Predicting emergent sequences from network connectivity

Katie Morrison
University of Northern Colorado
School of Mathematical Sciences
katherine.morrison@unco.edu

joint work with
Caitlyn Parmelee, Samantha Moore, Carina Curto

Motivating Questions

1. How does the connectivity structure of a network shape emergent dynamics?

2. More specifically, how does connectivity shape sequential neural activity?

3. Can we predict the order of neural firing in an emergent sequence?

Outline of Talk

I. Combinatorial Threshold-Linear Network (CTLN) model
   - Introduction to model
   - Theoretical results
   - Full variety of dynamics exhibited by model

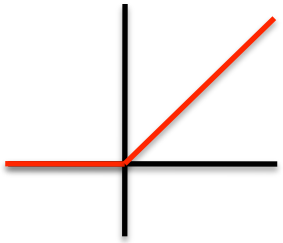
II. Predicting emergent sequences
   - Neuroscience motivation
   - Algorithm for sequence prediction in CTLNs
   - Taxonomy of small networks

I. Combinatorial Threshold-Linear Networks (CTLNs)

A toy model of recurrent networks:
Combinatorial Threshold-Linear Networks (CTLN)

Threshold-linear dynamics

$$\frac{dx_i}{dt} = -x_i + \left[\sum_{j=1}^{n} W_{ij}x_j + \theta\right]_+$$



Graph to network connectivity

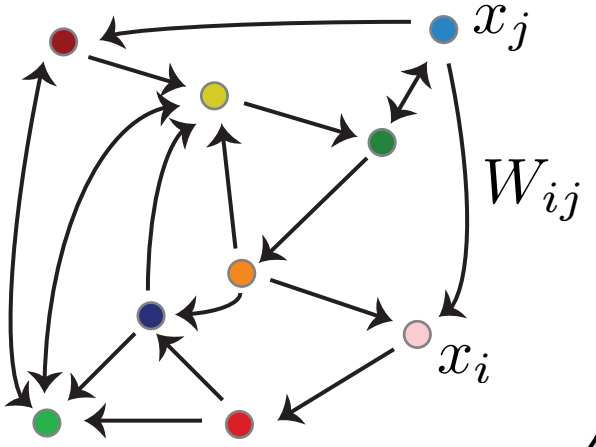
$$W_{ij} = \begin{cases} 0 & \text{if } i = j \\ -1+\varepsilon & \text{if } i \leftarrow j \text{ in } G \\ -1-\delta & \text{if } i \not\leftarrow j \text{ in } G \end{cases}$$

Parameter constraints

$$\delta > 0 \qquad 0 < \varepsilon < \frac{\delta}{\delta+1}$$

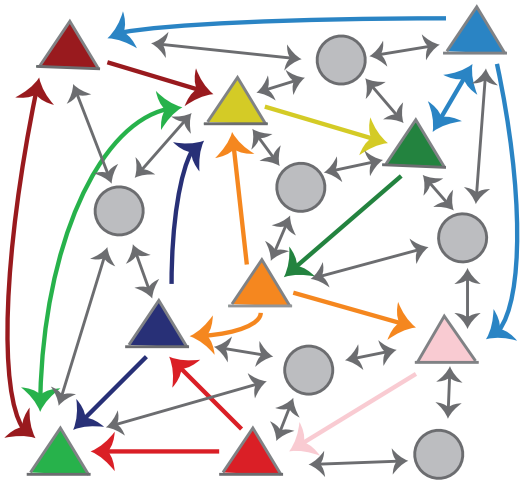
$$\theta > 0$$

Graph encodes pattern of weak vs. strong inhibition

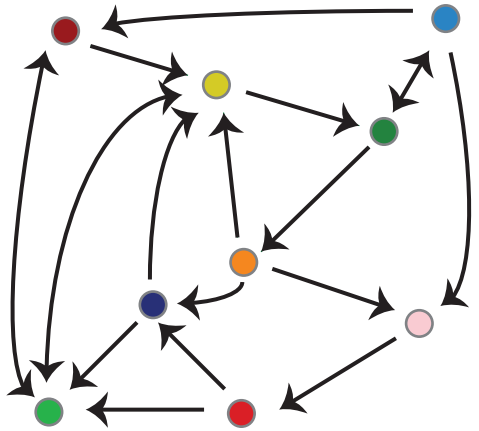


Pyramidal neurons in a sea of inhibition

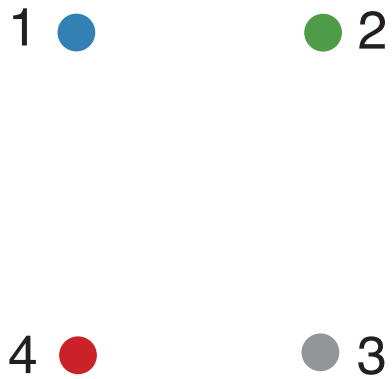
Network of excitatory and inhibitory cells



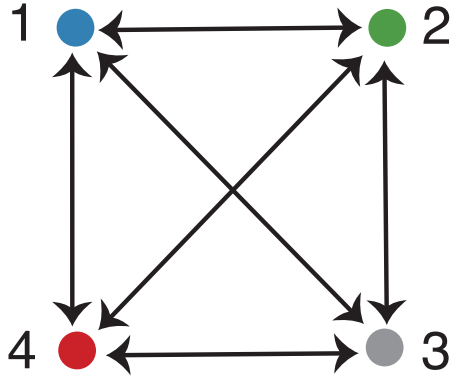
Graph of interactions among excitatory cells



The boring extremes



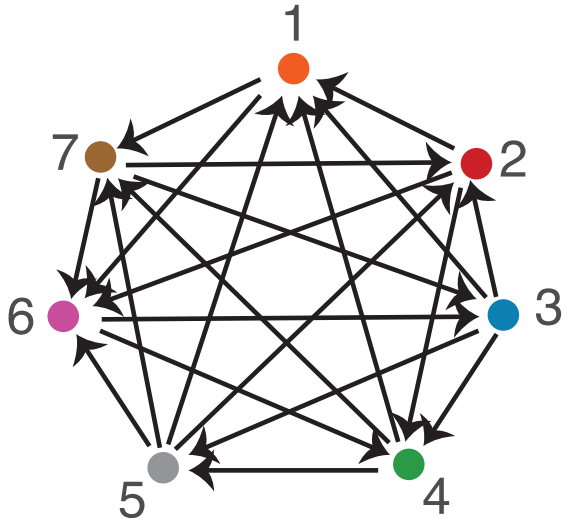
Winner-Take-All behavior
(n stable fixed points)

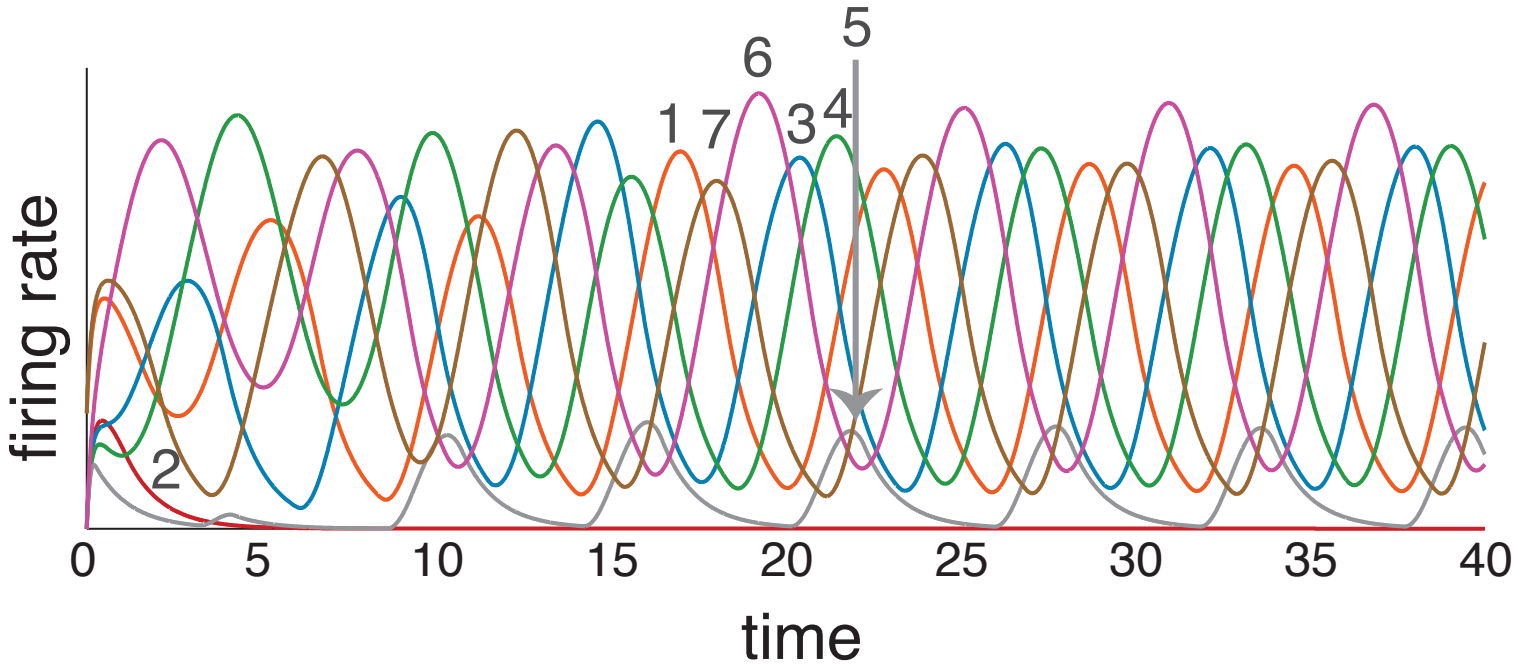


Synchronous behavior
(1 stable fixed point)

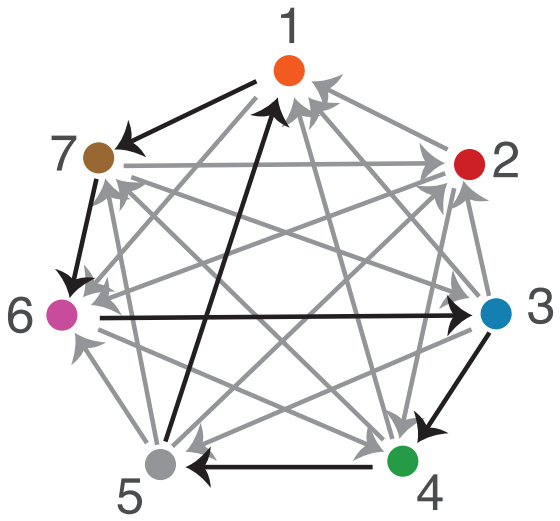
The dynamics get interesting when you have
a mix of strong and weak inhibition...

