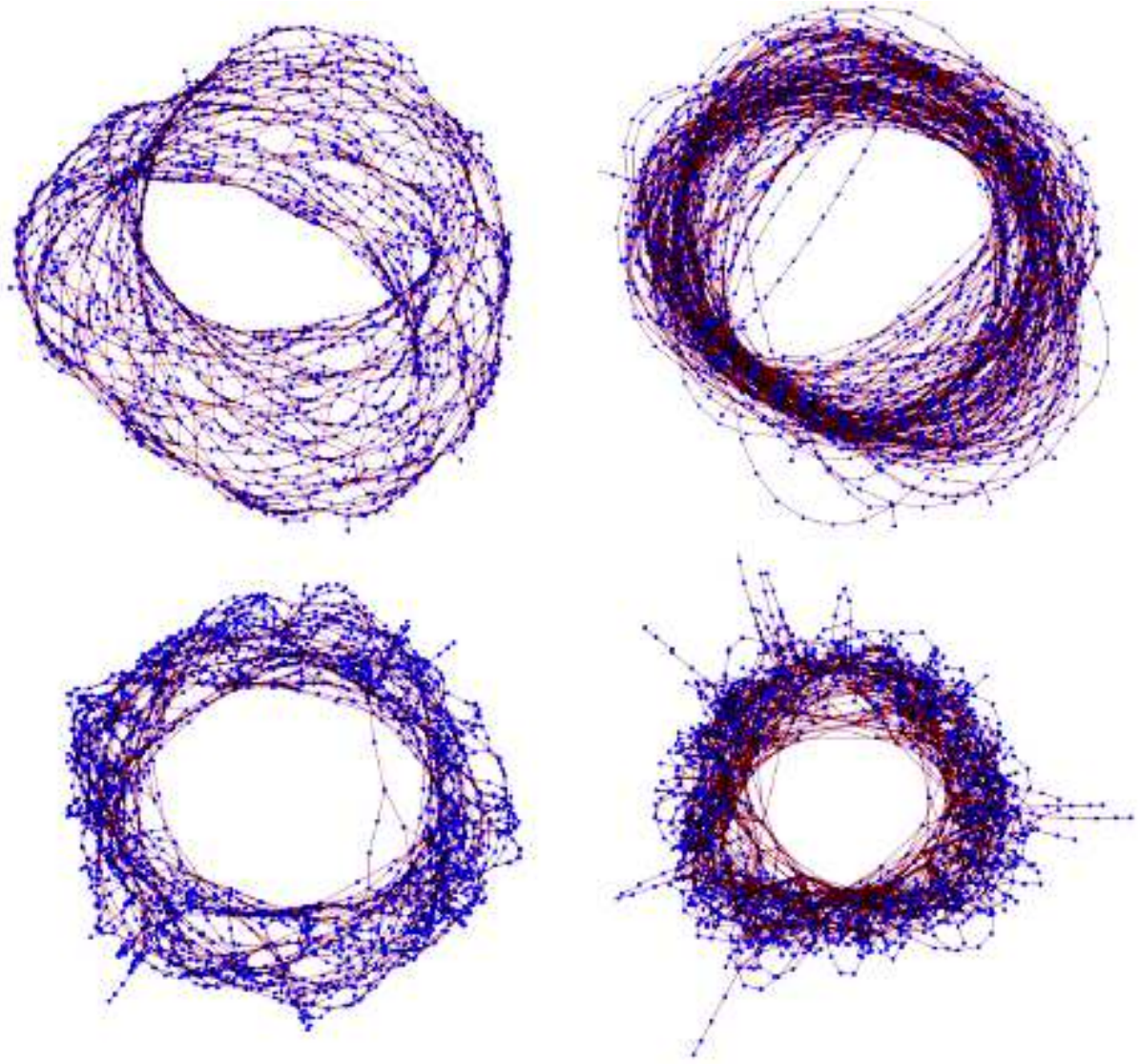
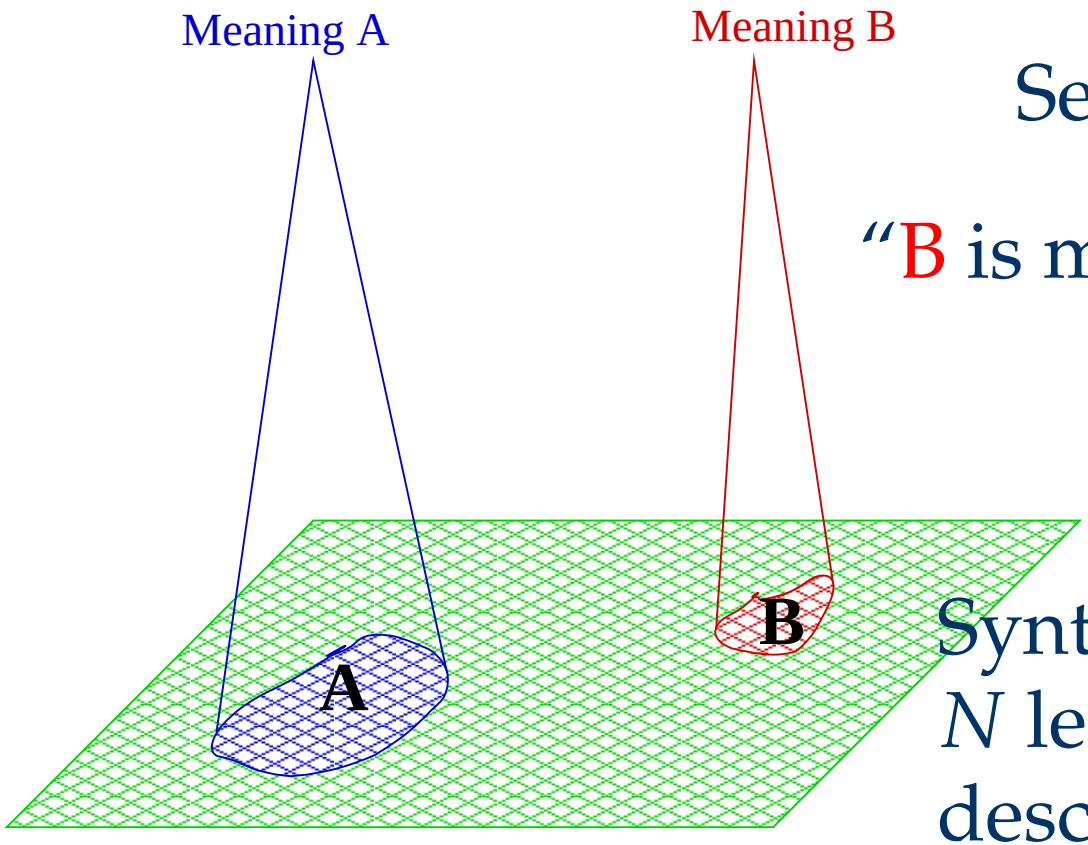
Generating Networks with Surplus Complexity



