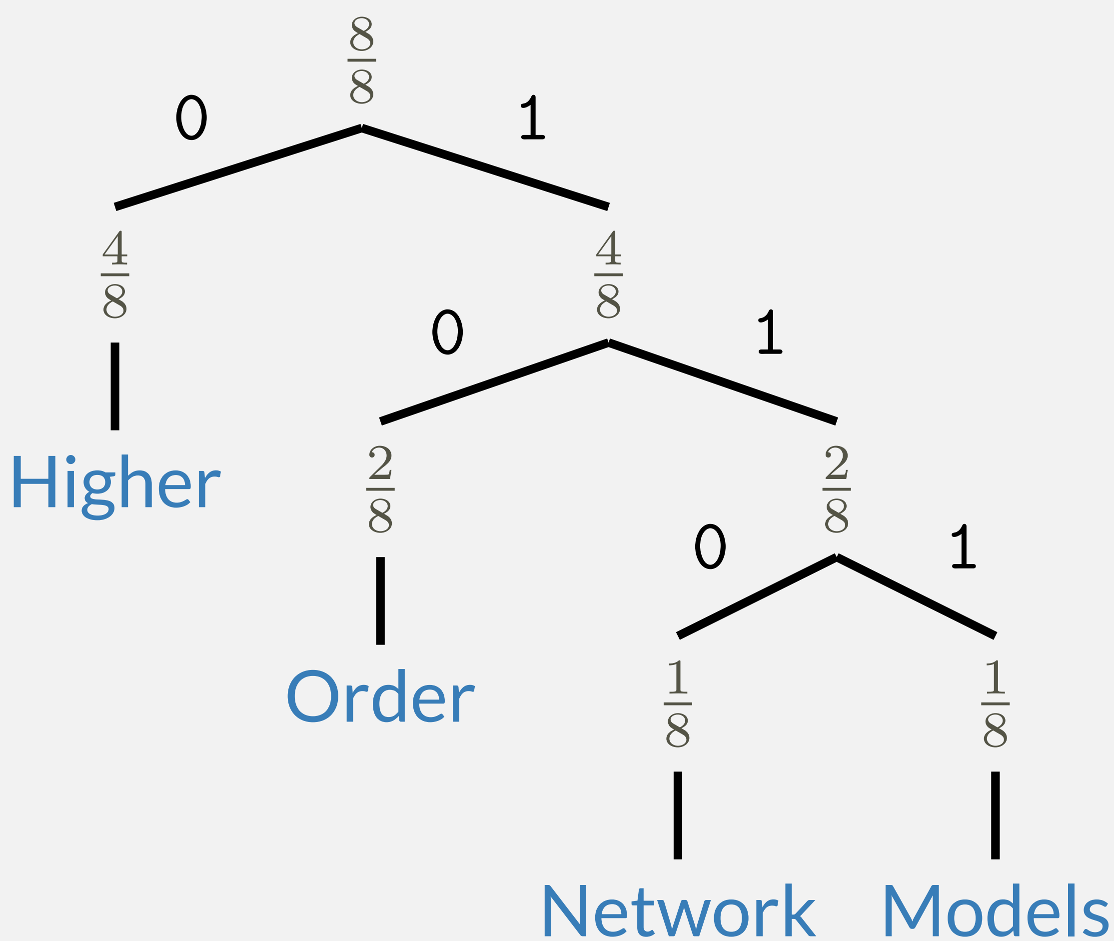
00 10 10 00 00 00 01 10 00 00 10 00 10 00 00 01 00 11 11 00 00 01 11 00 00 00 00 00 00 00 00 00 00 00 00 00 10 01 01 00 11 10 00 00 01 01 11 00 10 01 00 01 11 10 01 01 00 00 10 00 00 01 01 01 00 00 01 10 01 00 00 00 00 01 11 01 00 00 01 11 10 01 10 00 01 11 01 00 00 01 11 00 10 10 11 01 10 01 01



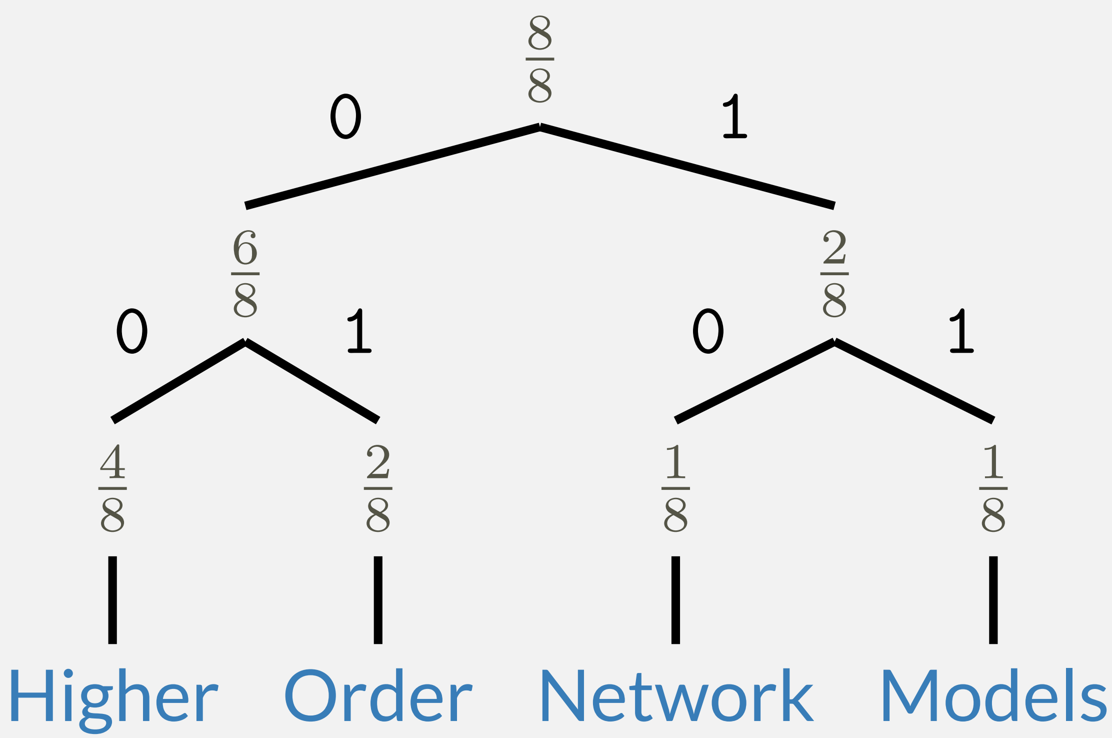
00101000000001100000100010000001001-
11100000111000000000000000000000000-
00100101001110000001011100100100011-
11001010000100000010101000001100100-
00000001110100000111100110000111010-
00001110010101101100101



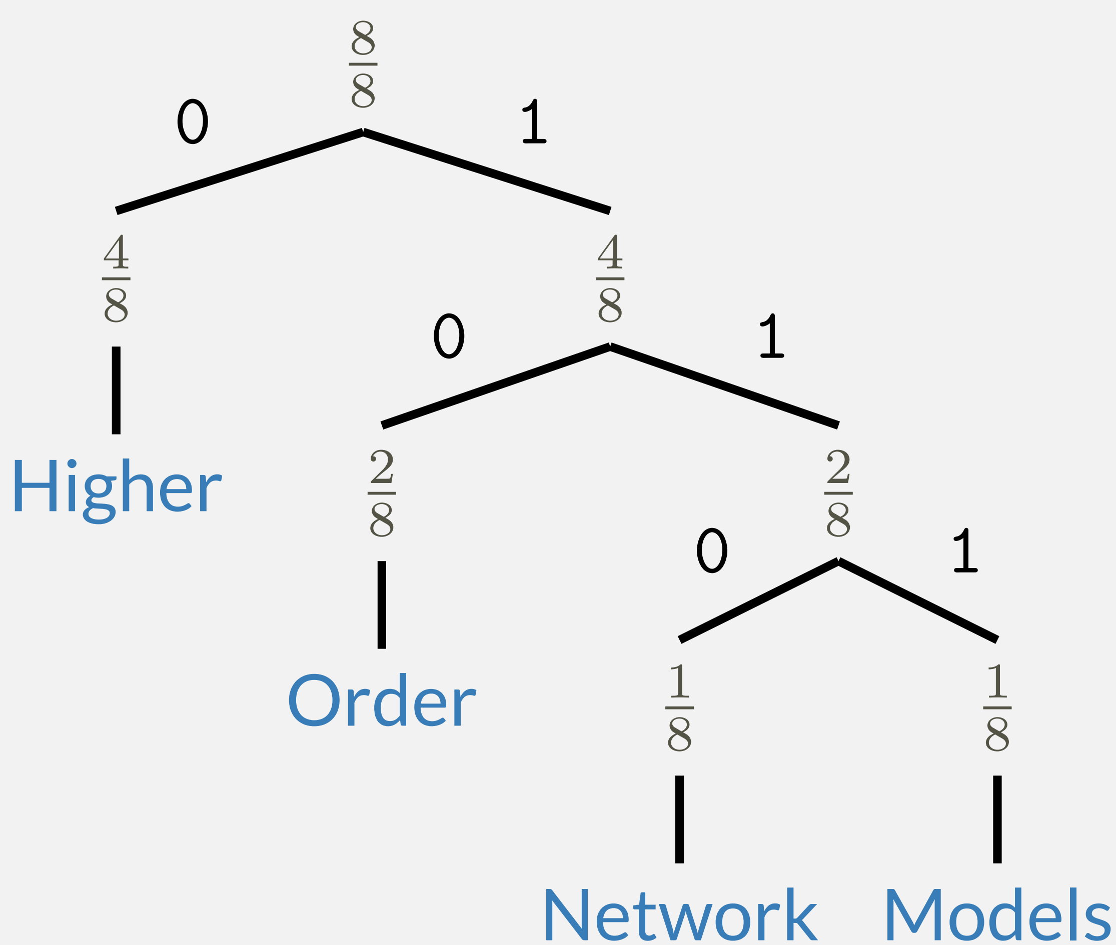
00101000000001100000100010000001001-
11100000111000000000000000000000000-
00100101001110000001011100100100011-
11001010000100000010101000001100100-
00000001110100000111100110000111010-
0000111001010101101100101



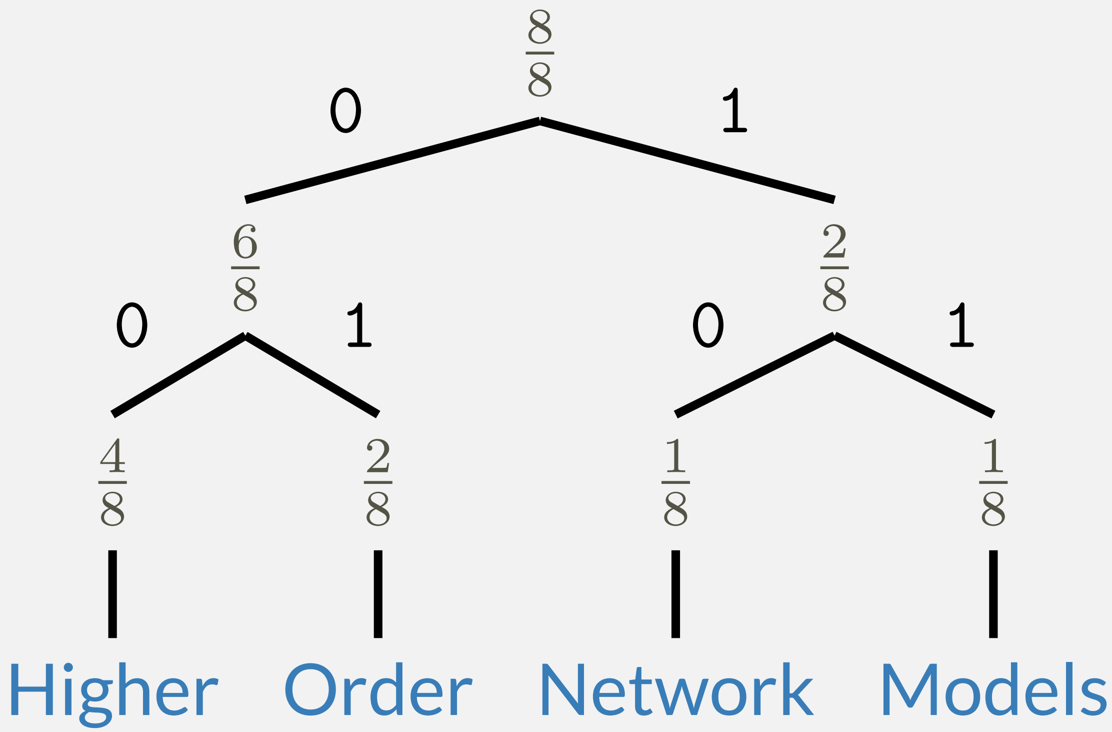
01101100001011000110011000100111111-
00101110000000000000110101001111100-
010101110110100101111110101000110001-
010100010110100000010111100010111110-
10110010111100010111011011011110110-
1010