Sequential activity from a mix of strong and weak inhibition

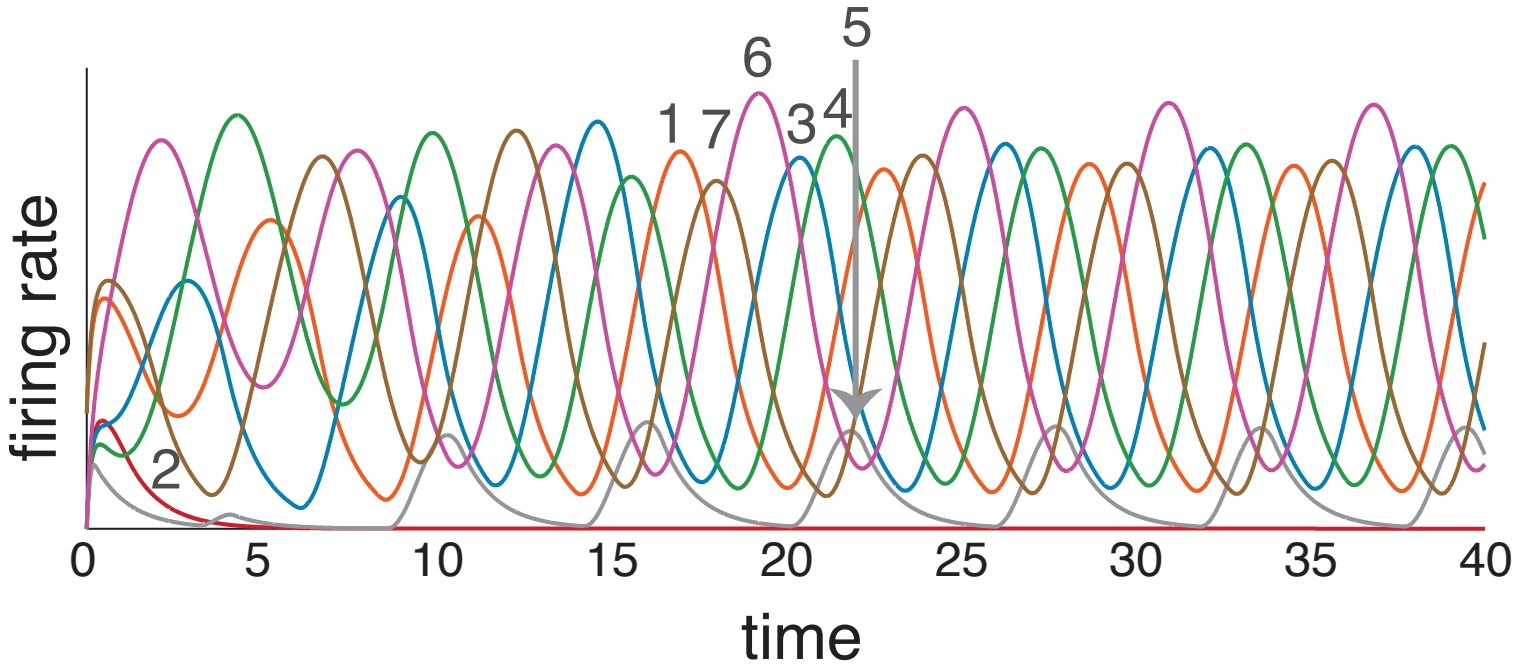




Sequential activity from a mix of strong and weak inhibition

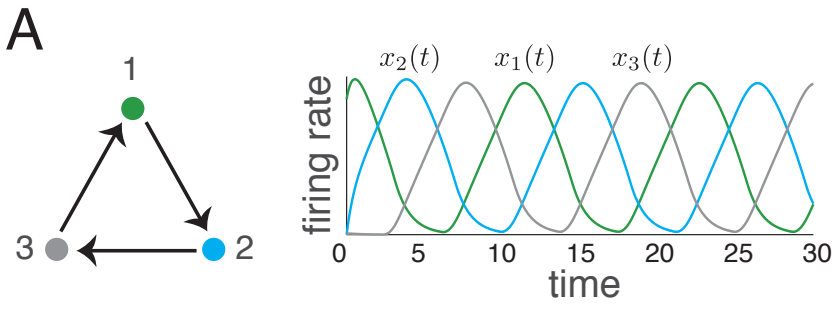


Can we predict this emergent sequence
176345
from connectivity?



CTLN Theorems

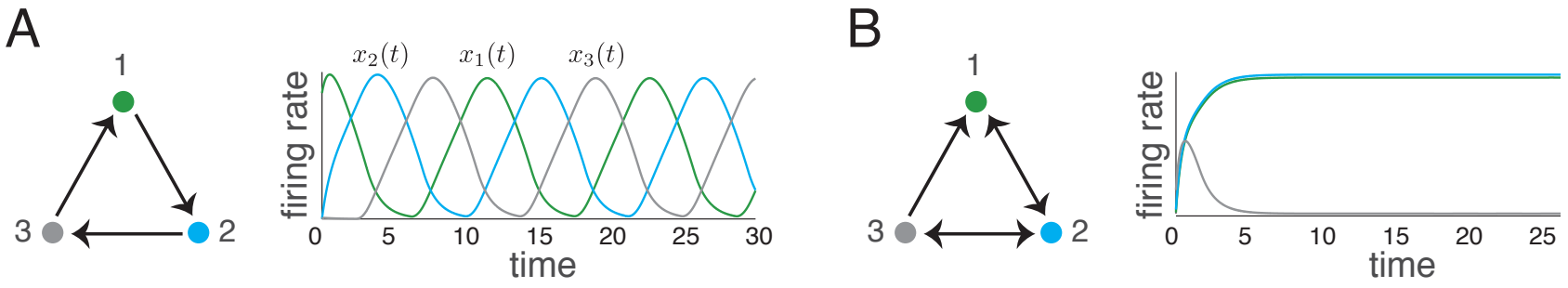
Theorem 1. If G is an oriented graph with no sinks, then the network has no stable fixed points.
Furthermore, the network has bounded activity and thus dynamics are guaranteed to be oscillatory or chaotic.



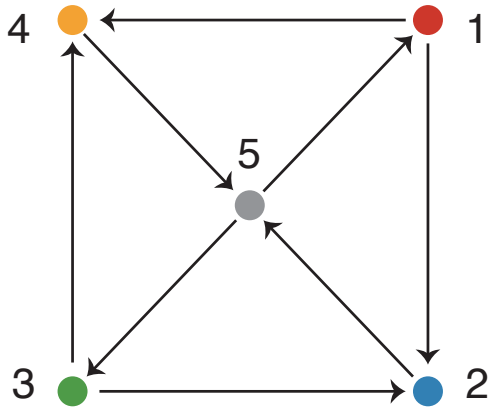
CTLN Theorems

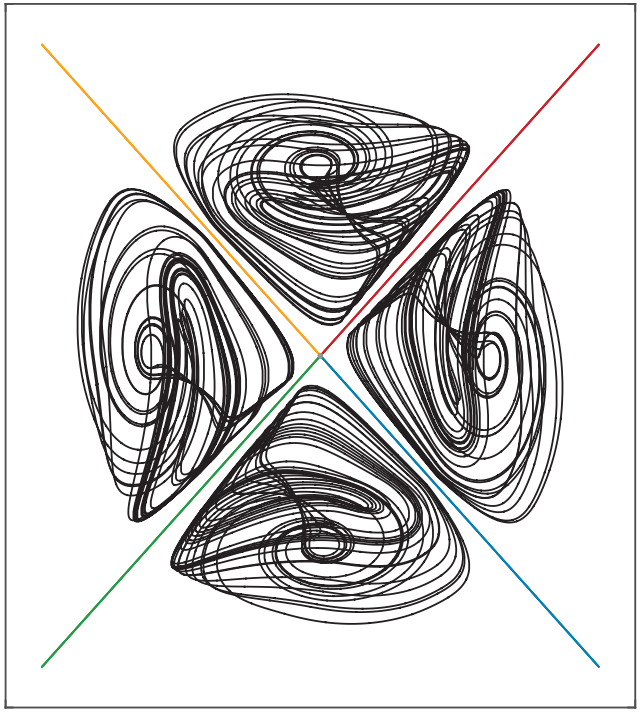
Theorem 1. If G is an oriented graph with no sinks, then the network has no stable fixed points.
Furthermore, the network has bounded activity and thus dynamics are guaranteed to be oscillatory or chaotic.

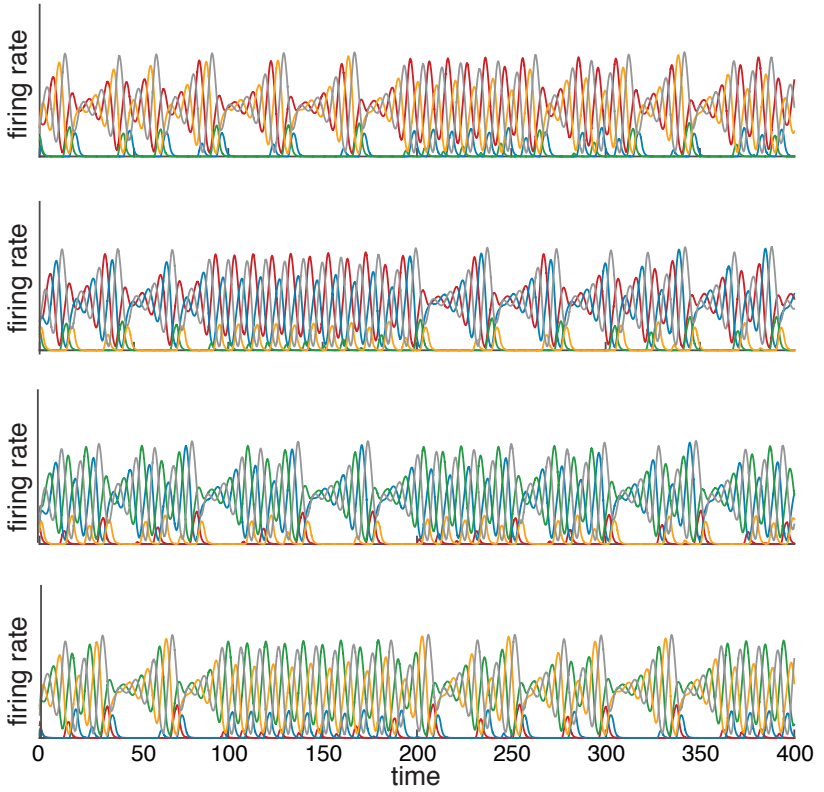
Theorem 2. For any G, a clique is the support of a stable fixed point if and only if it is a target-free clique.



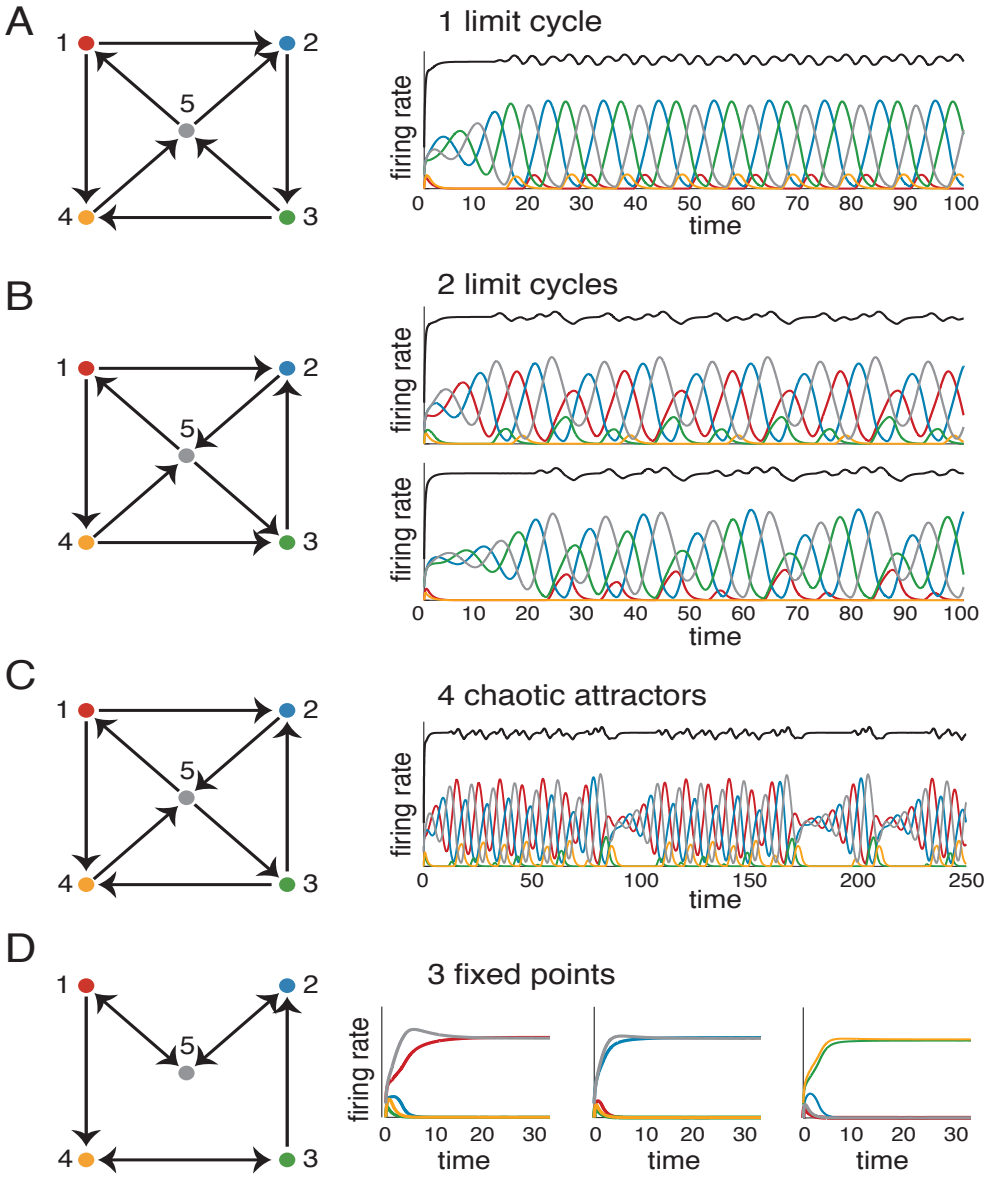
Multiple chaotic attractors in a single network