Complexity, or amount of information

Note: Information = data + meaning



Semantic Space

"B is more complex than A

Syntactic Space
N letters in alphabet
descriptions of length ℓ

Complexity as Information

We need two things to determine the complexity of something:

- An encoding over a finite alphabet (eg a bitstring)
- A classifier function (aka observer) of the strings that can determine whether two encodings refer to the same object

Kolmogorov complexity of a bit-string

- *Classifier* is a Turing machine
- *Encodings* are programs of the Turing machine that output the *thing* (a bitstring)
- In the limit as $n \to \infty$, complexity is dominated by the length of the shortest program (compiler theorem).
- Random strings are maximally complex

Effective complexity of English literature

- *Encoding* is the latin script encoding words of English
- *Classifier* is a human being deciding whether two strings of latin script mean the same thing.
- The vast majority of strings are meaningless (gibberish), hence have vanishing complexity. Random strings have low complexity.
- Repetitive, (algorithmic) strings are also fairly low complexity.

Encoding of Digraphs

n nodes, ℓ links, rank encoded linklist r

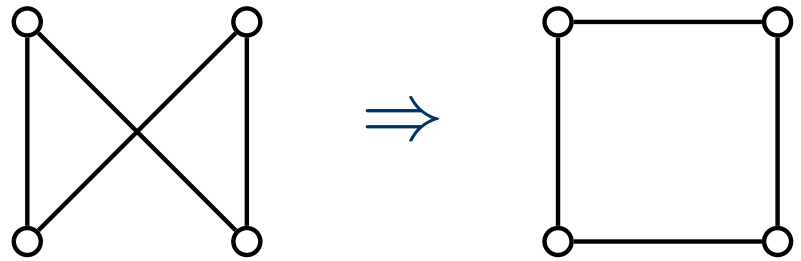
$$\overbrace{111\ldots10}^{\log_2 n+1}\ \overbrace{}^{\log_2 n}\ \overbrace{}^{\log_2 n(n-1)}\ \overbrace{\phantom{\binom{n(n-1)}{\ell}}}^{\log_2 \binom{n(n-1)}{\ell}}$$

$$\overbrace{111\ldots10}^{\log_2 n+1}\ \overbrace{n}^{\log_2 n}\ \overbrace{\ell}^{\log_2 n(n-1)}\ \overbrace{r}^{\log_2 \binom{n(n-1)}{\ell}}$$

- Empty and full digraphs have minimal complexity of $\approx 4\log_2 n$ bits
- Digraph has same complexity as its complement
- Complexity peaks at intermediate link counts

Classification of Digraphs

- Nodes are unlabelled
- Node position is irrelevant
- Two digraphs are equivalent if automorphic



Automorphism problem

- Determine if two (di)graphs are automorphic
- Count the number of automorphisms
- Suspected as being NP
- Practical algorithms exist: Nauty, Saucy, SuperNOVA

Weighted Links

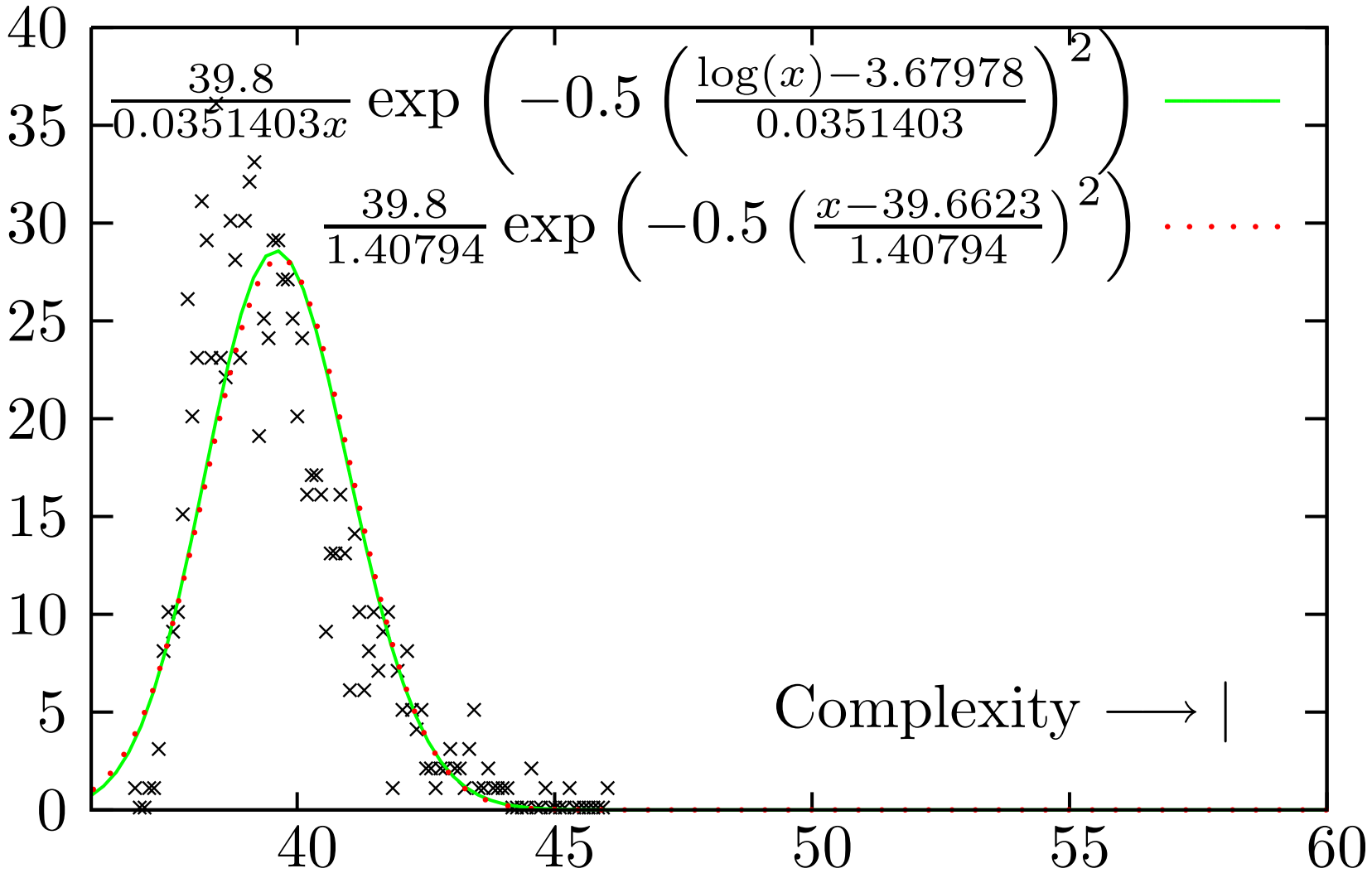
Want a graph that is "in between" two graph structures to have "in between" complexity. For network $N \times L$ with weights $w_i, \forall i \in L$