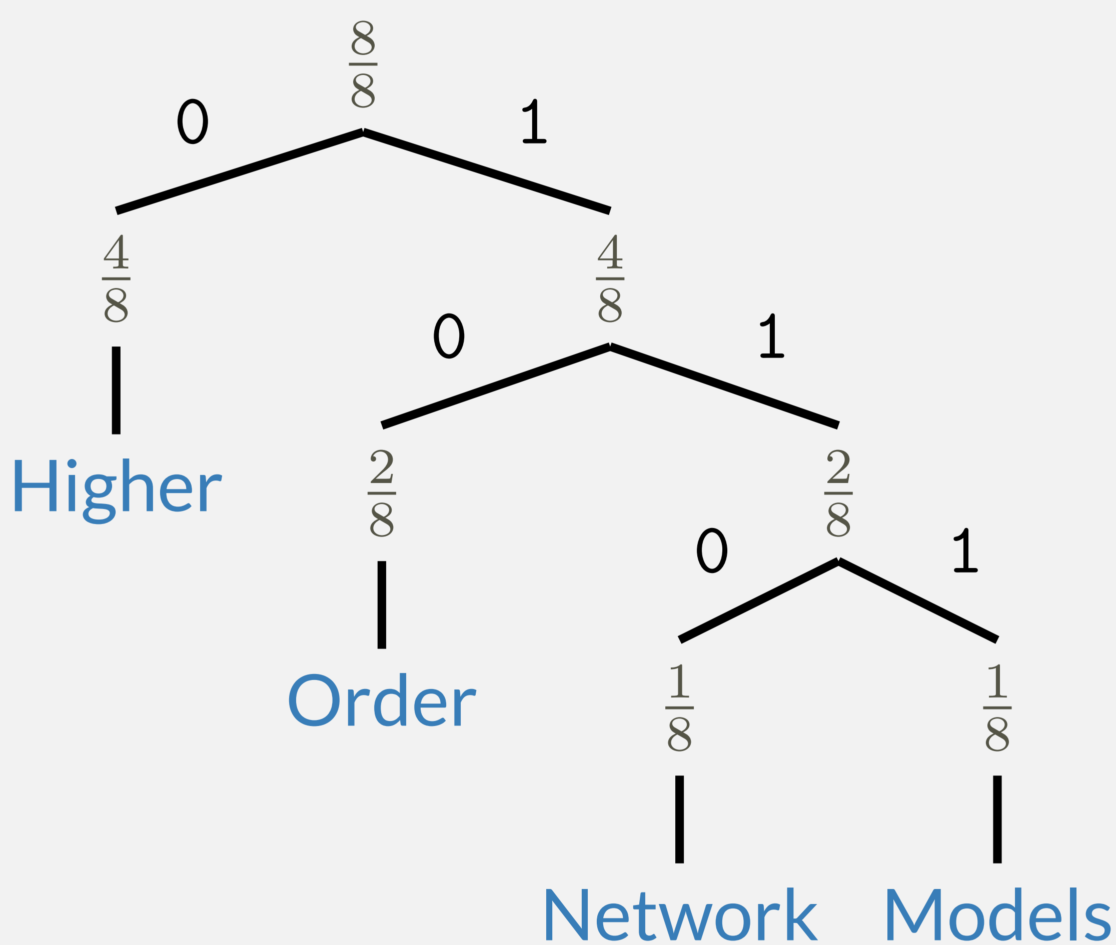
$$L = \sum_i p_i l_i = \frac{4}{8}2 + \frac{2}{8}2 + \frac{1}{8}2 + \frac{1}{8}2 = \mathbf{2}$$



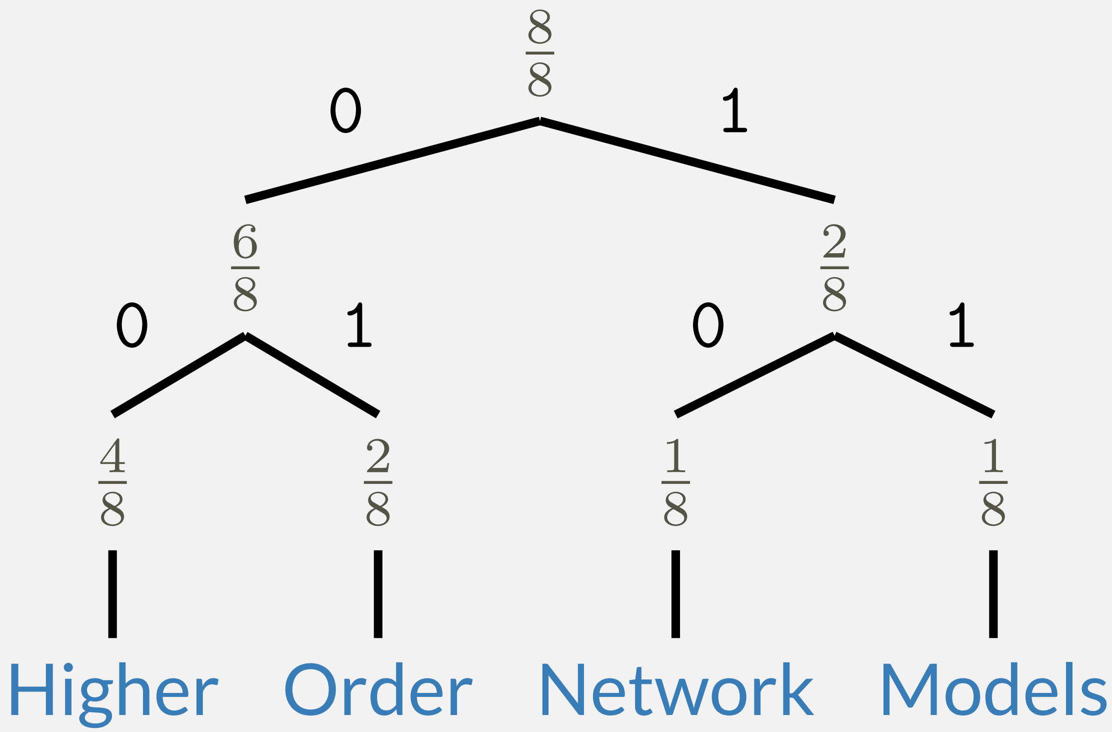
$$L = \sum_i p_i l_i = \frac{4}{8}1 + \frac{2}{8}2 + \frac{1}{8}3 + \frac{1}{8}3 = \mathbf{\frac{7}{4}}$$



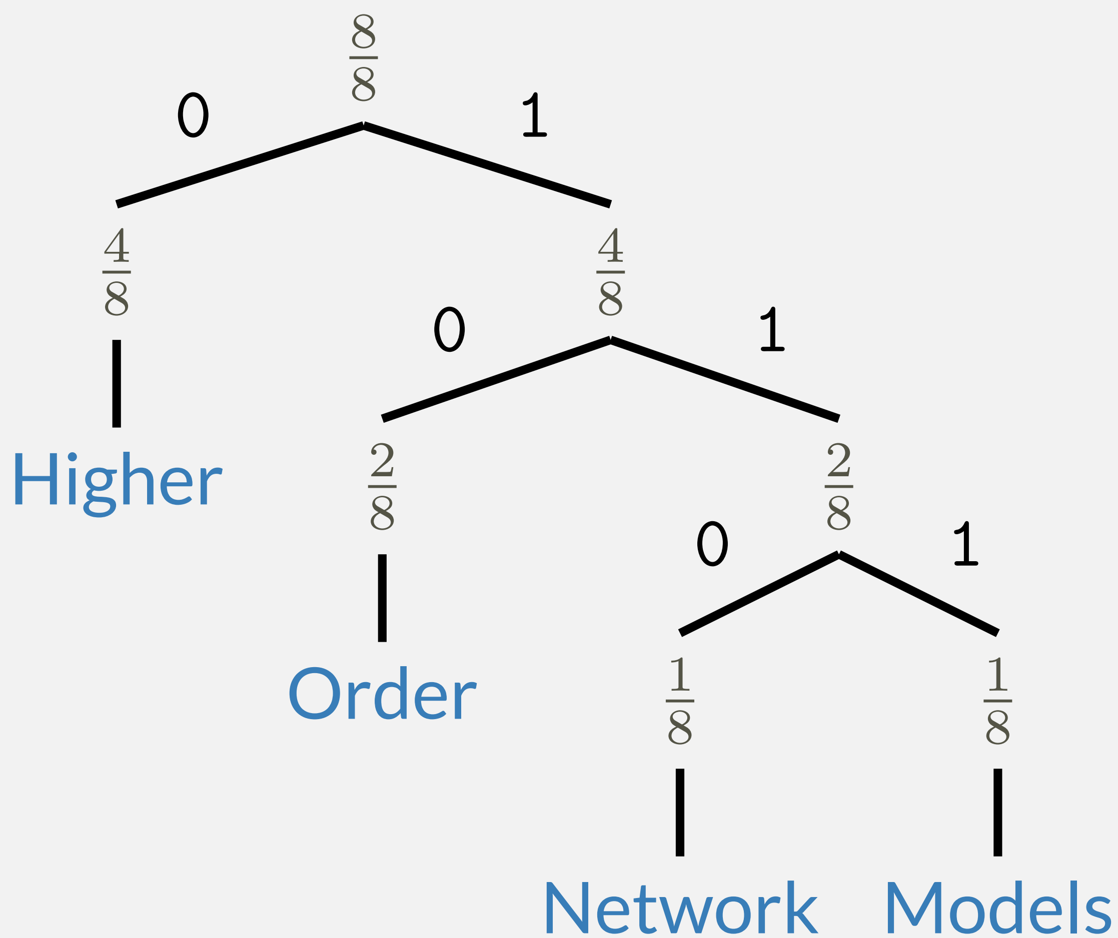
$$L = \sum_i p_i l_i = \frac{4}{8}2 + \frac{2}{8}2 + \frac{1}{8}2 + \frac{1}{8}2 = \mathbf{2}$$



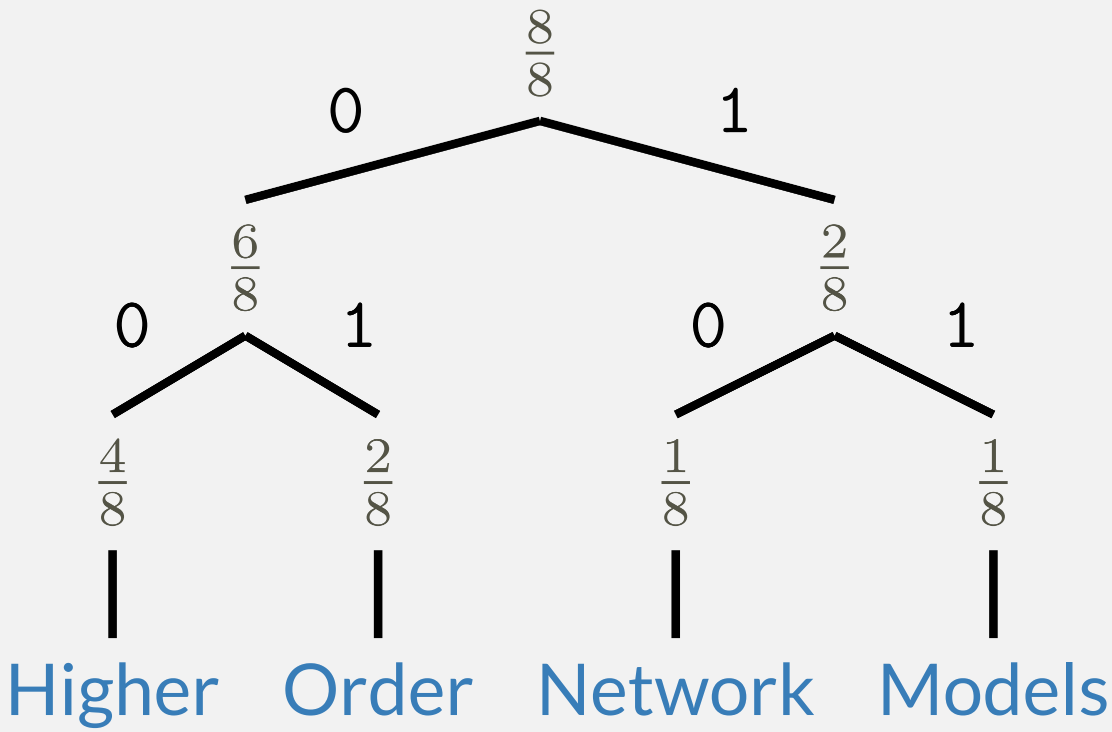
$$L = \sum_i p_i l_i = \frac{4}{8}1 + \frac{2}{8}2 + \frac{1}{8}3 + \frac{1}{8}3 = \mathbf{\frac{7}{4}}$$
$$= \frac{4}{8}\log_2\frac{8}{4} + \frac{2}{8}\log_2\frac{8}{2} + \frac{1}{8}\log_2\frac{8}{1} + \frac{1}{8}\log_2\frac{8}{1}$$