Local features are not sufficient to predict attractor type

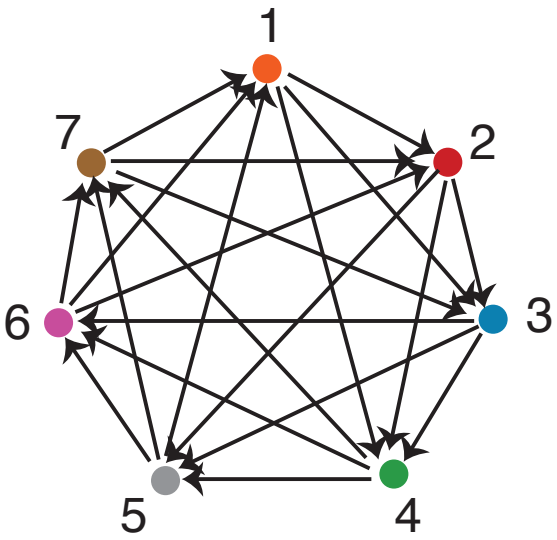


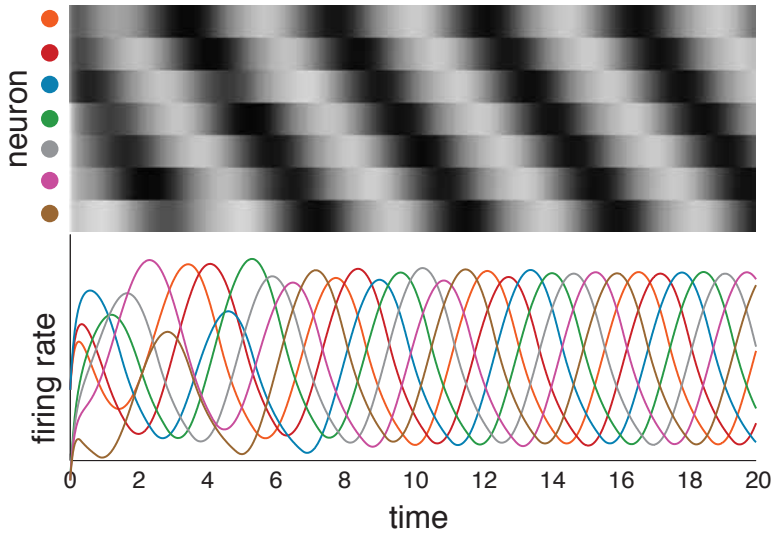
Four graphs with identical in/out-degree sequence:

(1,2), (1,2), (2,1), (2,1), (2,2)

but dramatically different attractors

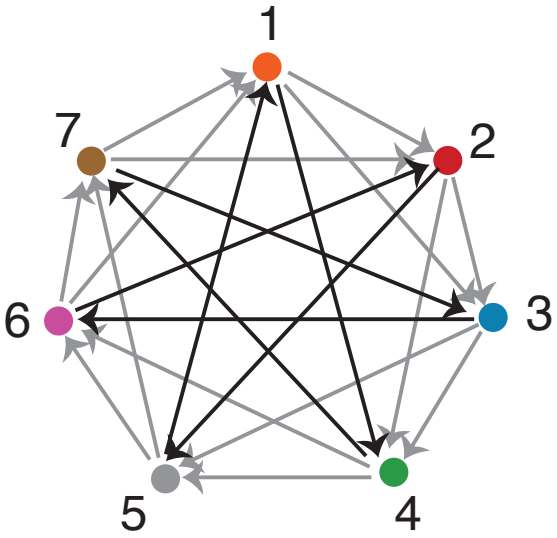
Coexistence of attractors of different types

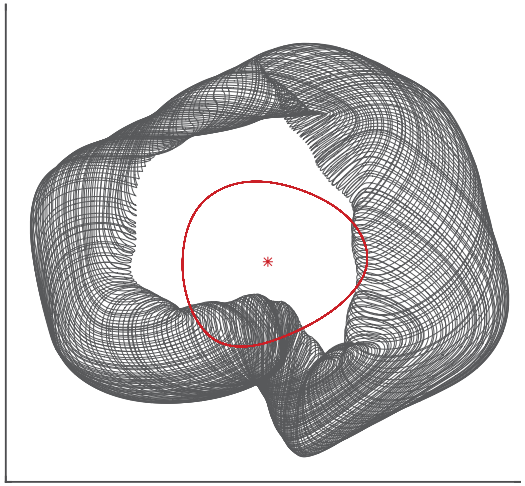


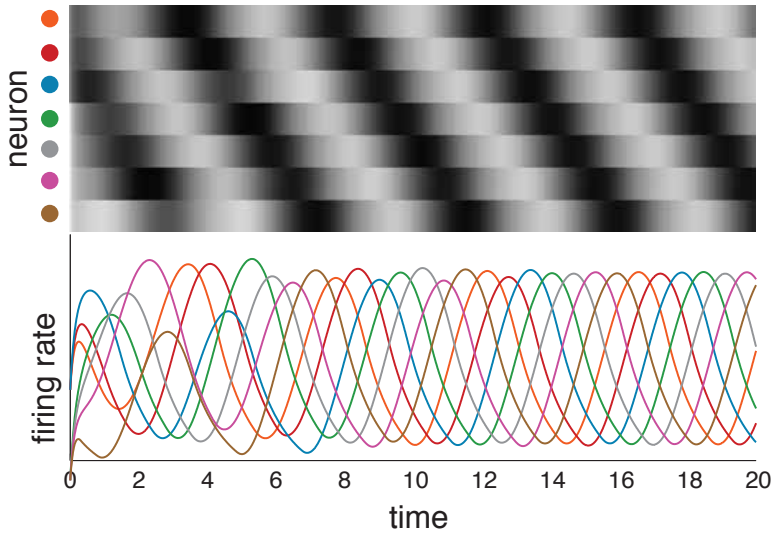


Sequence 1234567

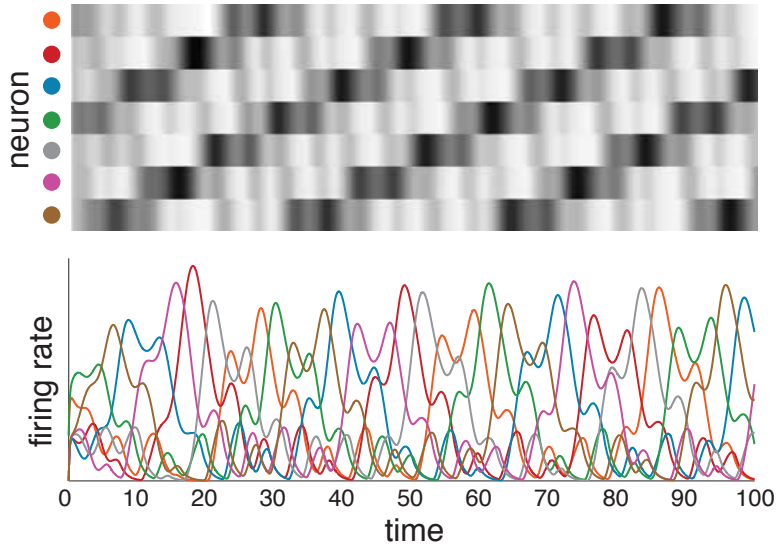
Coexistence of attractors of different types