$$\mathcal{C}(N \times L) = \int_0^1 \mathcal{C}(N \times \{i \in L : w_i < w\})dw$$

Food Web Data

Dataset	nodes	links	$\mathcal{C}$	$e^{\langle \ln \mathcal{C}_{ER} \rangle}$	$\Delta = \mathcal{C} - e^{\langle \ln \mathcal{C}_{ER} \rangle}$	$\frac{\lvert \ln \mathcal{C} - \langle \ln \mathcal{C}_{ER} \rangle \rvert}{\sigma_{ER}}$
celegansneural	297	2345	442.7	251.6	191.1	29
celegansmetabolic	453	4050	25421.8	25387.2	34.6	∞
lesmis	77	508	199.7	114.2	85.4	24
adjnoun	112	850	3891	3890	0.98	∞
yeast	2112	4406	33500.6	30218.2	3282.4	113.0
Chesapeake	39	177	66.8	45.7	21.1	10.4
Everglades	69	912	54.5	32.7	21.8	11.8
Florida	128	2107	128.4	51.0	77.3	20.1
Maspalomas	24	83	70.3	61.7	8.6	5.3
Michigan	39	219	47.6	33.7	14.0	9.5
Mondego	46	393	45.2	32.2	13.0	10.0
Narragan	35	219	58.2	39.6	18.6	11.0
Rhode	19	54	36.3	30.3	6.0	5.3
StMarks	54	354	110.8	73.6	37.2	16.0

Shuffled model



Networks exhibiting complexity surplus

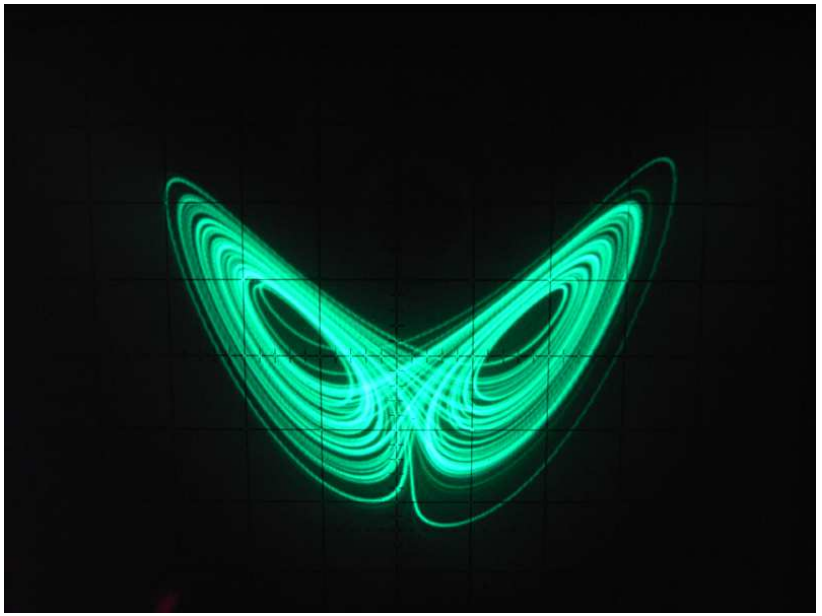
- Most real-world networks
- Not Erdös-Renyi networks (obviously)
- Not networks generated by *preferential attachment*
- Some evolutionary systems: *EcoLab*, Tierra, but not Webworld
- Networks induced by dynamical chaos
- Networks induced by cellular automata