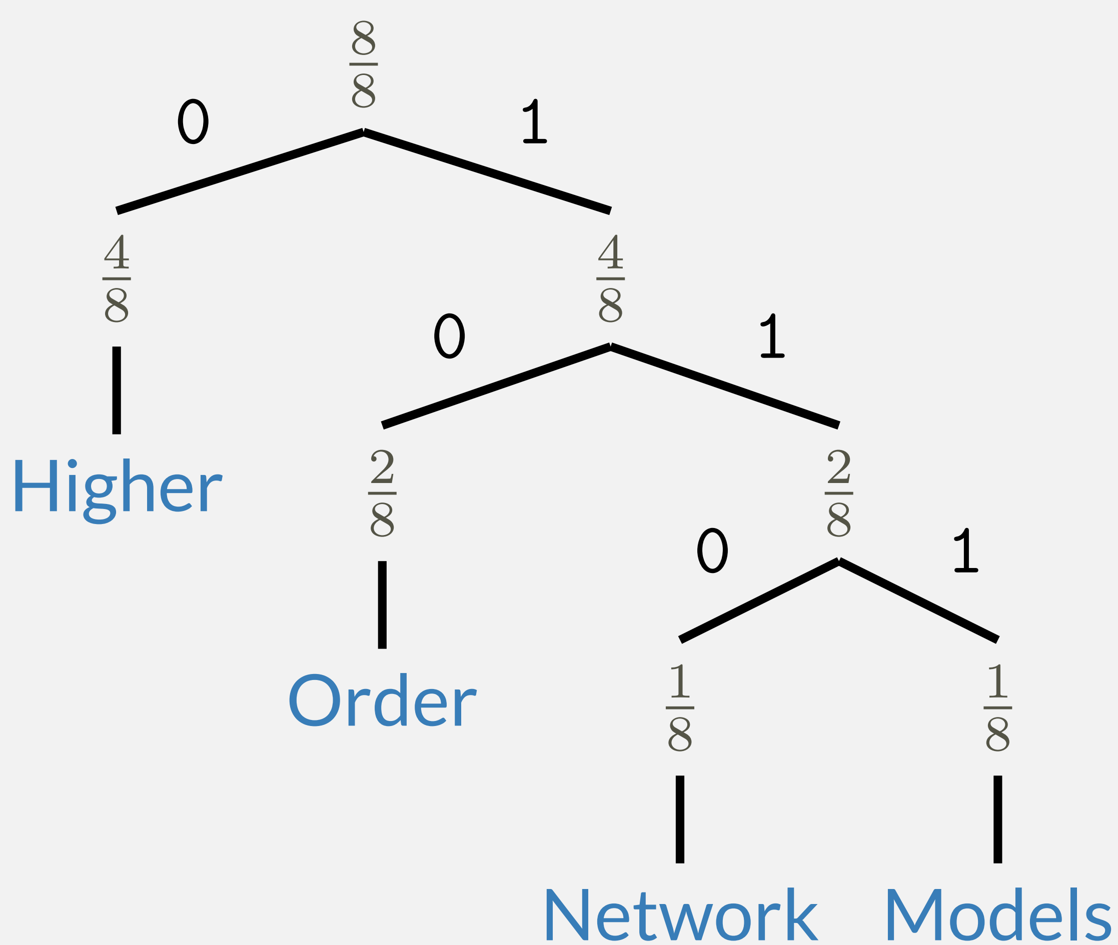
$$L = \sum_i p_i l_i = \frac{4}{8}2 + \frac{2}{8}2 + \frac{1}{8}2 + \frac{1}{8}2 = \mathbf{2}$$



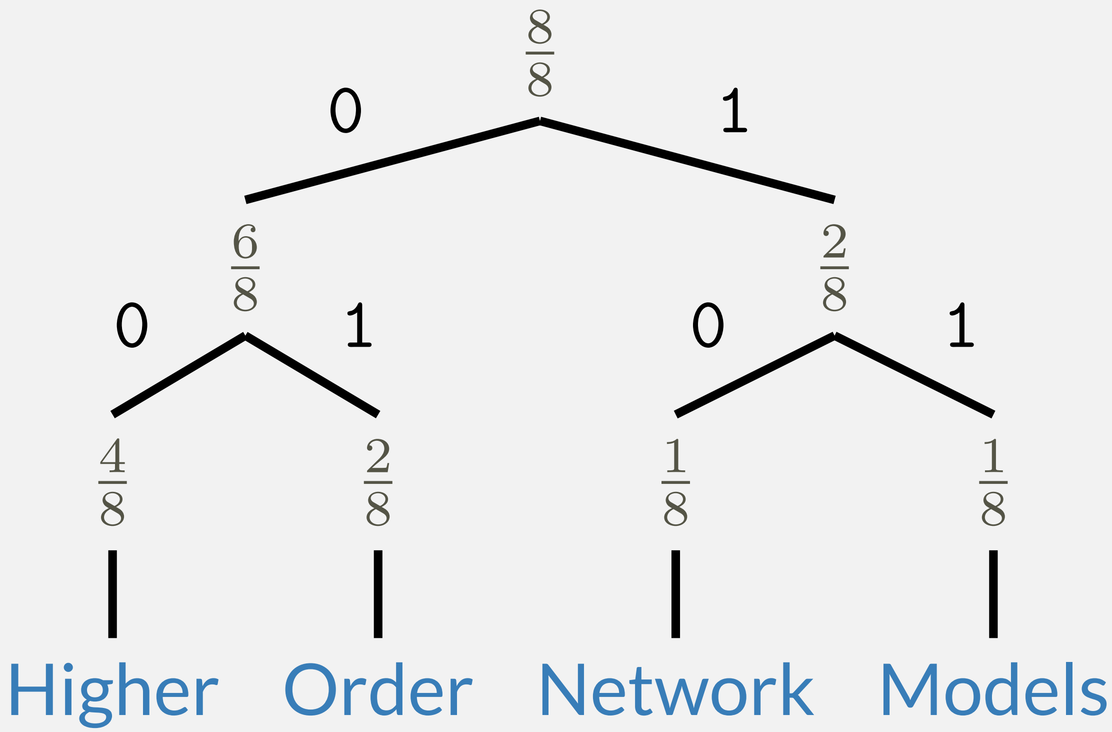
$$\begin{aligned} L = \sum_i p_i l_i &= \frac{4}{8}1 + \frac{2}{8}2 + \frac{1}{8}3 + \frac{1}{8}3 = \mathbf{\frac{7}{4}} \\ &= \frac{4}{8}\log_2\frac{8}{4} + \frac{2}{8}\log_2\frac{8}{2} + \frac{1}{8}\log_2\frac{8}{1} + \frac{1}{8}\log_2\frac{8}{1} \\ &= -\frac{4}{8}\log_2\frac{4}{8} - \frac{2}{8}\log_2\frac{2}{8} - \frac{1}{8}\log_2\frac{1}{8} - \frac{1}{8}\log_2\frac{1}{8} \end{aligned}$$



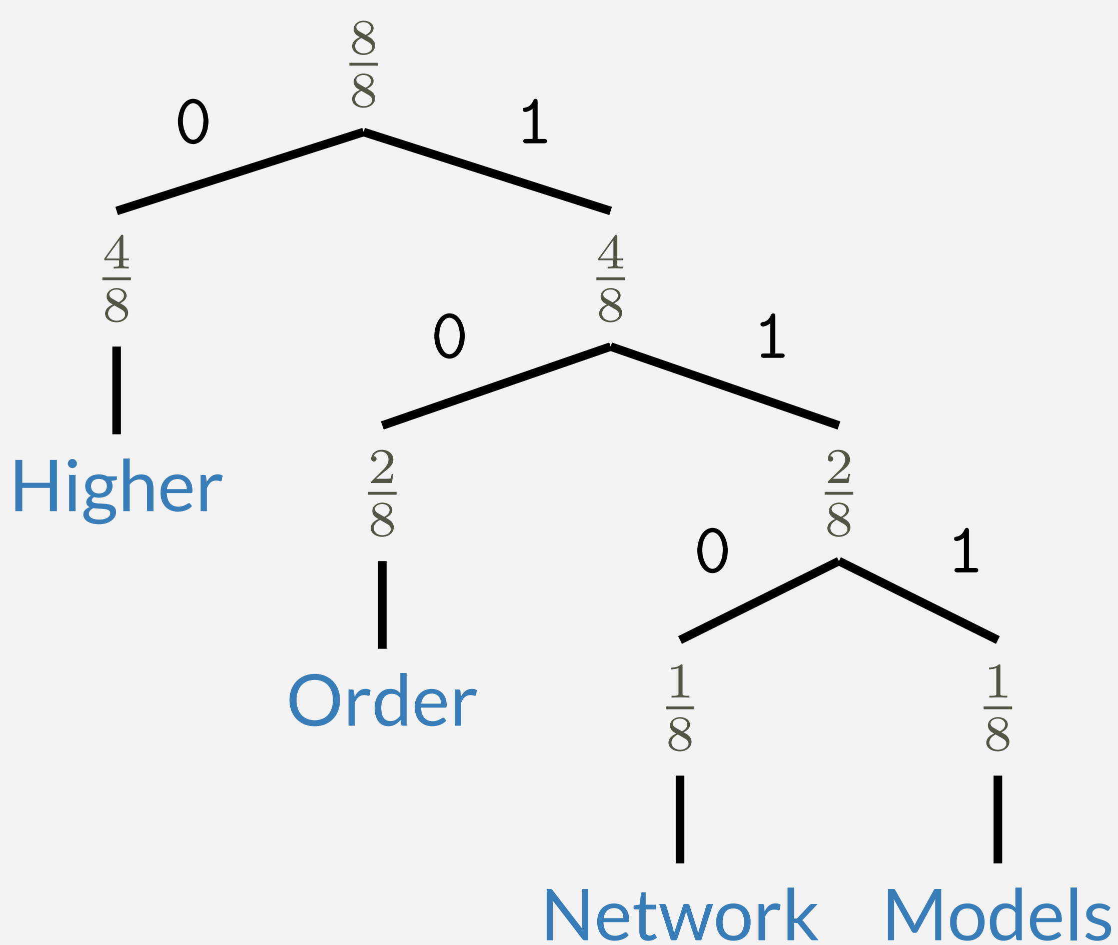
$$L = \sum_i p_i l_i = \frac{4}{8}2 + \frac{2}{8}2 + \frac{1}{8}2 + \frac{1}{8}2 = \mathbf{2}$$



$$\begin{aligned} L = \sum_i p_i l_i &= \frac{4}{8}1 + \frac{2}{8}2 + \frac{1}{8}3 + \frac{1}{8}3 = \frac{\mathbf{7}}{\mathbf{4}} \\ &= \frac{4}{8}\log_2\frac{8}{4} + \frac{2}{8}\log_2\frac{8}{2} + \frac{1}{8}\log_2\frac{8}{1} + \frac{1}{8}\log_2\frac{8}{1} \\ &= -\frac{4}{8}\log_2\frac{4}{8} - \frac{2}{8}\log_2\frac{2}{8} - \frac{1}{8}\log_2\frac{1}{8} - \frac{1}{8}\log_2\frac{1}{8} \\ &= -\sum_i p_i \log_2 p_i \end{aligned}$$



$$L = \sum_i p_i l_i = \frac{4}{8}2 + \frac{2}{8}2 + \frac{1}{8}2 + \frac{1}{8}2 = \mathbf{2}$$



$$\begin{aligned}
L = \sum_i p_i l_i &= \frac{4}{8}1 + \frac{2}{8}2 + \frac{1}{8}3 + \frac{1}{8}3 = \mathbf{\frac{7}{4}} \\
&= \frac{4}{8}\log_2 \frac{8}{4} + \frac{2}{8}\log_2 \frac{8}{2} + \frac{1}{8}\log_2 \frac{8}{1} + \frac{1}{8}\log_2 \frac{8}{1} \\
&= -\frac{4}{8}\log_2 \frac{4}{8} - \frac{2}{8}\log_2 \frac{2}{8} - \frac{1}{8}\log_2 \frac{1}{8} - \frac{1}{8}\log_2 \frac{1}{8} \\
&= -\sum_i p_i \log_2 p_i \\
&\equiv H(\mathcal{P})
\end{aligned}$$

Summary of coding theory

Q: How can we use information theory for inference?

A: The minimum description length principle: Regularities in data can be used to compress the data and the best compression captures most regularities.