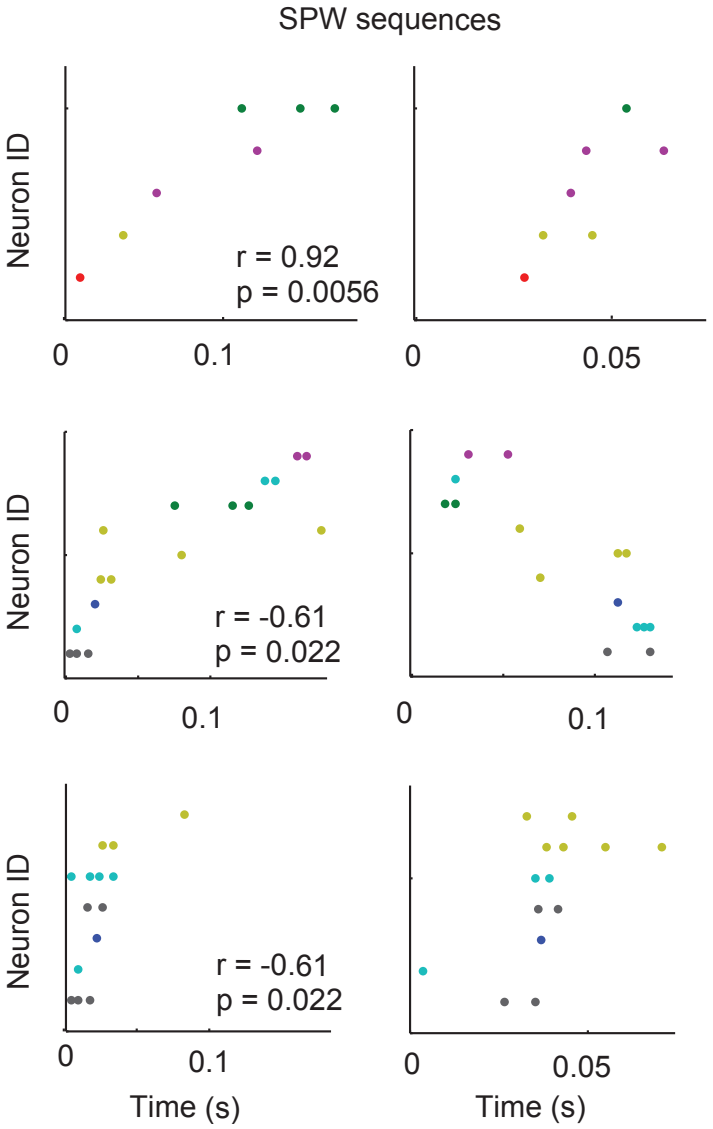
Sequence 1234567

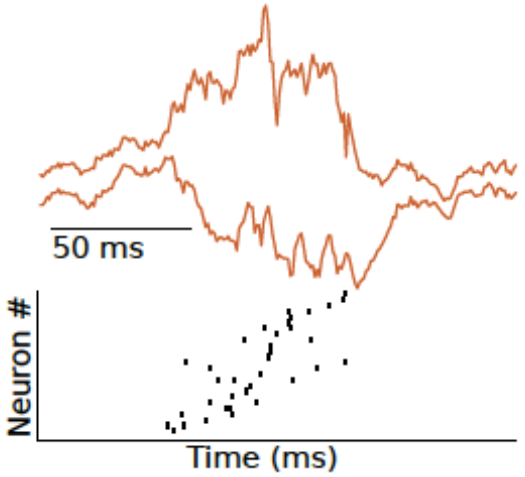


Sequence 1473625

II. Predicting emergent sequences

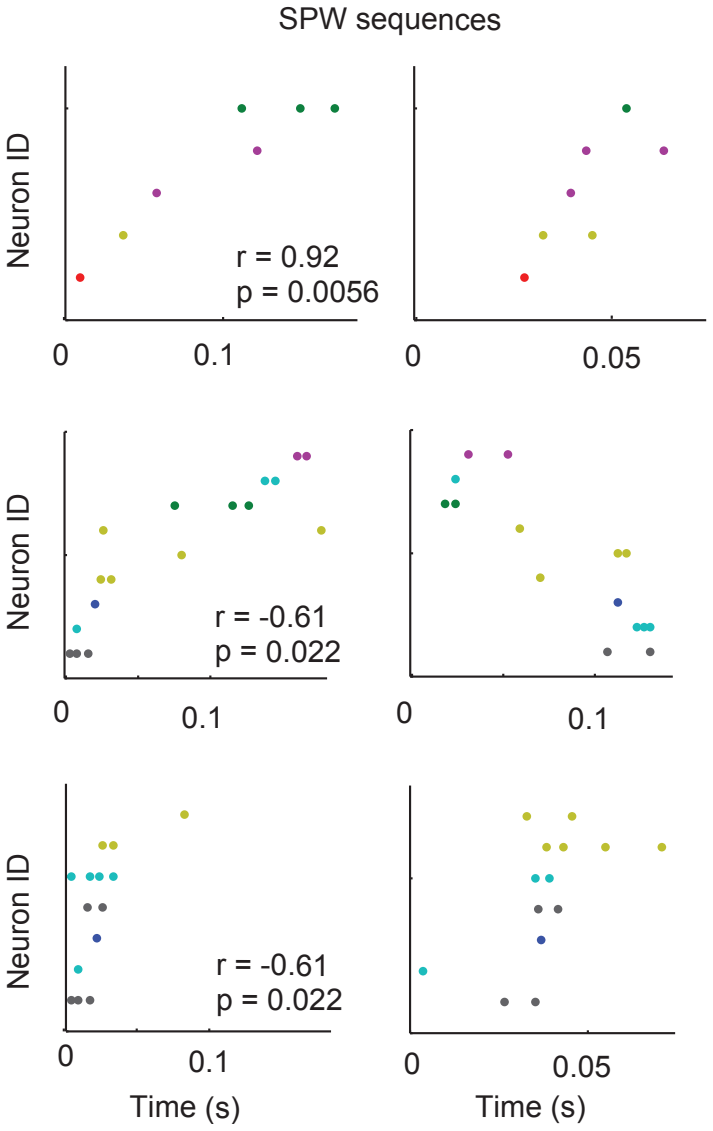
Ripple sequences in hippocampus



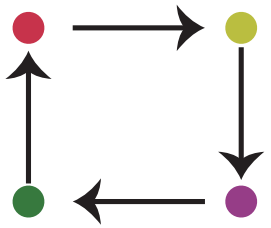


Data from Eva Pastalkova's Lab at Janelia Research Campus

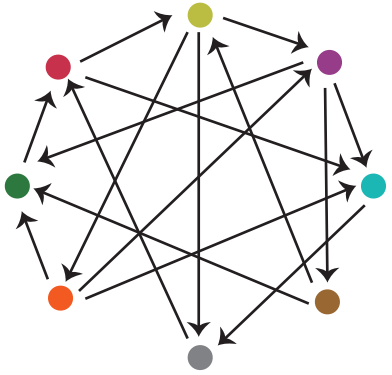
Ripple sequences in hippocampus



Is the connectivity pattern "engineered" to produce these sequences?



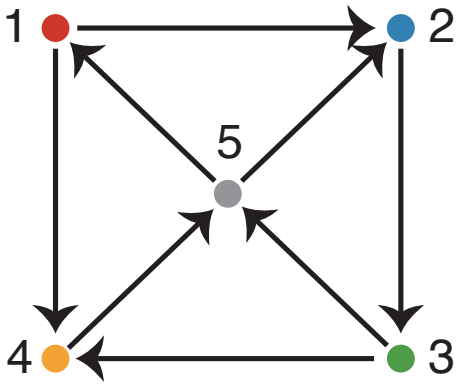
Or are they emergent from more complex connectivity patterns?



Data from Eva Pastalkova's Lab at Janelia Research Campus

Emergent sequences

G

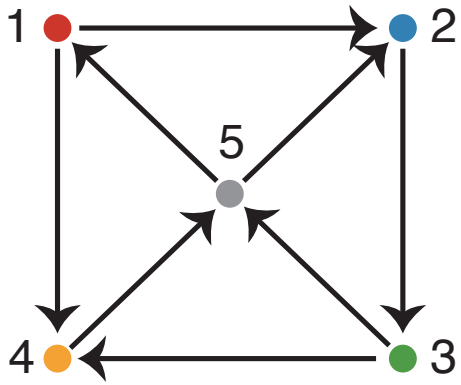


Cycles in G:

145
235
1235
2345
12345

Emergent sequences

G



Cycles in G:

145

235

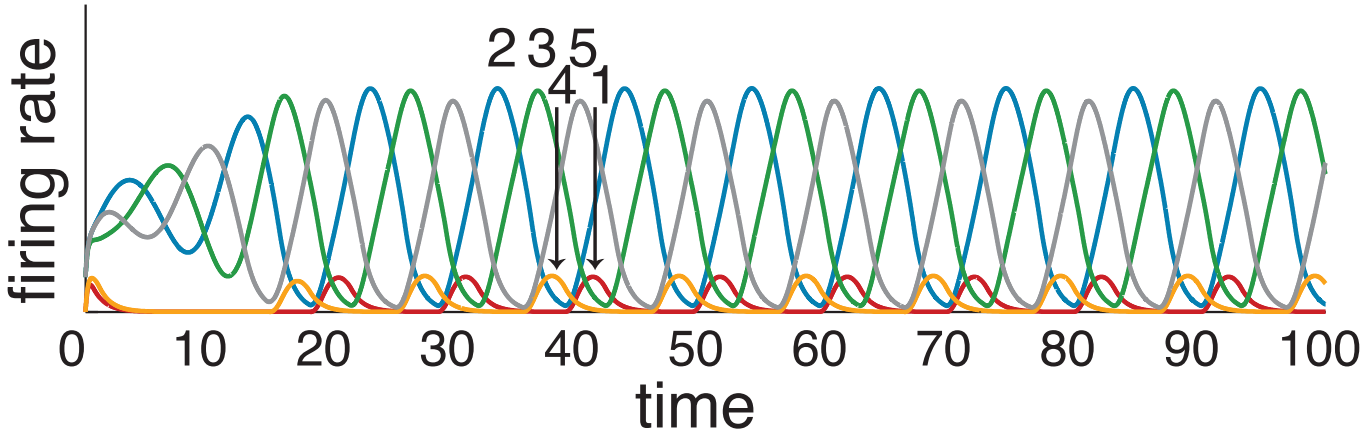
1235

2345

12345

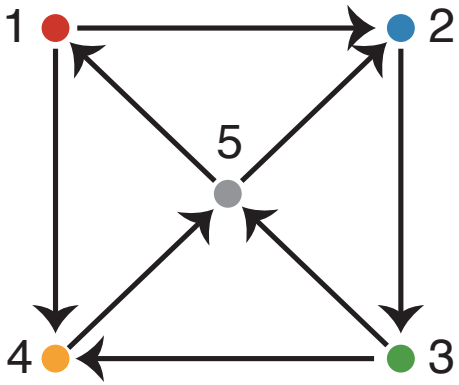
Only 1 attractor observed:

Limit cycle with sequence 23_45_1



Emergent sequences

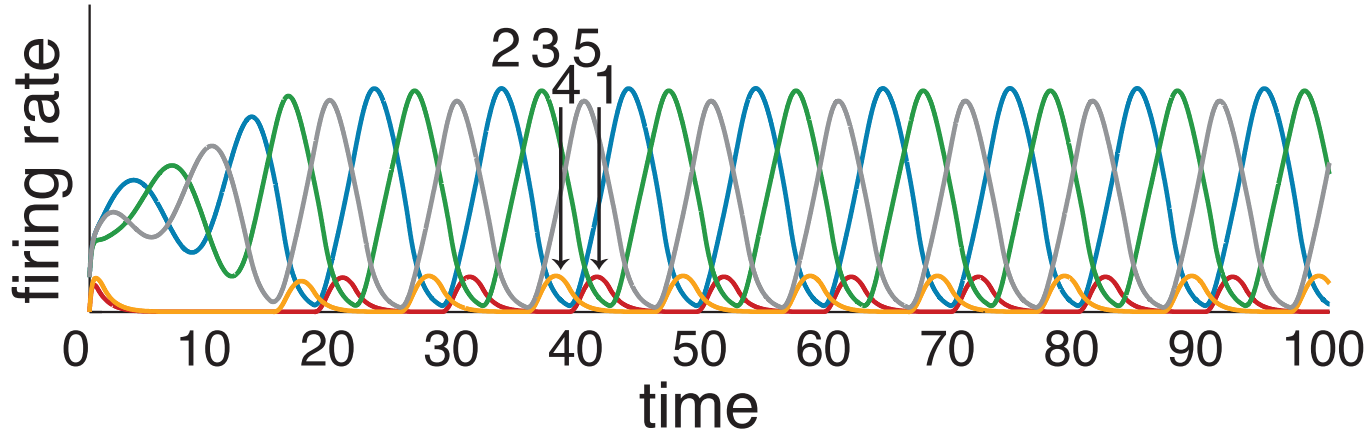
G



Cycles in G:

145
235
1235
2345
12345

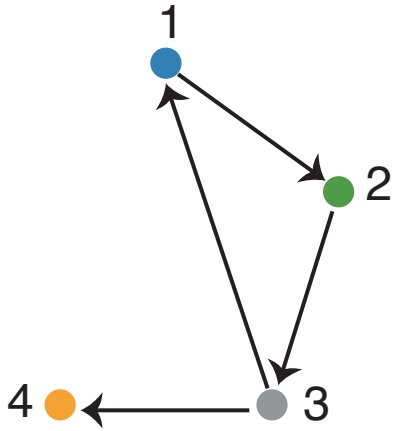
Only 1 attractor observed:
Limit cycle with sequence 23_45_1



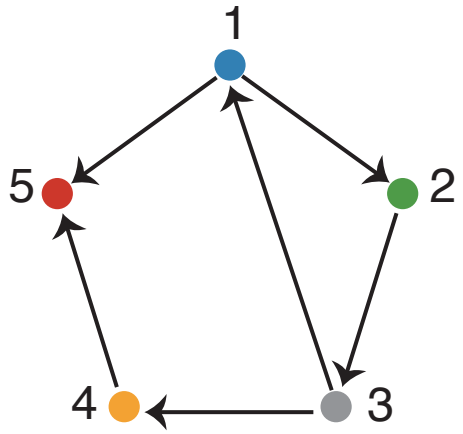
Can we predict the sequence from the graph?

Sequence prediction algorithm

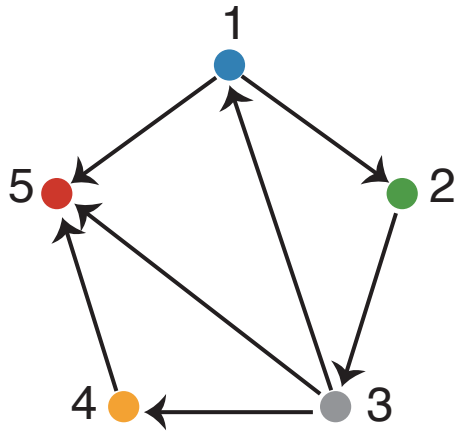
Terminology



node 4 is a sink

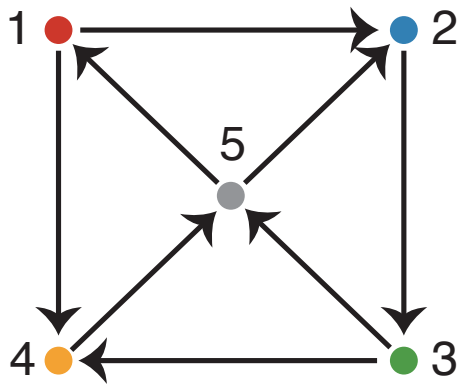


123 is a core 3-cycle: no node receives 2 or more edges from it



123 is a non-core 3-cycle: node 5 receives 2 edges from 123

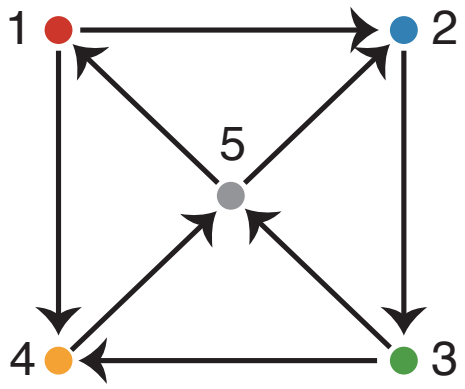
Sequence prediction algorithm



Cycles in G:

145	non-core 3-cycle
235	core 3-cycle
1235	contains core 3-cycle
2345	contains core 3-cycle
12345	contains core 3-cycle

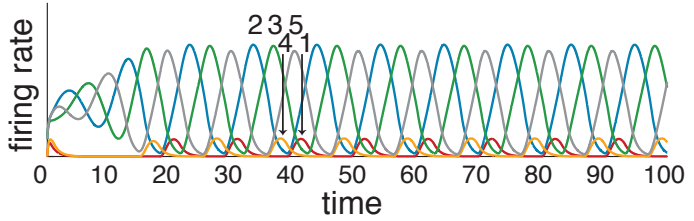
Sequence prediction algorithm



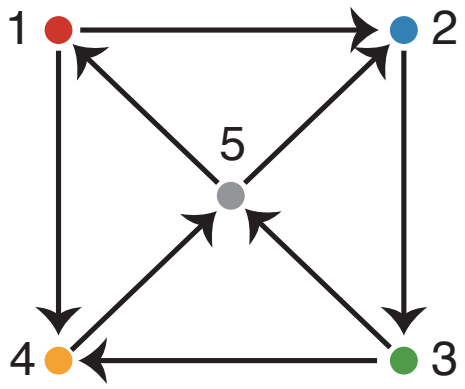
Cycles in G:

145	non-core 3-cycle
235	core 3-cycle
1235	contains core 3-cycle
2345	contains core 3-cycle
12345	contains core 3-cycle

Only 1 attractor observed:
Limit cycle with sequence 23_45_1



Sequence prediction algorithm

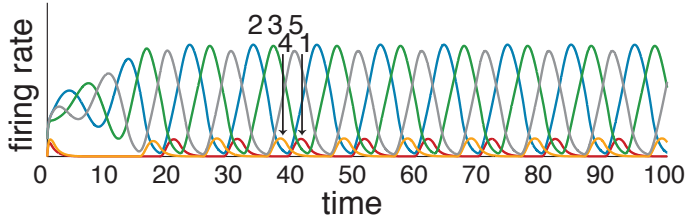


Cycles in G:

145	non-core 3-cycle
235	core 3-cycle
1235	contains core 3-cycle
2345	contains core 3-cycle
12345	contains core 3-cycle

Only 1 attractor observed:
Limit cycle with sequence 23_45_1

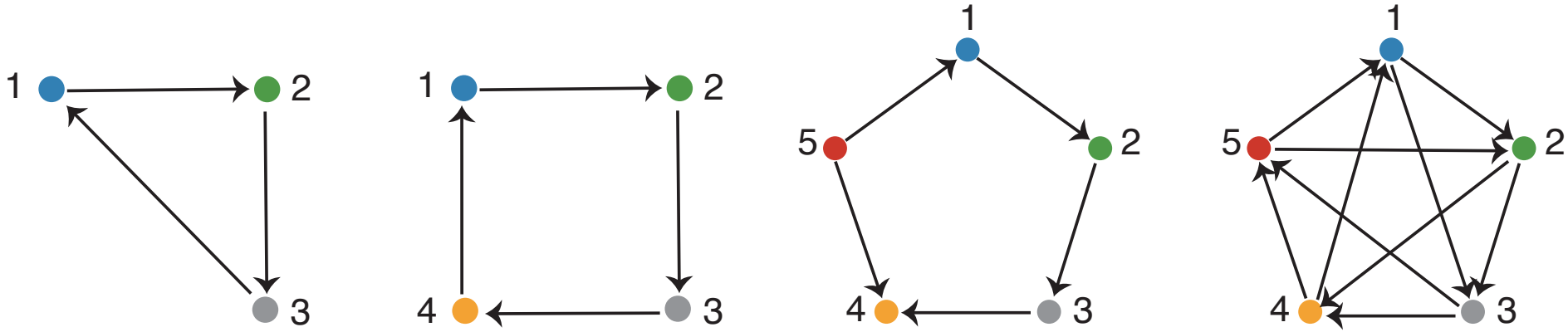
corresponds to core cycle 235



Sequence prediction algorithm

1. Remove the node with lowest in-degree that does **not** create a sink or a non-core 3-cycle. When there are multiple choices track all possible options.

Repeat Step 1 until no more nodes can be removed. This yields the core cycle(s). The core cycle gives the high-firing neurons in the sequence.



possible core cycles

Sequence prediction algorithm

1. Remove the node with lowest in-degree that does **not** create a sink or a non-core 3-cycle. When there are multiple choices track all possible options.

   Repeat Step 1 until no more nodes can be removed. This yields the core cycle(s). The core cycle gives the high-firing neurons in the sequence.

2. For each choice of removed nodes, re-insert nodes into the sequence in reverse order of their removal. Only insert if they receive input from the core cycle and insert them after input node.

   Output: list of predicted sequences

Sequence prediction algorithm

1. Remove the node with lowest in-degree that does **not** create a sink or a non-core 3-cycle. When there are multiple choices track all possible options.

   Repeat Step 1 until no more nodes can be removed. This yields the core cycle(s). The core cycle gives the high-firing neurons in the sequence.

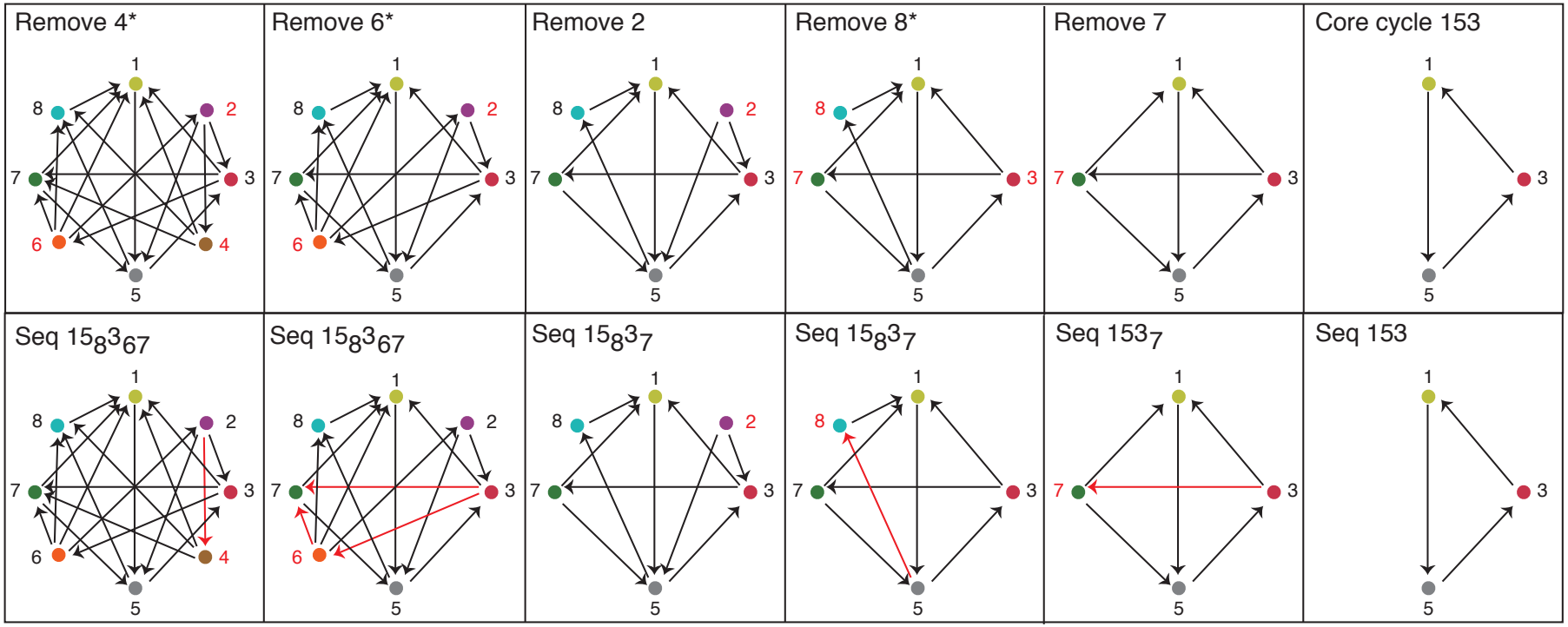
2. For each choice of removed nodes, re-insert nodes into the sequence in reverse order of their removal. Only insert if they receive input from the core cycle and insert them after input node.

   Output: list of predicted sequences

3. Examine list of sequences; if different choices produce two sequences with the same core cycle, these sequences merge and yield synchronous firing of the non-core neurons whose positions align.

Sequence prediction algorithm

Decompose graph →



Predicted Sequence: $15_8\,3_{67}$

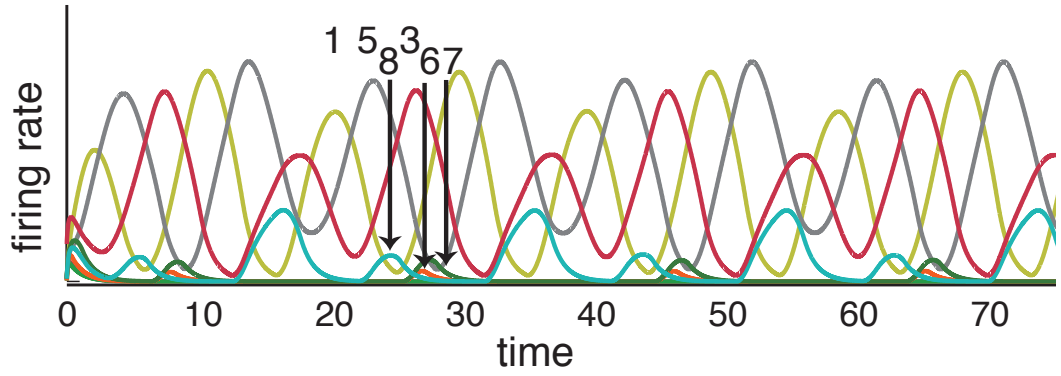
← Reconstruct sequence

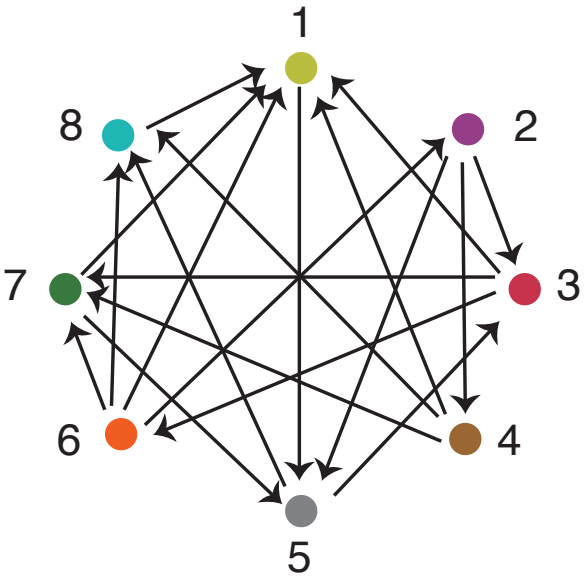
* indicates choice for nodes. The first choice between nodes 4 & 6 does <u>not</u> affect the sequence. The second choice between 3, 7 & 8 would have resulted in another predicted sequence $15_3 8$.

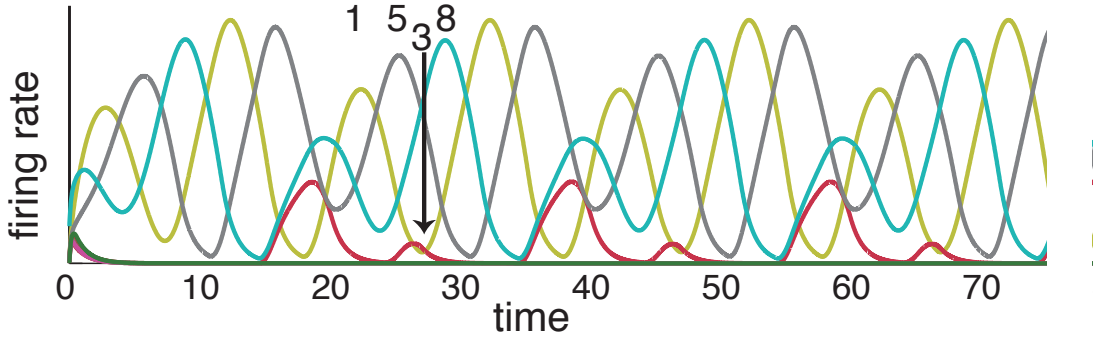
Predicted sequences: $15_8 3_{67}$ and $15_3 8$

Sequence prediction algorithm

Predicted sequences: $15_8 3_{67}$ and $15_3 8$







Only two attractors observed –
sequences precisely match those predicted by algorithm!