Inducing a network from a discrete timeseries

- A timeseries $X = x_0, x_1, \ldots, x_N$, where $x_i \in \mathcal{X} \subset \mathbb{Z}$
- Construct a network with $|N|$ nodes.
- Link weights w_{ij} are given by the number of *transitions* between x_i and x_j in X.

Example: Lorenz system

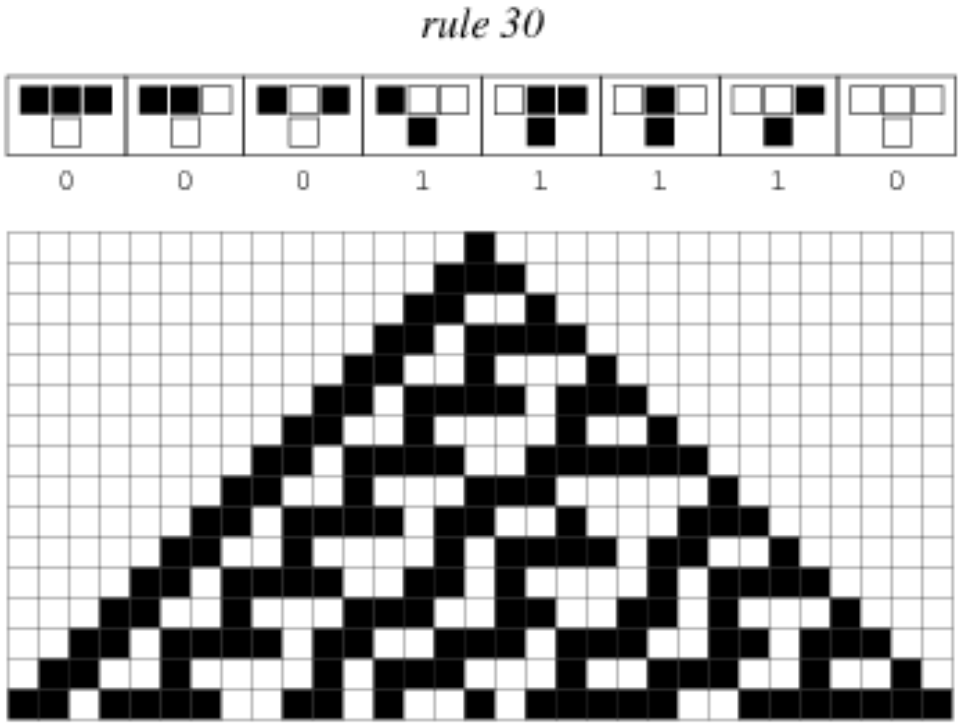
$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = x(\rho - z) - y$$
$$\dot{z} = xy - \beta z$$



Dynamical Chaos results

Dataset	nodes	links	$\mathcal{C}$	$e^{\langle \ln \mathcal{C}_{ER} \rangle}$	$\mathcal{C} - e^{\langle \ln \mathcal{C}_{ER} \rangle}$	$\frac{\lvert \ln \mathcal{C} - \langle \ln \mathcal{C}_{ER} \rangle \rvert}{\sigma_{ER}}$
celegansneural	297	2345	442.7	251.6	191.1	29
PA1	100	99	98.9	85.4	13.5	2.5
Lorenz	8000	62	560.2	56.0	504.2	58.3
Hénon-Heiles	10000	31	342.0	57.3	284.7	55.6

1D cellula automata



1D CA results