Coding theory

The minimum description length principle

Claude Shannon, David Huffmann, Jorma Rissanen, …

Network flows

The map equation

Carl Bergstrom, Daniel Edler, Alcides Viamontes Esquivel, Renaud Lambiotte, Martin Rosvall, …

Higher-order flows

Sparse Markov models

Andrea Lancichinetti, Renaud Lambiotte, Manlio de Domenico, Alex Arenas, Christian Persson, Ludvig Bohlin, Daniel Edler, Martin Rosvall, …

NETWORKS describe where flows move to depending on where they are

MAPS depict regularities using less information

If we can find a good code
for describing flows on a network,
we will have solved the dual problem
of finding the important structures
with respect to that flow

We use a modular code structure
that can exploit regions in the network
in which units of flow tend to stay
for a relatively long time

Two-level partitions

How many modules are present? And which nodes are members of which modules?

Maximal compression of flow with constraints:

1. Modular code structure
2. No more than two levels
3. Each node can only belong to one module

Two-level partitions with the map equation

How many modules are present? And which nodes are members of which modules?

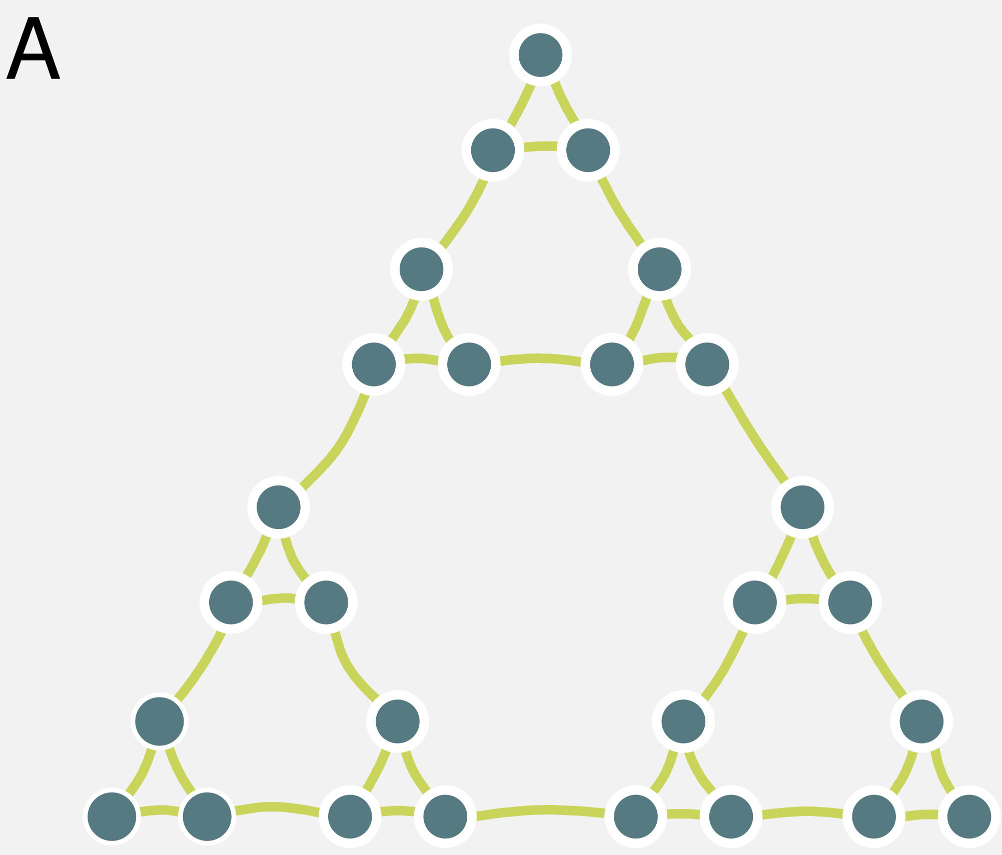
Maximal compression of flow with constraints:

1. Modular code structure
2. No more than two levels
3. Each node can only belong to one module

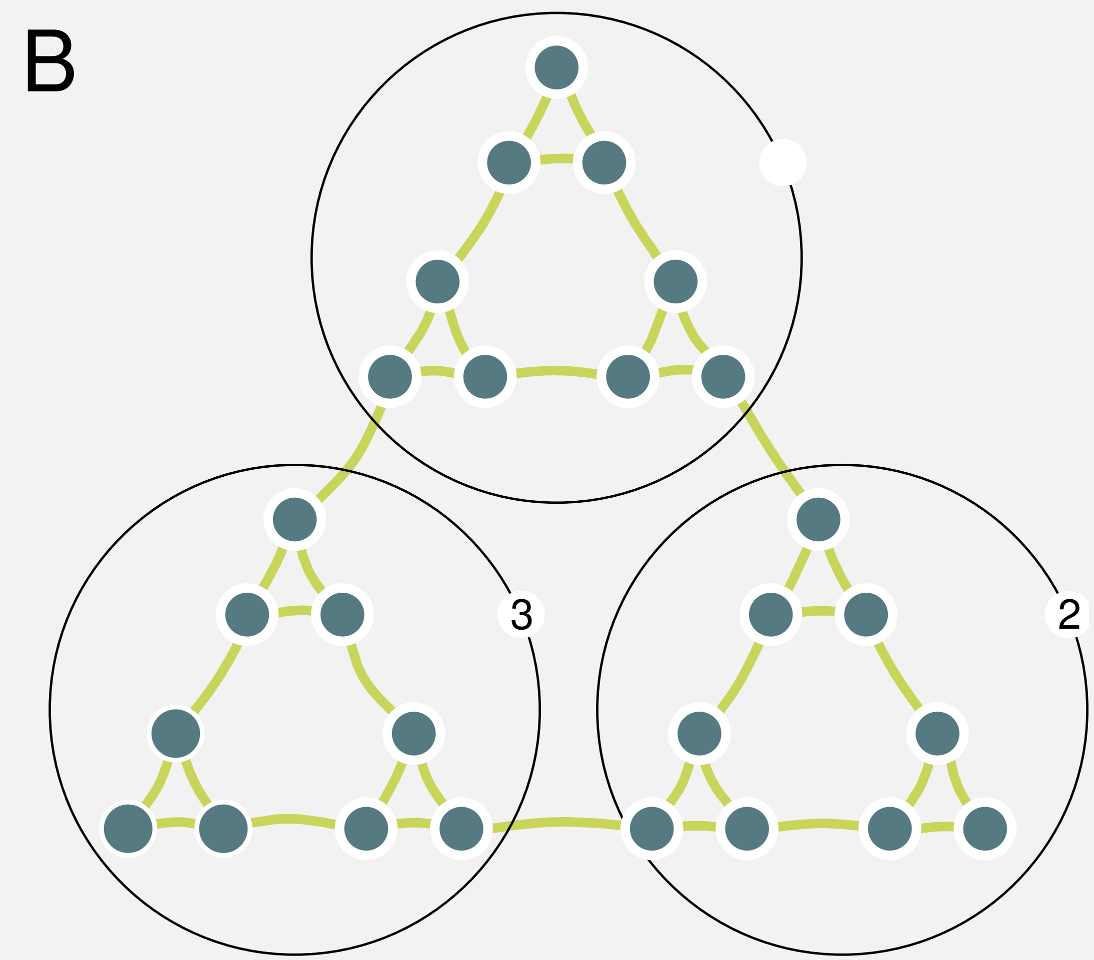
Two-level partitions with the map equation

$$L(\mathsf{M}) = q_{\curvearrowleft} H(\mathcal{Q}) + \sum_{i=1}^{m} p^{i}_{\circlearrowright} H(\mathcal{P}^{i})$$

Two-level partitions with the map equation



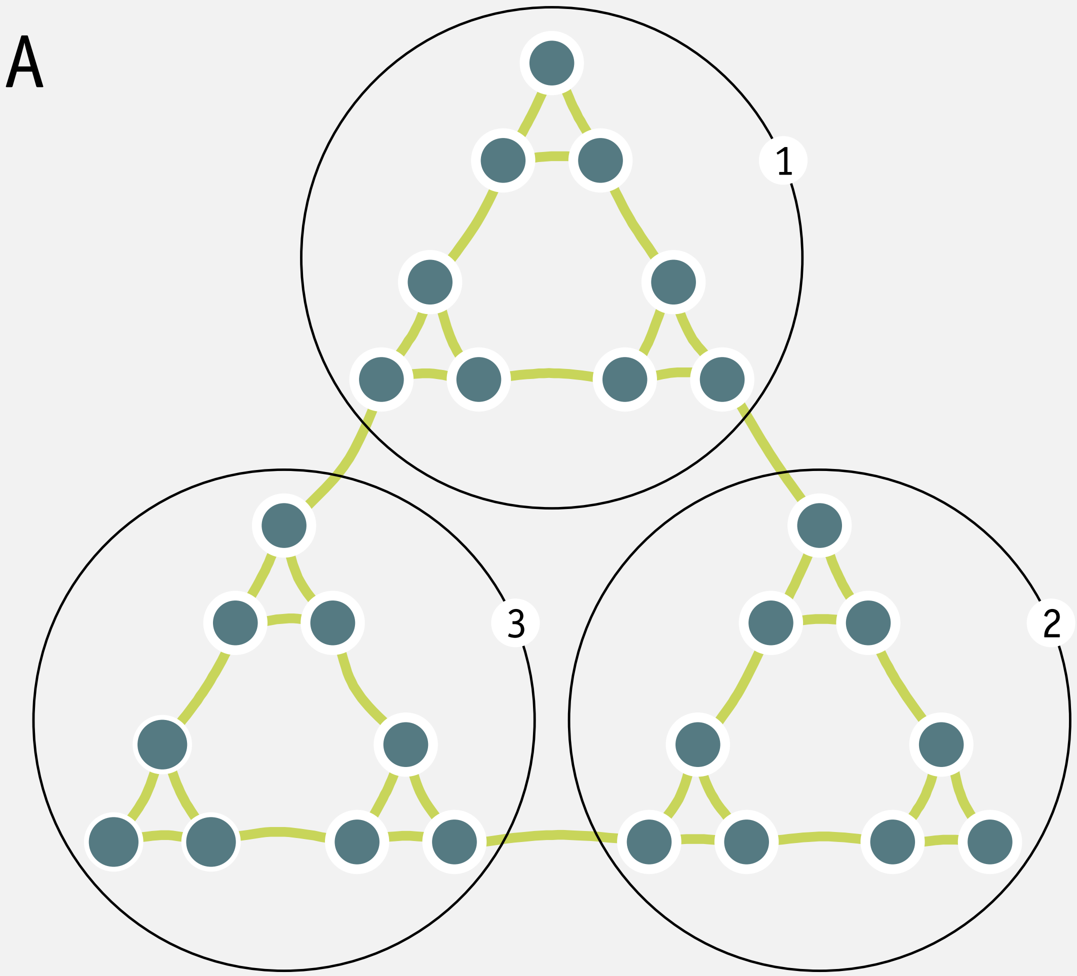
$$L(\mathsf{M}) = H(\mathcal{P}) = 4.75 \text{ bits.}$$



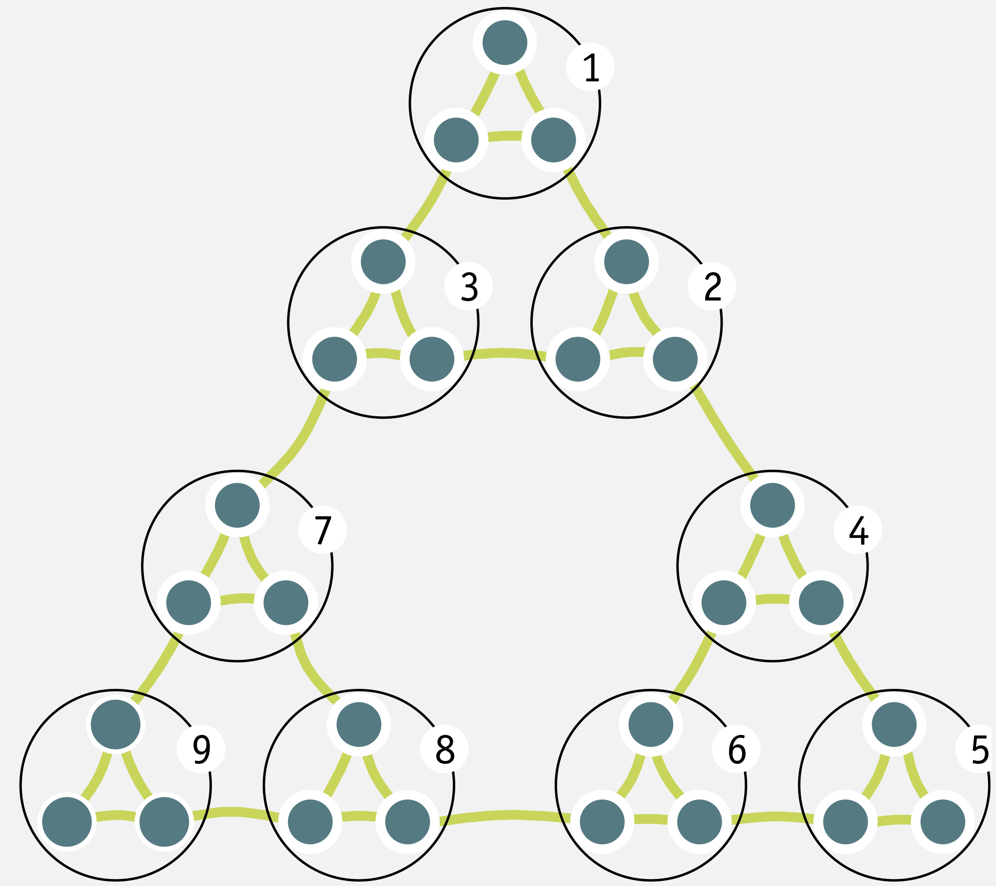
$$L(\mathsf{M}) = \underbrace{q_{\curvearrowleft} H(\mathcal{Q})}_{0.12 \text{ bits}} + \underbrace{\begin{cases} p^1_{\circlearrowright} H(\mathcal{P}^1) \\ p^2_{\circlearrowright} H(\mathcal{P}^2) \\ p^3_{\circlearrowright} H(\mathcal{P}^3) \end{cases}}_{3.56 \text{ bits}} = 3.68 \text{ bits.}$$