Algorithm prediction results

Complete analysis of attractors of the 160 non-isomorphic graphs on $n \leq 5$ neurons that are oriented with no sinks.

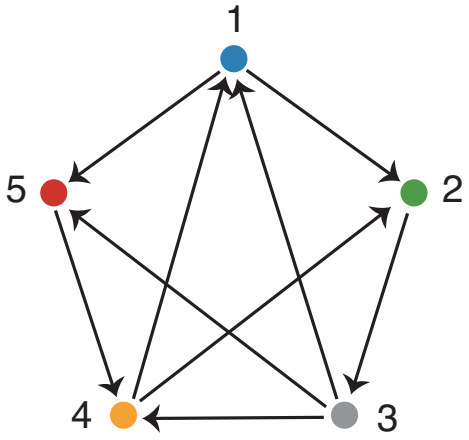
Algorithm correctly predicts the set of sequences of attractors for 156 of the 160 graphs.

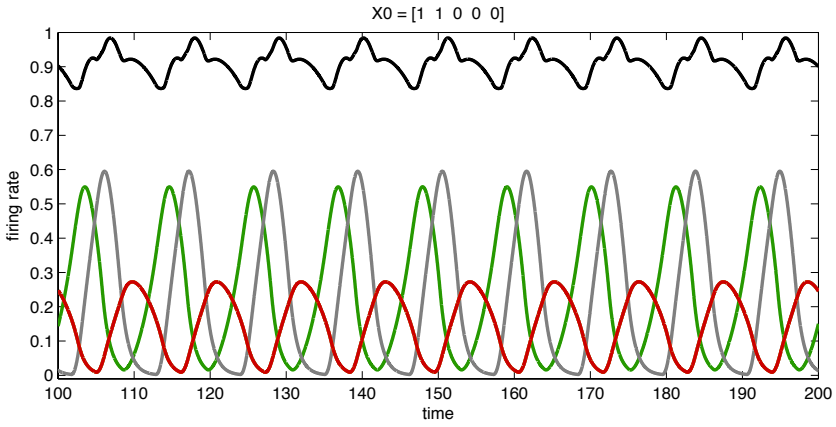
Algorithm prediction results

Complete analysis of attractors of the 160 non-isomorphic graphs on $n \leq 5$ neurons that are oriented with no sinks.

Algorithm correctly predicts the set of sequences of attractors for 156 of the 160 graphs.

Algorithm failure: unexpected synchrony – neurons 1,4,5 fire synchronously





Algorithm output:

$123 \rightarrow 1_5 23_4$

$234 \rightarrow 23_5 4_1$

$154 \rightarrow 1_2 54$

Algorithm prediction results

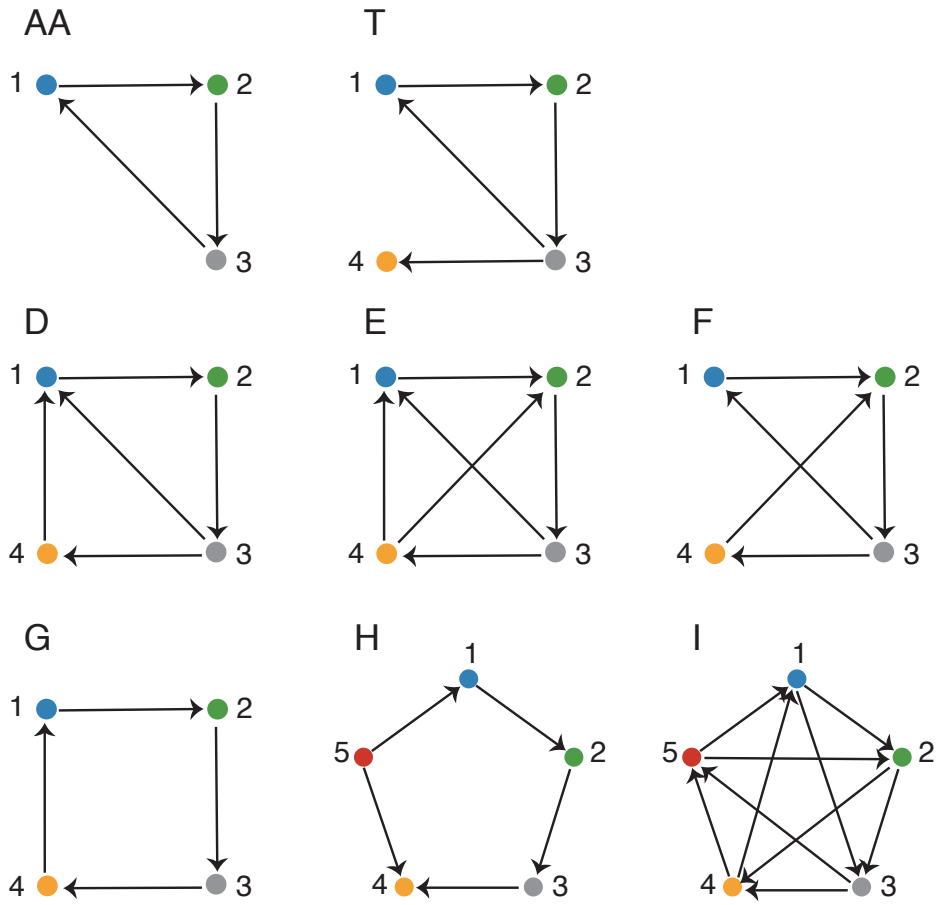
Complete analysis of attractors of the 160 non-isomorphic graphs on $n \leq 5$ neurons that are oriented with no sinks.

Algorithm correctly predicts the set of sequences of attractors for 156 of the 160 graphs.

Of the ~200 total attractors, there are only 19 distinctly different attractor types!

Taxonomy for oriented graphs on 5 neurons

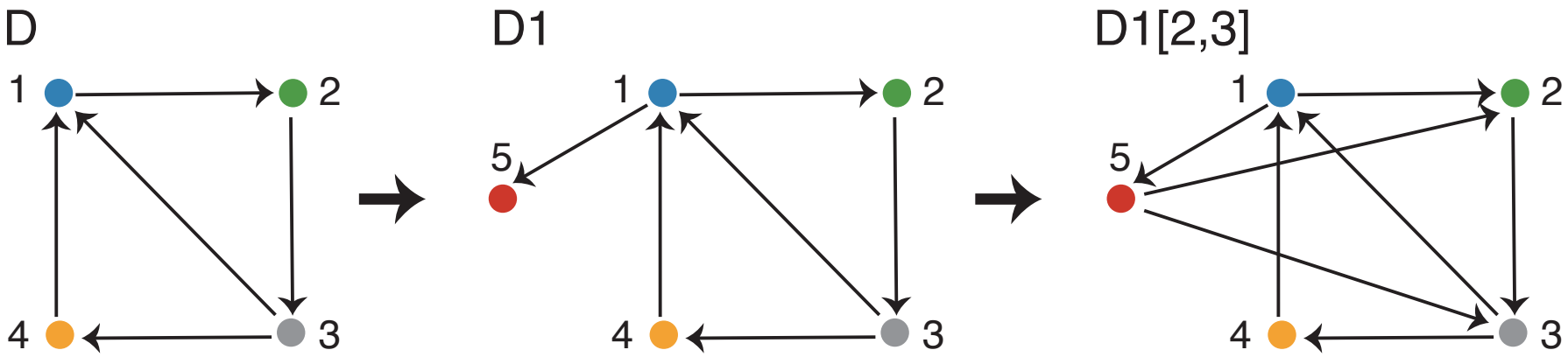
We found that the base graph on 4 neurons after 1st step of algorithm is a good predictor of attractor type.



Neuron numbering chosen to maximally align sequences in observed attractors

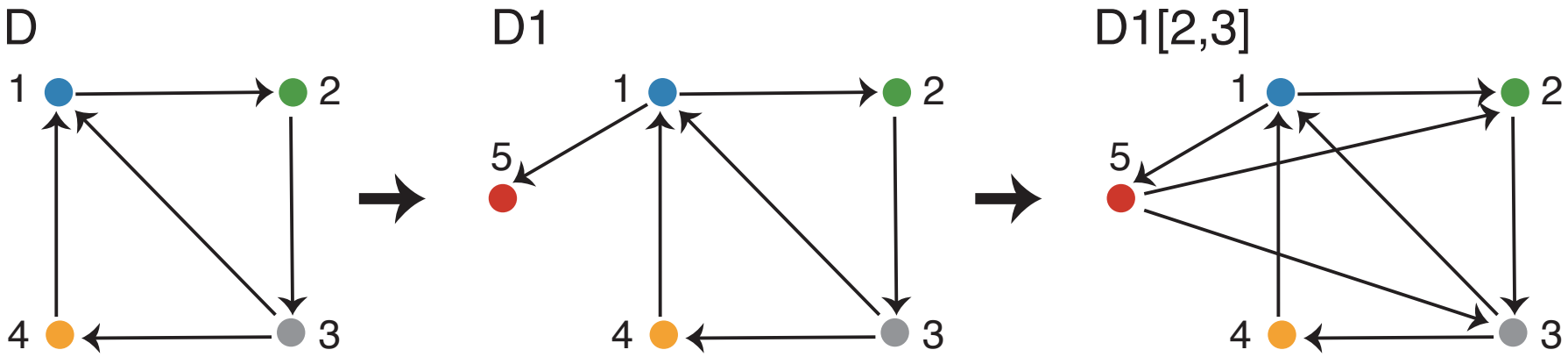
Taxonomy for oriented graphs on 5 neurons

Every oriented graph with no sinks on n=5 can be built up and named using these base graphs:



Taxonomy for oriented graphs on 5 neurons

Every oriented graph with no sinks on n=5 can be built up and named using these base graphs:



Relabeling nodes 4 & 5, this graph also corresponds to E2[3]

Dictionary of attractor types

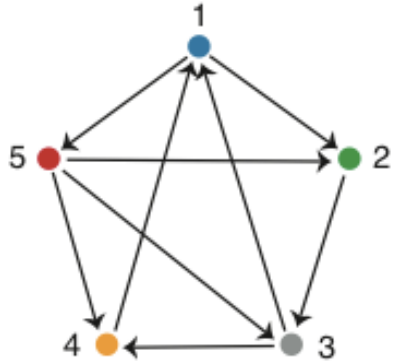
Using the taxonomy, we could identify 19 attractor types and classify graphs based on which attractor types they exhibited – this classification largely aligned with the taxonomy!

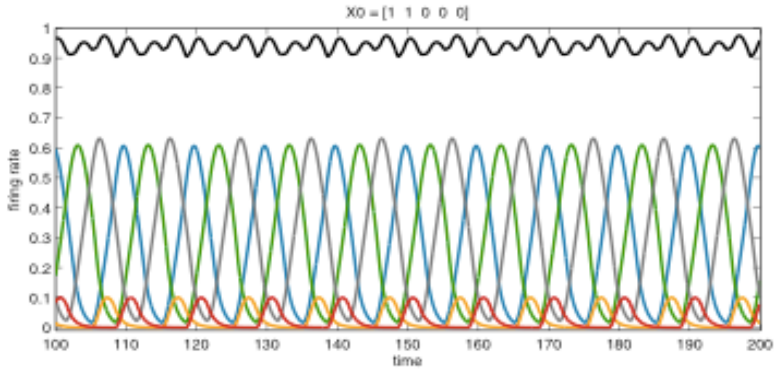
Attractor types include

- Limit cycles with simple cycles
- Limit cycles with synchronous firing
- Period-doubled limit cycles
- Quasperiodic attractors
- Chaotic attractors

Dictionary of attractor types

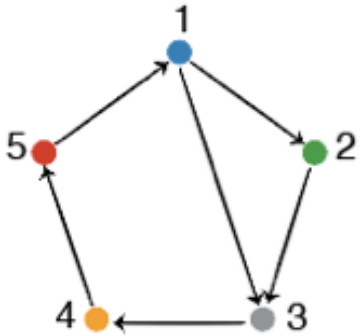
AT-4 (limit cycle)

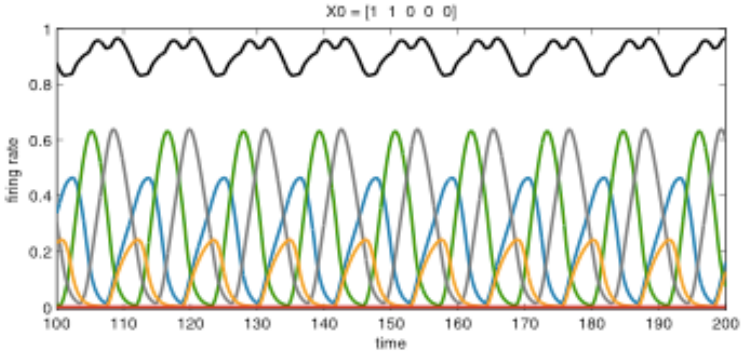




Rep. graph D1[2,3,4], seq 15234. All graphs: D1[2,3,4], D1[2,3] (aka E2[3]), D1[2,4], D1[2] (aka D2[3]), D2[1,3,4], D2[1,3] (aka E1[2]), D2[3,4], D3[4,*], E1[2,3,4], E1[2,4], E1[3,4], E1[3] (aka F2[1,3]), E1[4], E2[1,3,4], E2[3,4], E2[3] (aka D1[2,3]), E2[1,3] (aka E1[2,3]), E3[1,2,4], E3[1,4], E3[2,4], E3[4].

AT-11 (limit cycle)



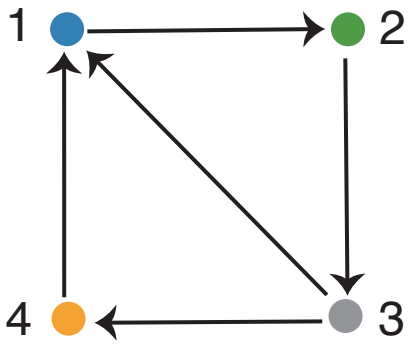


Rep. graph T4[1], seq 1234. All graphs: T4[1,2,3], T4[1,2], T4[1,3], T4[2,3]_seq1 (aka D1[4]_seq2), T4[1], T4[2]_seq1 (aka G1[4]_seq2), T4[3]_seq1, T4[3]_seq2.

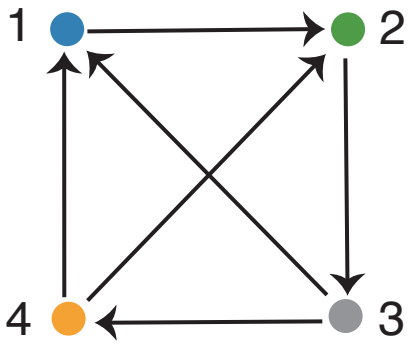
Some things we've learned…

1. Core 3-cycles yield attractors while non-core 3-cycles do not have corresponding attractors.
   More generally, we have rules for when certain subgraphs that support an attractor will have that attractor survive in the full network.

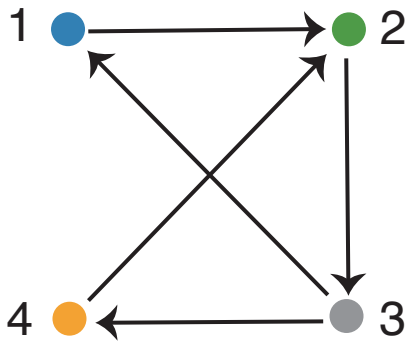
D



E



F



234 is a non-core 3-cycle in E

Some things we've learned…

1. Core 3-cycles yield attractors while non-core 3-cycles do not have corresponding attractors.

2. Sources die, i.e. do not participate in any attractors. More generally, nodes not receiving from the core cycle do not participate in the corresponding attractor.

Some things we've learned…

1. Core 3-cycles yield attractors while non-core 3-cycles do not have corresponding attractors.

2. Sources die, i.e. do not participate in any attractors. More generally, nodes not receiving from the core cycle do not participate in the corresponding attractor.

3. When two networks share certain common subgraphs, they tend to have at least one attractor in common.

   For $n \leq 5$ only 19 attractor types! (~200 observed attractors)

Some things we've learned…

1. Core 3-cycles yield attractors while non-core 3-cycles do not have corresponding attractors.

2. Sources die, i.e. do not participate in any attractors. More generally, nodes not receiving from the core cycle do not participate in the corresponding attractor.

3. When two networks share certain common subgraphs, they tend to have at least one attractor in common.

   For $n \leq 5$ only 19 attractor types! (~200 observed attractors)

4. Attractors can exhibit unexpected synchrony that is not the result of graph symmetries or even the weaker symmetries picked up by the algorithm.

Some things we've learned…

1. Core 3-cycles yield attractors while non-core 3-cycles do not have corresponding attractors.

2. Sources die, i.e. do not participate in any attractors. More generally, nodes not receiving from the core cycle do not participate in the corresponding attractor.

3. When two networks share certain common subgraphs, they tend to have at least one attractor in common.

   For $n \leq 5$ only 19 attractor types! (~200 observed attractors)

4. Attractors can exhibit unexpected synchrony that is not the result of graph symmetries or even the weaker symmetries picked up by the algorithm.

5. We're starting to understand how non-local properties of connectivity shape the emergent dynamics of a network.

Thanks!

Collaborators and consultants on this work:
Carina Curto (Penn State University)
Caitlyn Parmelee (Keene State College)
Samantha Moore (University of Northern Colorado)

Anda Degeratu (Freiberg)
Vladimir Itskov (Penn State)

Preprint https://arxiv.org/abs/1605.04463

Diversity of emergent dynamics in competitive threshold-linear networks: a preliminary report Katherine Morrison[1,2], Anda Degeratu[3], Vladimir Itskov[1] & Carina Curto[1]

Phd Thesis https://arxiv.org/abs/1702.06538

Applications of Discrete Mathematics for Understanding Dynamics of Synapses and Networks in Neuroscience Caitlyn Parmelee

In Prep **Predicting emergent sequences from network connectivity**

Katherine Morrison, Caitlyn Parmelee, Samantha Moore, Carina Curto

Funding: NIH R01 EB022862, NSF DMS 1516881

A competitive network theory of species diversity

Stefano Allesina[a,1] **and Jonathan M. Levine**[b]

[a]Department of Ecology and Evolution, Computation Institute, University of Chicago, Chicago, IL 60637; and [b]Department of Ecology, Evolution, and Marine Biology, University of California, Santa Barbara, CA 93106

5638–5642 | PNAS | **April 5, 2011** | vol. 108 | no. 14

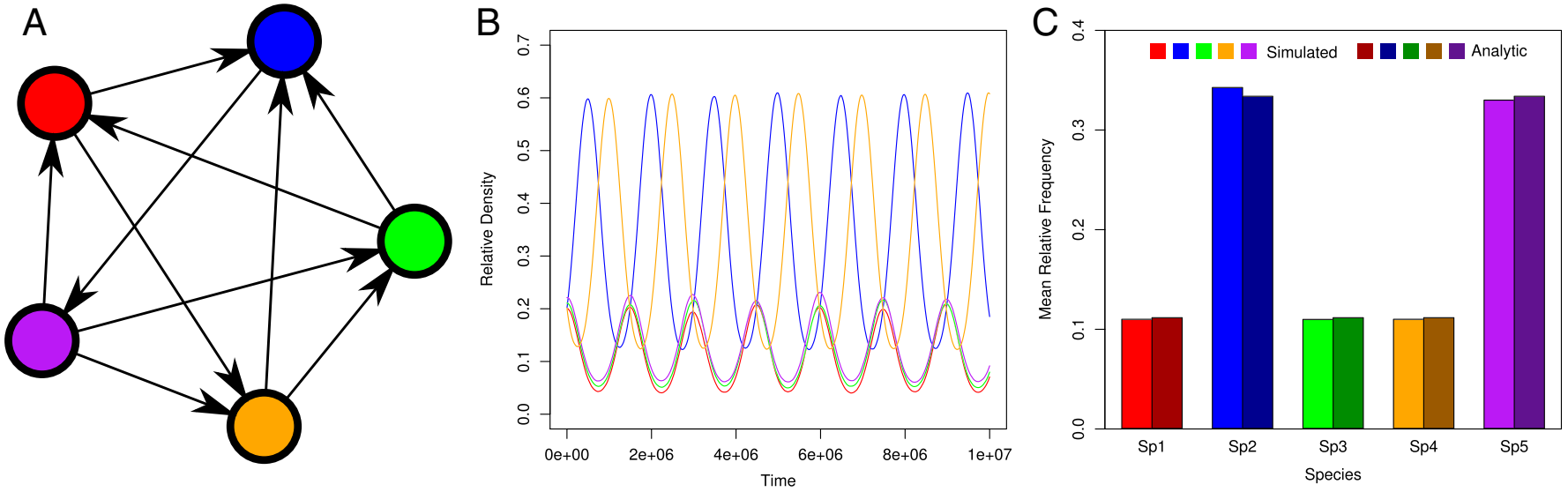


Fig. 1. (*A*) Species' competitive abilities can be represented in a tournament in which we draw an arrow from the inferior to the superior competitor for all species pairs. A tournament is a directed graph composed by n nodes (the species) connected by $n(n-1)/2$ edges (arrows). (*B*) Simulations of the dynamics for the tournament. The simulation begins with 25,000 individuals assigned to species at random (with equal probability per species). At each time step, we pick two individuals at random and allow the superior to replace the individual of the inferior. We repeat these competitions 10^7 times, which generates relative species abundances that oscillate around a characteristic value (*SI Text*). (*C*) The average simulated density of each species from *B* (shown in lighter bars) almost exactly matches the analytic result obtained using linear programming (shown in darker bars).