Two-level partitions with the map equation

A



B



$$L(\mathsf{M}) = \underbrace{q_{\curvearrowleft} H(\mathcal{Q})}_{0.97\ \text{bits}} + \underbrace{\begin{cases} p^1_{\circlearrowright} H(\mathcal{P}^1) \\ p^2_{\circlearrowright} H(\mathcal{P}^2) \\ p^3_{\circlearrowright} H(\mathcal{P}^3) \\ p^4_{\circlearrowright} H(\mathcal{P}^4) \\ p^5_{\circlearrowright} H(\mathcal{P}^5) \\ p^6_{\circlearrowright} H(\mathcal{P}^6) \\ p^7_{\circlearrowright} H(\mathcal{P}^7) \\ p^8_{\circlearrowright} H(\mathcal{P}^8) \\ p^9_{\circlearrowright} H(\mathcal{P}^9) \end{cases}}_{2.60\ \text{bits}} = 3.57\ \text{bits}.$$

$$L(\mathsf{M}) = \underbrace{q_{\curvearrowleft} H(\mathcal{Q})}_{0.12\ \text{bits}} + \underbrace{\begin{cases} p^1_{\circlearrowright} H(\mathcal{P}^1) \\ p^2_{\circlearrowright} H(\mathcal{P}^2) \\ p^3_{\circlearrowright} H(\mathcal{P}^3) \end{cases}}_{3.56\ \text{bits}} = 3.68\ \text{bits}.$$

Multilevel partitions

Into how many hierarchical levels is a given network organized? How many modules are present at each level? And which nodes are members of which modules?

Maximal compression of flow with constraints:

1. Modular code structure
2. ~~No more than two levels~~
3. Each node can only belong to one module

Multilevel partitions with the map equation

Into how many hierarchical levels is a given network organized? How many modules are present at each level? And which nodes are members of which modules?

Maximal compression of flow with constraints:

1. Modular code structure
2. ~~No more than two levels~~
3. Each node can only belong to one module

Summary of network flows with the map equation

Q: What are communities in the map equation framework?

A: Communities with long flow persistence times inferred with the minimum description length principle.

Coding theory

The minimum description length principle

Claude Shannon, David Huffmann, Jorma Rissanen, ...

Network flows

The map equation

Carl Bergstrom, Daniel Edler, Alcides Viamontes Esquivel, Renaud Lambiotte, Martin Rosvall, ...

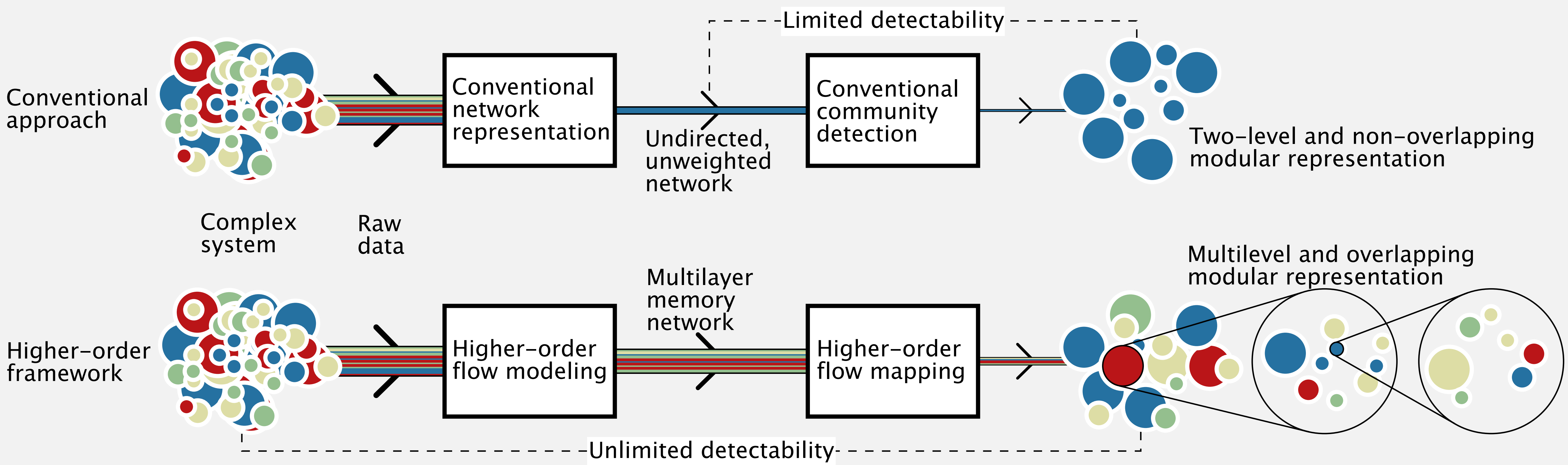
Higher-order flows

Sparse Markov models

Andrea Lancichinetti, Renaud Lambiotte, Manlio de Domenico, Alex Arenas, Christian Persson, Ludvig Bohlin, Daniel Edler, Martin Rosvall, ...

Networks with memory

describe where flows move to depending on where they come from


Limited detectability
Conventional approach
Conventional network representation
Undirected, unweighted network
Conventional community detection
Two-level and non-overlapping modular representation
Complex system
Raw data
Multilayer memory network
Multilevel and overlapping modular representation
Higher-order framework
Higher-order flow modeling
Higher-order flow mapping
Unlimited detectability

Networks with memory

describe where flows move to depending on where they come from

From pathways to networks with and without memory

A **Assemble data**

Itinerary	Number of tickets
New York ➡ Chicago ➡ New York	49,632
New York ➡ Chicago ➡ San Francisco	1,031
San Francisco ➡ New York ➡ Chicago ➡ San Francisco	120
Atlanta ➡ Chicago ➡ Atlanta	17,207
Jacksonville ➡ Atlanta ➡ Chicago ➡ Atlanta ➡ Jacksonville	418
⋮	⋮

From pathways to networks with and without memory

A **Assemble data**

Itinerary	Number of tickets
New York → Chicago → New York	49,632
New York → Chicago → San Francisco	1,031
San Francisco → New York → Chicago → San Francisco	120
Atlanta → Chicago → Atlanta	17,207
Jacksonville → Atlanta → Chicago → Atlanta → Jacksonville	418
⋮	⋮

Extract network

B

Bigrams	
New York → Chicago	174,085
Chicago → New York	172,830
Chicago → San Francisco	95,977
San Francisco → New York	99,140
Chicago → Atlanta	72,569
Atlanta → Chicago	72,467
⋮	⋮

C

Trigrams	
New York → Chicago → New York	50,105
New York → Chicago → San Francisco	3,319
San Francisco → New York → Chicago	211
Atlanta → Chicago → Atlanta	24,919
⋮	⋮

From pathways to networks with and without memory

Extract network

B

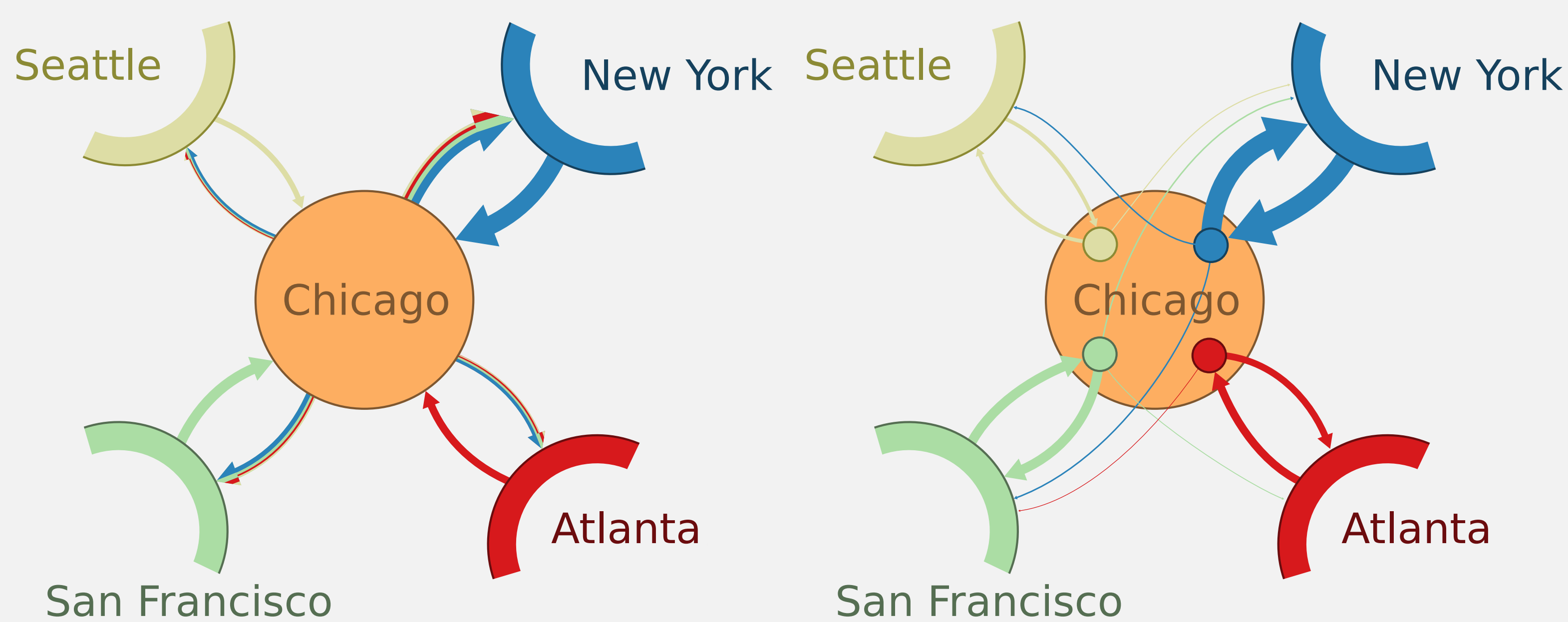
Bigrams	
New York → Chicago	174,085
Chicago → New York	172,830
Chicago → San Francisco	95,977
San Francisco → New York	99,140
Chicago → Atlanta	72,569
Atlanta → Chicago	72,467
⋮ ⋮	⋮