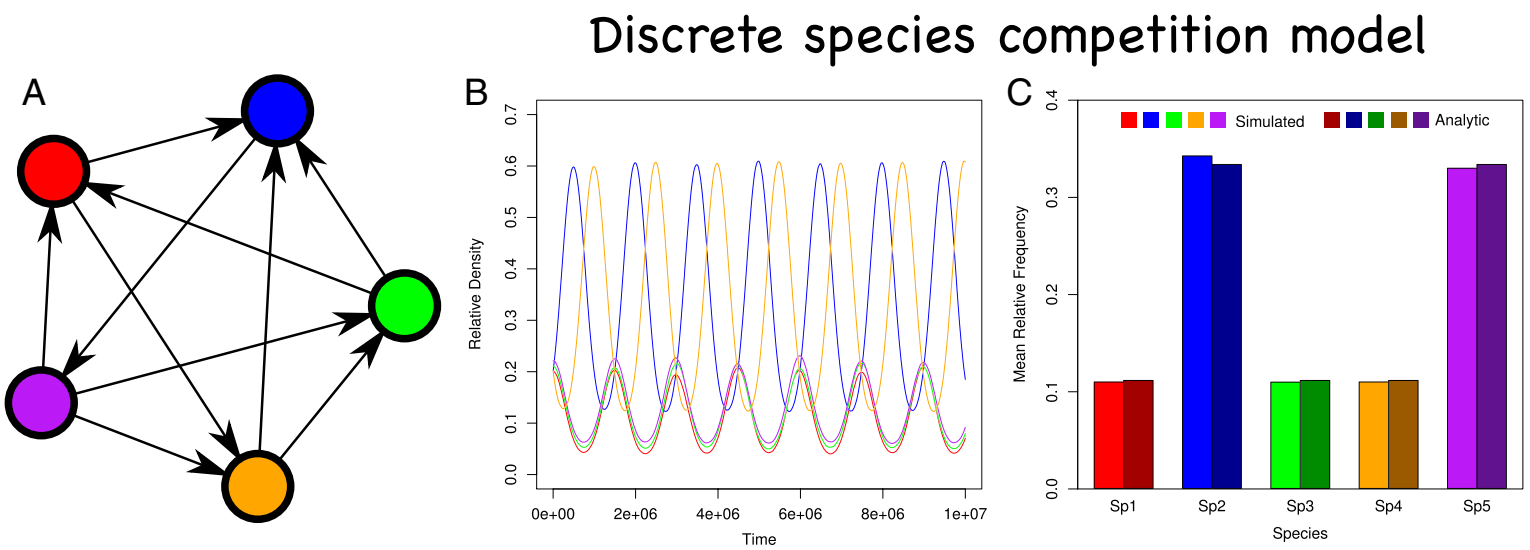
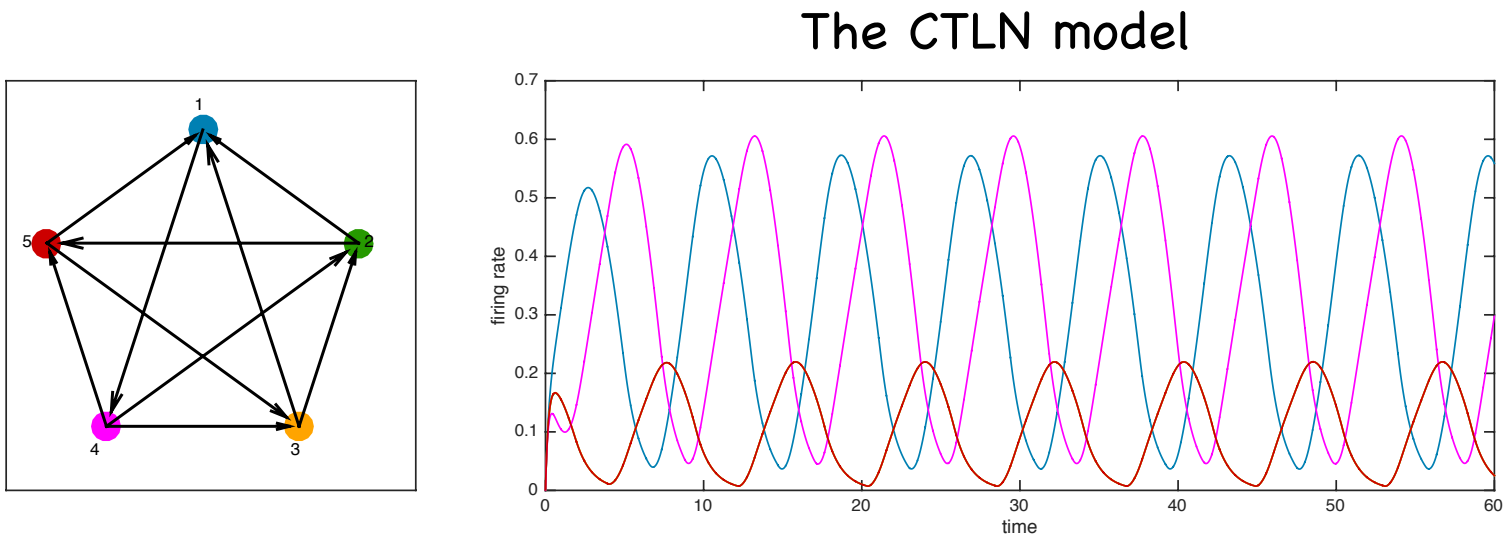
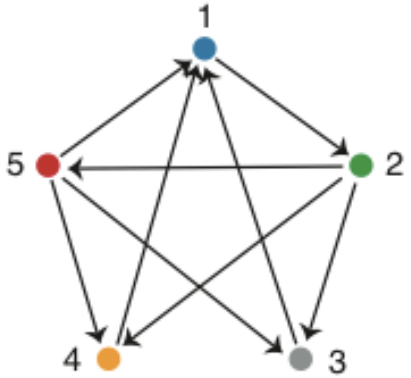


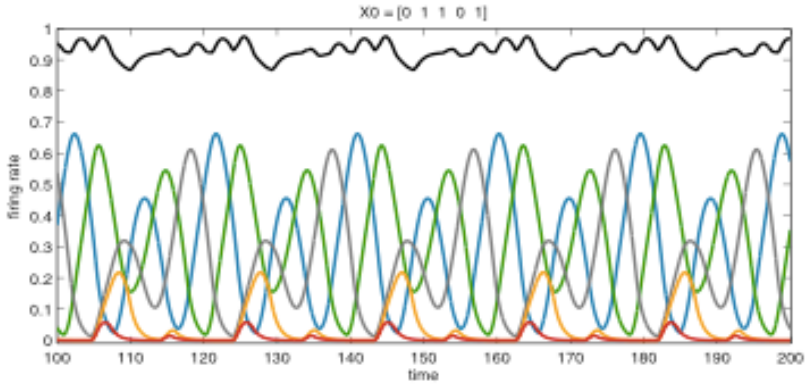
Fig. 1. (*A*) Species' competitive abilities can be represented in a tournament in which we draw an arrow from the inferior to the superior competitor for all species pairs. A tournament is a directed graph composed by n nodes (the species) connected by $n(n-1)/2$ edges (arrows). (*B*) Simulations of the dynamics for the tournament. The simulation begins with 25,000 individuals assigned to species at random (with equal probability per species). At each time step, we pick two individuals at random and allow the superior to replace the individual of the inferior. We repeat these competitions 10^7 times, which generates relative species abundances that oscillate around a characteristic value (*SI Text*). (*C*) The average simulated density of each species from *B* (shown in lighter bars) almost exactly matches the analytic result obtained using linear programming (shown in darker bars).



Dictionary of attractor types

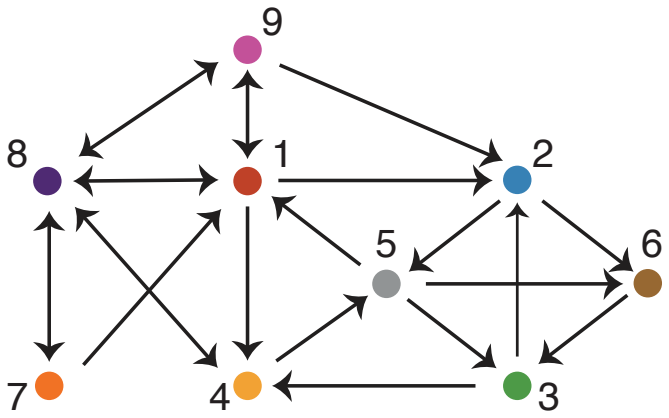
AT-16 (limit cycle)

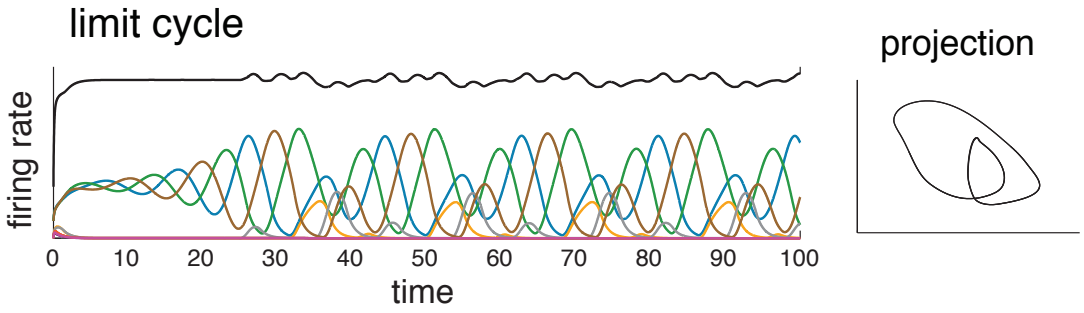


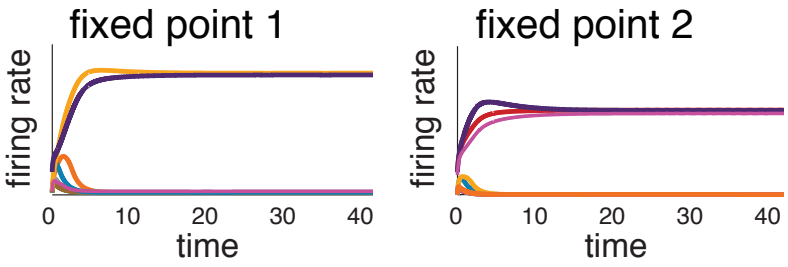


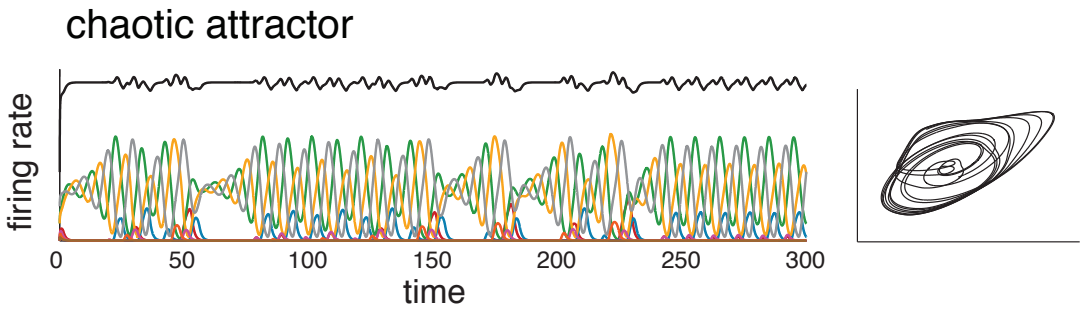
Rep. graph F3[1,2,4], seq 1254312543. All graphs: F1[2,3,4]_seq1, F1[2,3]_seq1 (aka E2[1]_seq1), F1[2,4]_seq1, F1[2]_seq1 (aka D2[1]_seq1), F1[4]_seq1, F1[3,4]_seq1, F1[3]_seq1 (aka F2[1]_seq1), F2[3]_seq1 (aka D1[3]_seq1), F2[3]_seq2 (aka D1[3]_seq2), F3[1,2,4]_seq1, F3[1,2,4]_seq2, F3[1,4]_seq1, F3[1,4]_seq2, E2[4]_seq1, D1[3,4]_seq2, E2[1,4]_seq1, E2[1,4]_seq2

A single network can display multiple attractors of different types

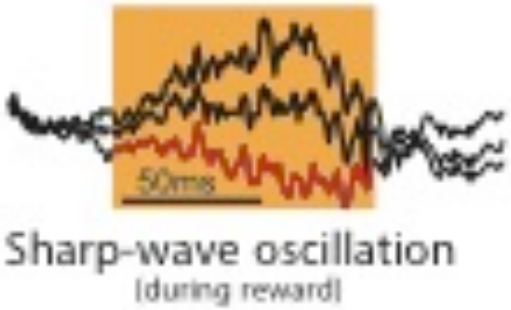


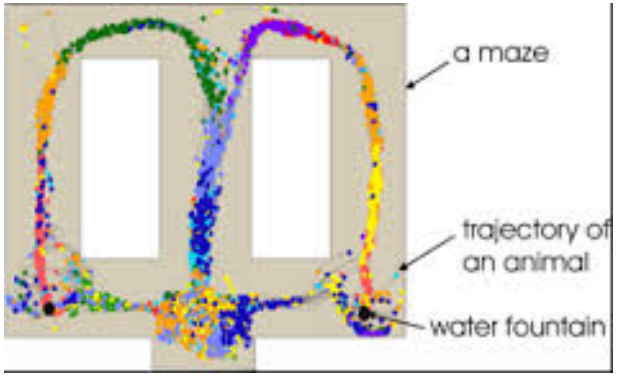


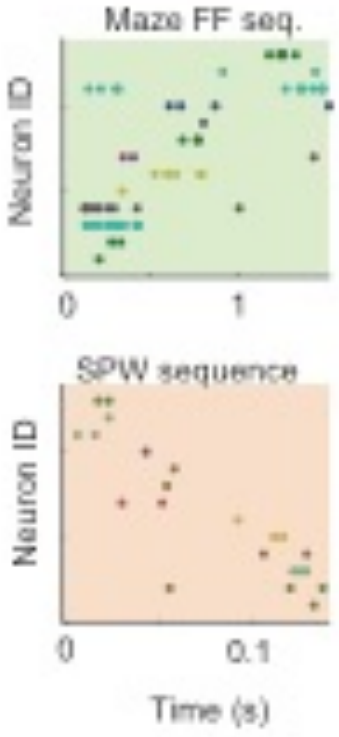




Ripple sequences in hippocampus







Data from Eva Pastalkova's Lab at Janelia Research Campus.