C

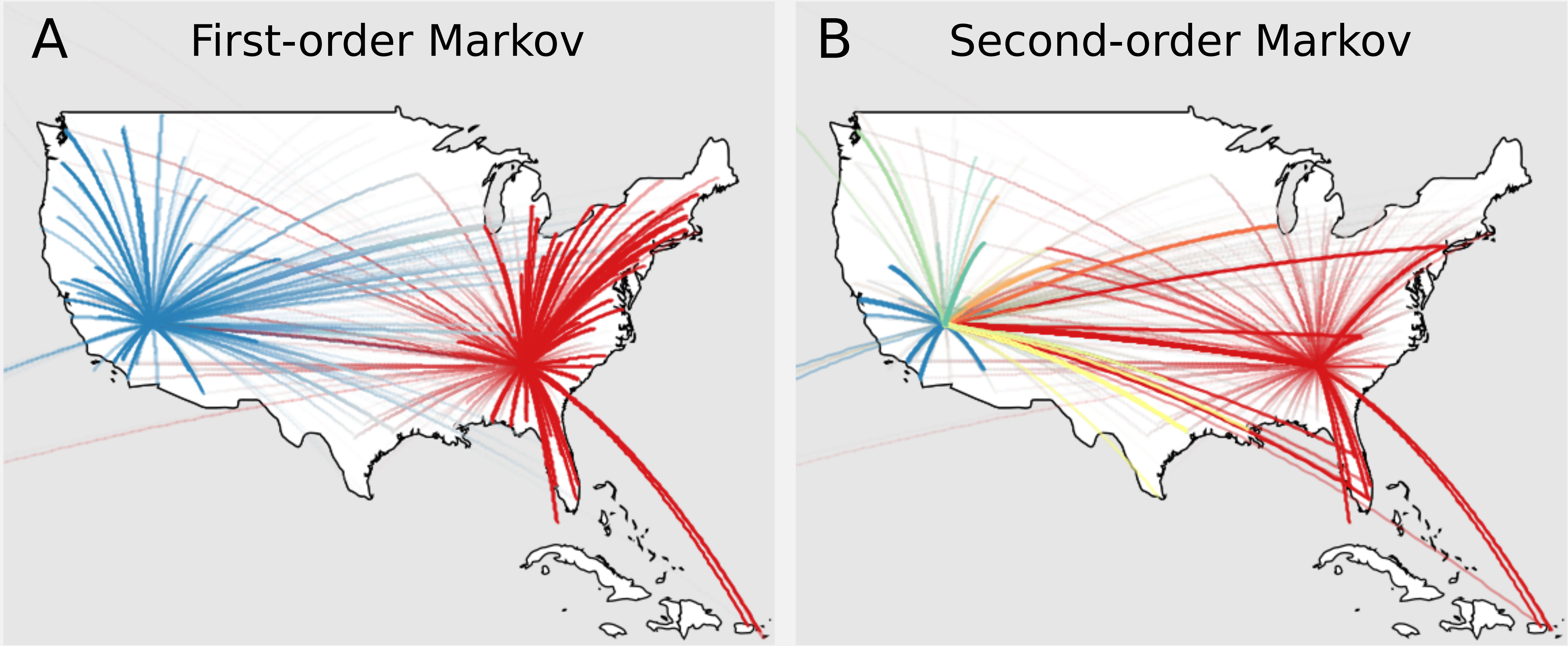
Trigrams	
New York → Chicago → New York	50,105
New York → Chicago → San Francisco	3,319
San Francisco → New York → Chicago	211
Atlanta → Chicago → Atlanta	24,919
⋮ ⋮ ⋮	⋮

D **Without memory**

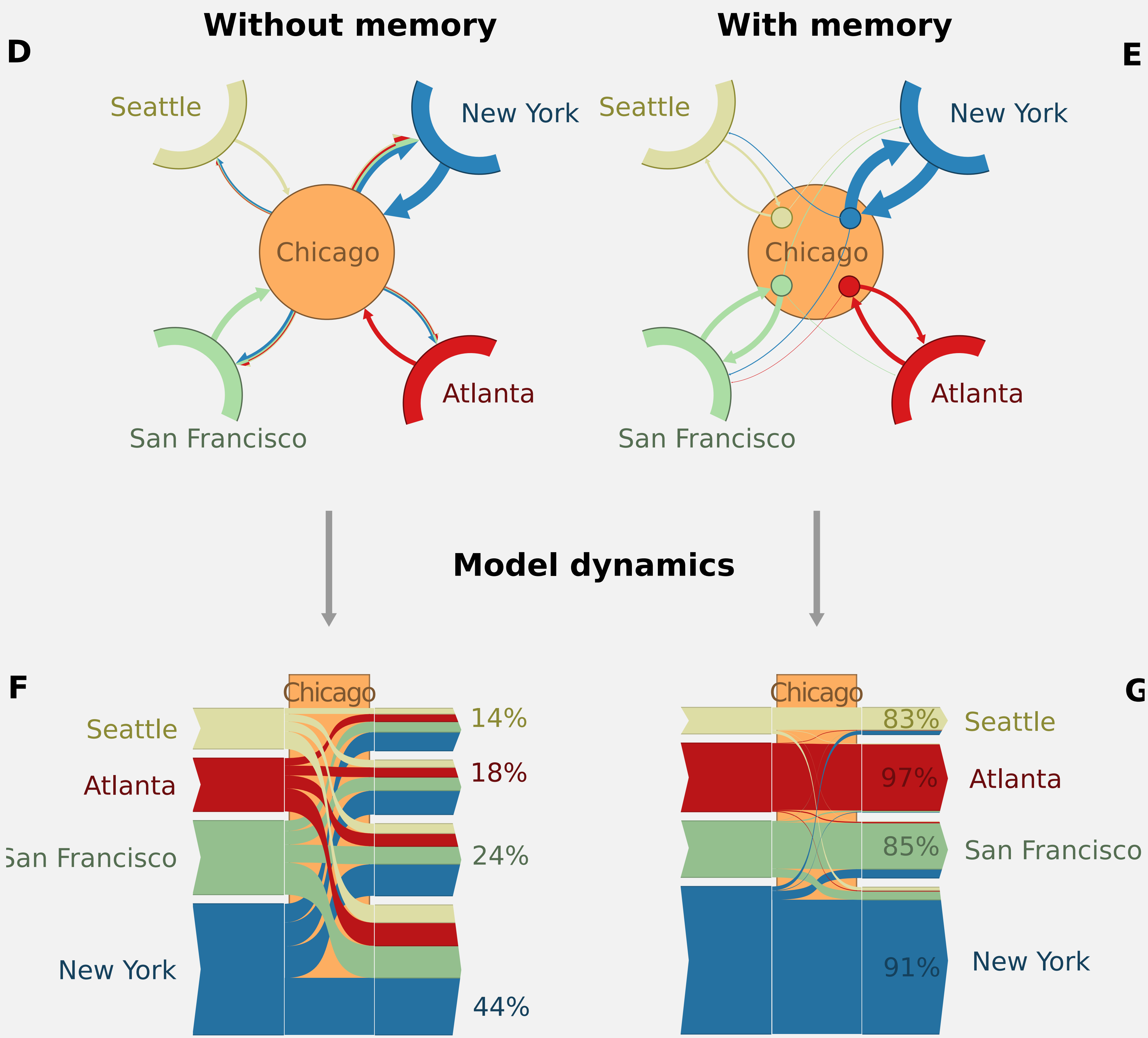
With memory **E**



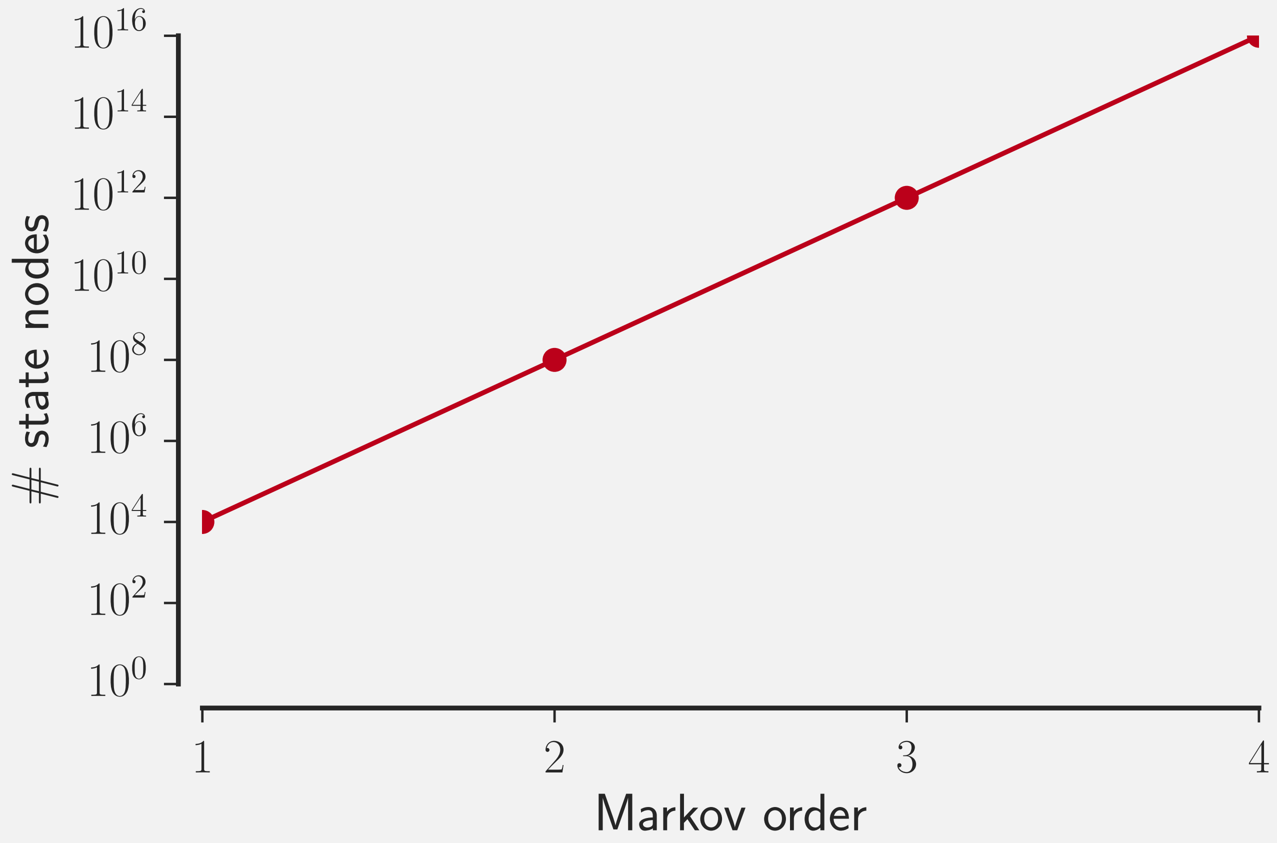
Memory affects clustering of air traffic

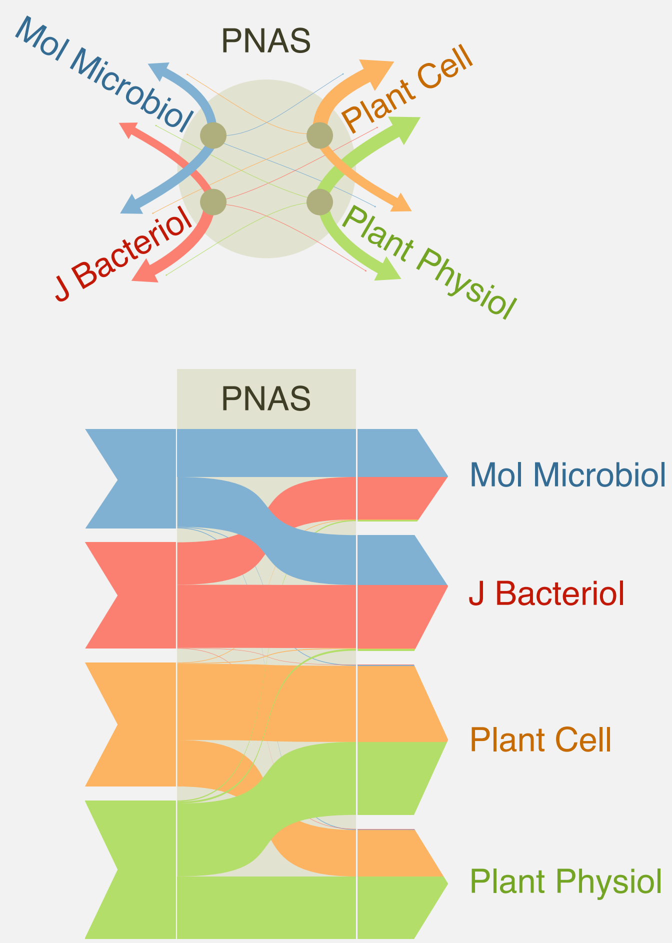


From pathways to networks with and without memory



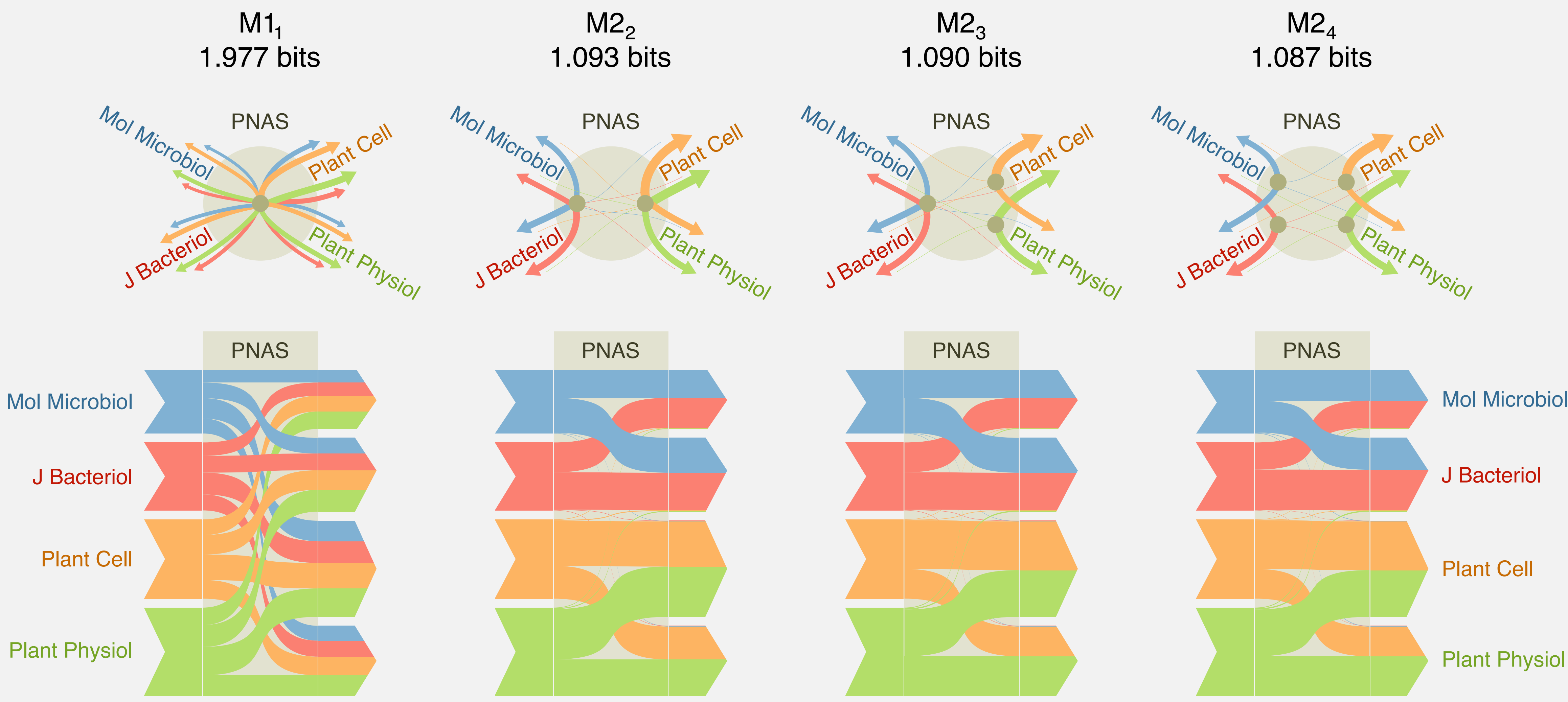
Challenge: How do we cope with scale and efficiently avoid under- and over-fitting?





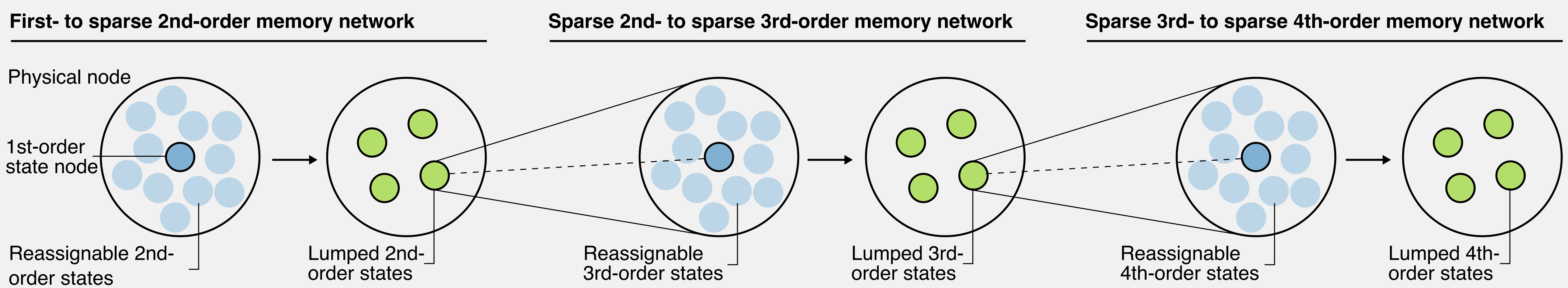
Challenge: How do we cope with scale and efficiently avoid under- and over-fitting?

Solution: Sparse Markov chain models



CHALLENGE: How do we cope with scale and efficiently avoid under- and over-fitting?

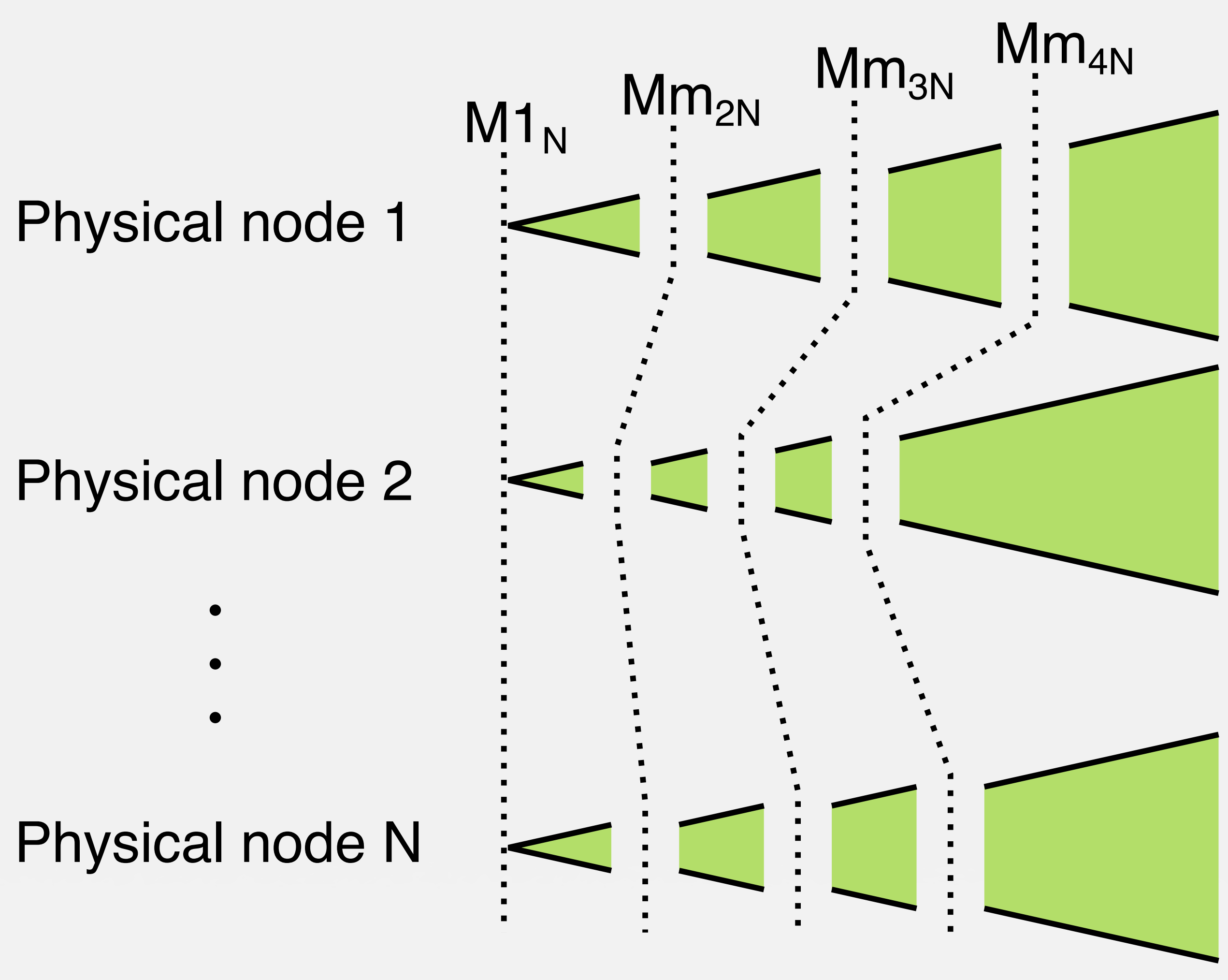
SOLUTION: Sparse Markov chain models



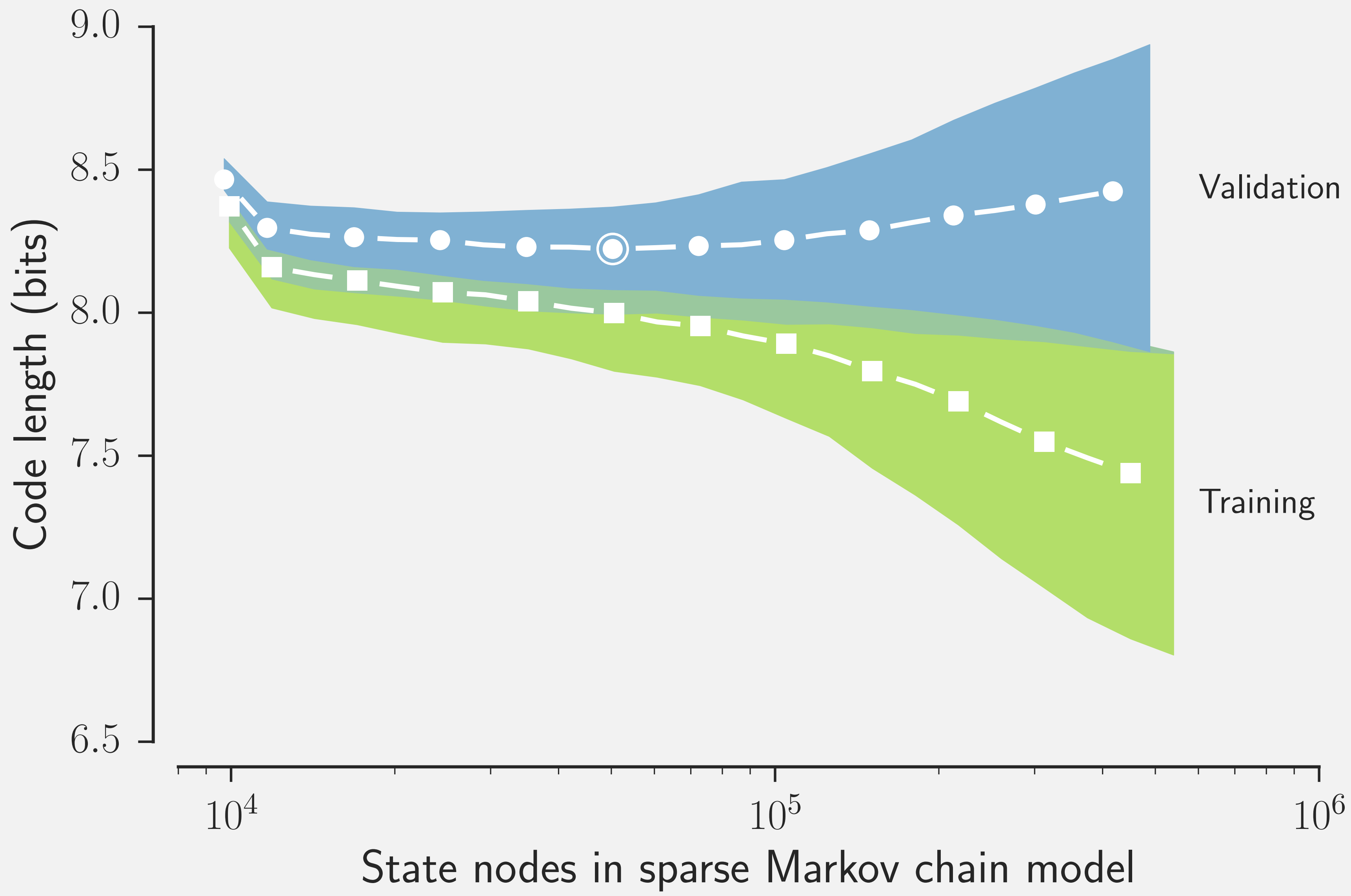
CHALLENGE: How do we cope with scale and efficiently avoid under- and over-fitting?

SOLUTION: Sparse Markov chain models

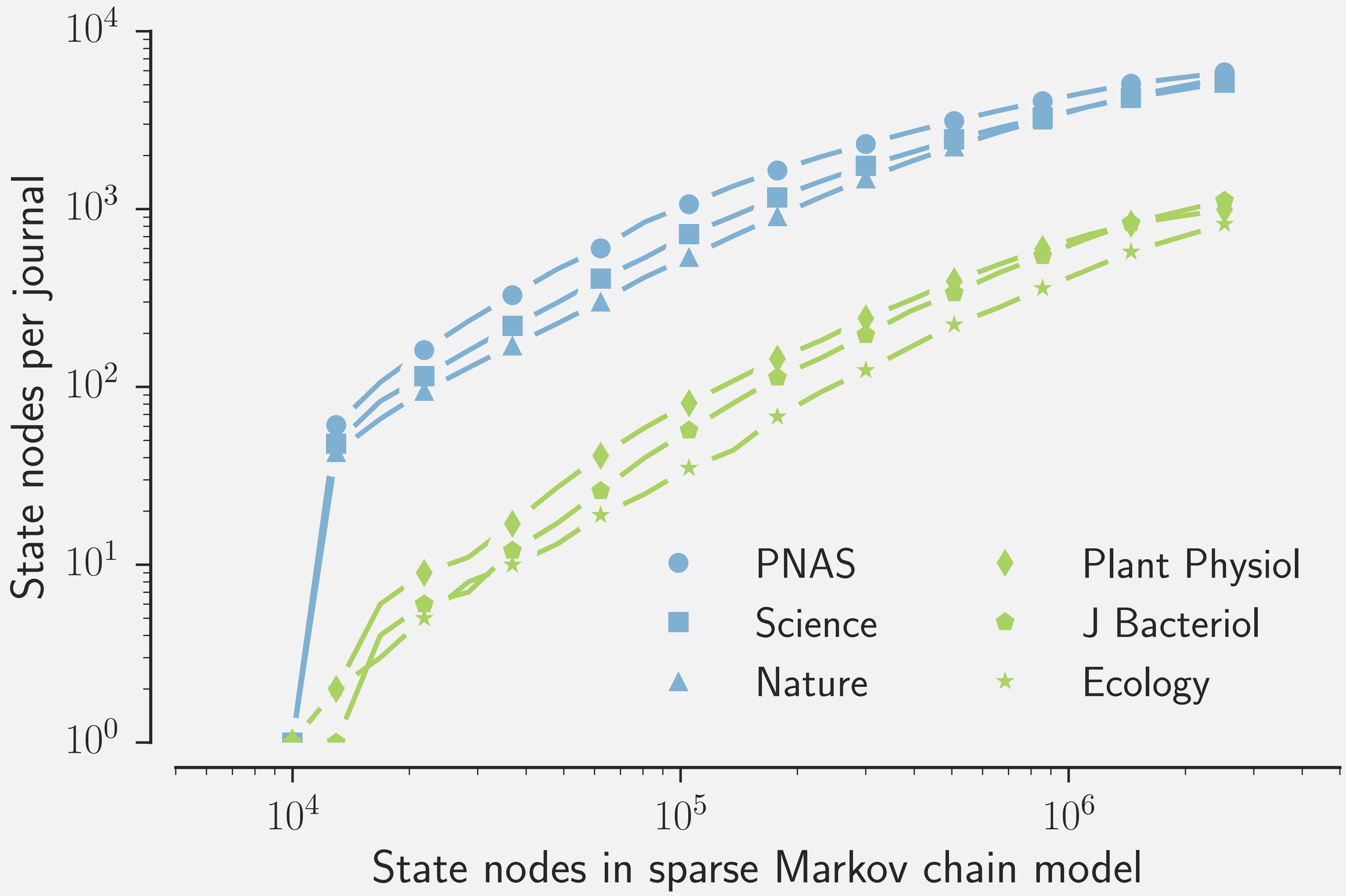
Expansion phase

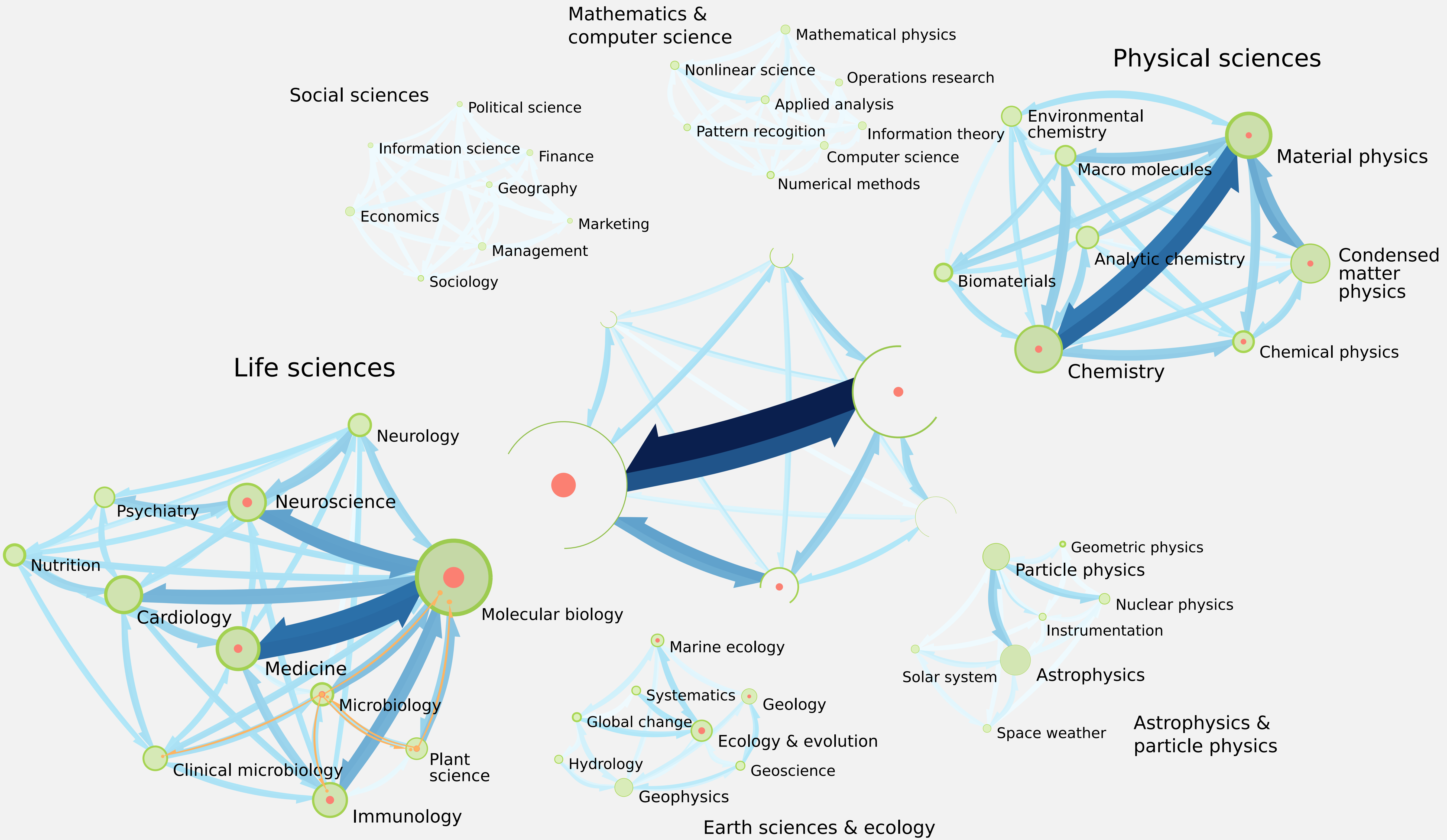


Ten-fold cross-validation to identify the best sparse Markov chain model for mapping



The sparse Markov chain models are efficient and allocate more state nodes to multidisciplinary journals





Mathematics &
computer science
Mathematical physics
Nonlinear science
Operations research
Applied analysis
Pattern recognition
Information theory
Computer science
Numerical methods
Social sciences
Political science
Information science
Finance
Geography
Economics
Marketing
Management
Sociology
Physical sciences
Environmental chemistry
Material physics
Macro molecules
Analytic chemistry
Condensed matter physics
Biomaterials
Chemical physics
Chemistry
Life sciences
Neurology
Psychiatry
Neuroscience
Nutrition
Cardiology
Molecular biology
Medicine
Microbiology
Plant science
Clinical microbiology
Immunology
Geometric physics
Particle physics
Nuclear physics
Instrumentation
Solar system
Astrophysics
Space weather
Astrophysics &
particle physics
Marine ecology
Systematics
Geology
Global change
Ecology & evolution
Hydrology
Geoscience
Geophysics
Earth sciences & ecology

Summary of higher-order network flows

Q: How can we design effective maps of flows such that we can identify actual overlapping communities?

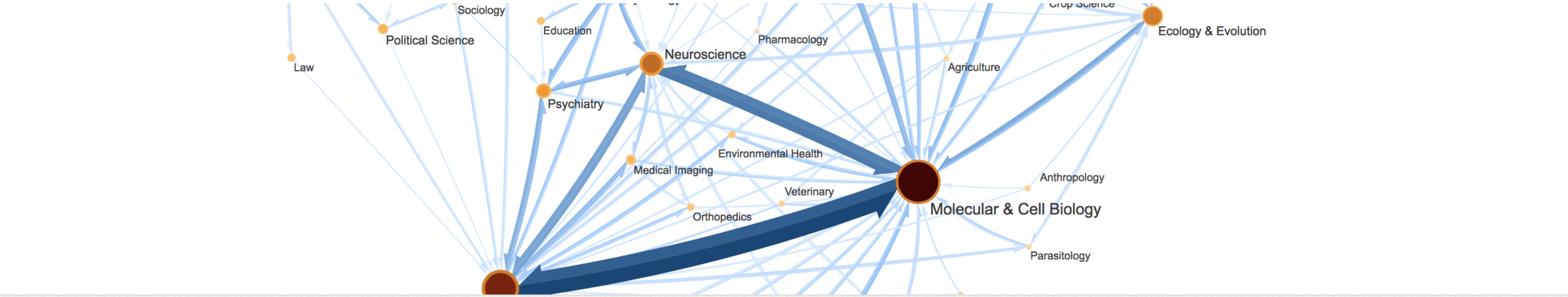
A: Higher-order network flows modeled with sparse Markov chains capture flows efficiently, and mapping them reveals overlapping communities with longer flow persistence time.

mapequation.org

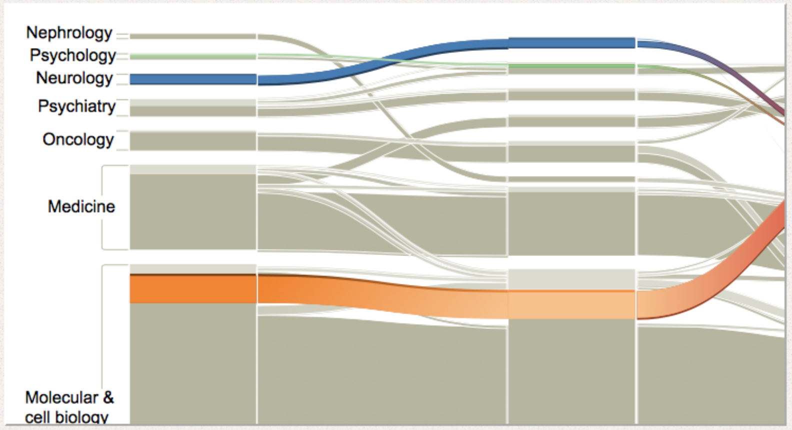
MapEquation

apps code publications about

Simplify and highlight important structures in complex networks



Apps »



Code »

```
using infomath::plogp;
for (unsigned int i = 0; i < numNodes; ++i)
{
enter_log_enter += plogp(m_moduleFlowData[i].enter
exit_log_exit += plogp(m_moduleFlowData[i].exitFlo
flow_log_flow += plogp(m_moduleFlowData[i].exitFlo
enterFlow += m_moduleFlowData[i].enterFlow;
}
enterFlow += exitNetworkFlow;
enterFlow_log_enterFlow = plogp(enterFlow);
```

Publications »

Maps of information flow reveal community structure in complex networks

Martin Rosvall and Carl T. Bergstrom

PNAS **105**, 1118 (2008). [arXiv:0707.0609]

To comprehend the multipartite organization of large-scale biological and social systems, we introduce a new information-theoretic approach to reveal community structure in

News

Nov 22, 2017 Research paper — Mapping intermittent communities — efficiently reveal intermittent communities in temporal networks
Sep 30, 2017 Research paper — Mapping higher-order network flows — in memory and multilayer networks with Infomap
April 11, 2017 Source code — Infomap on Windows — run Infomap in bash on ubuntu on Windows
July 4, 2016 Reserach paper — Maps of sparse Markov chains — efficiently reveal community structure in network flows with memory
December 5, 2015 Mapping tool — Infomap Bioregions — interactive mapping of biogeographical regions from species distributions
September 5, 2015 Source code — GossipMap — a distributed implementation of infomap by Seung-Hee Bae and Bill Howe
August 13, 2015 Interactive storyboard — Multilevel network sampling — infer network modes from multiple samples
August 13, 2015 Interactive storyboard — Higher-order Markov models — identify flows on memory and multilayer networks
July 23, 2015 Source code — Infomap — updates to memory and multilayer algorithms
April 1, 2015 Source code — Infomap with other languages — use Infomap as a library wity Python, iGraph, or R
March 6, 2015 Paper — Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems — Phys Rev X 5, 011027 (2015)