Controlled Markovian dynamics of structured graphs

ES Roberts, A Annibale, ACC Coolen

Dept of Mathematics and Randall Division
King's College London

NOMA11 @ Évora, Sept 15/16 2011

Outline

Outline

1. Background - tailored random graphs
2. Generating tailored random graphs numerically
3. Constrained Markovian graph dynamics
4. Degree-constrained dynamics of nondirected graphs
5. Degree-constrained dynamics of directed graphs
6. Summary

Outline

Outline

Outline

1. Background - tailored random graphs
2. Generating tailored random graphs numerically
3. Constrained Markovian graph dynamics
4. Degree-constrained dynamics of nondirected graphs
5. Degree-constrained dynamics of directed graphs
6. Summary

Outline

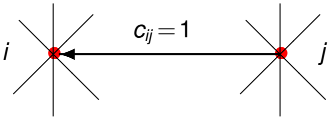
Background - tailored random graphs

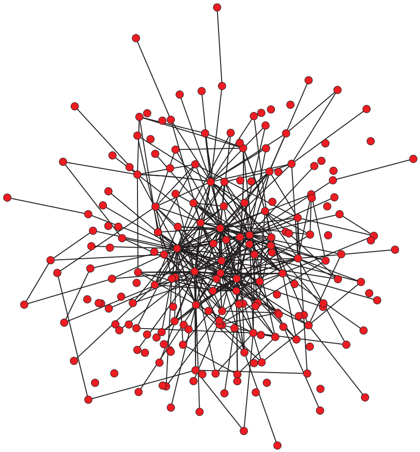
networks/graphs:

number of nodes: N
nodes (vertices): $i, j \in \{1, \ldots, N\}$

links (edges): $c_{ij} \in \{0, 1\}$
no self-links: $c_{ii} = 0$ for all i
graph: $\mathbf{c} = \{c_{ij}\}$

nondirected: $\forall (i,j): \; c_{ij} = c_{ji}$
directed: $\exists (i,j): \; c_{ij} \neq c_{ji}$





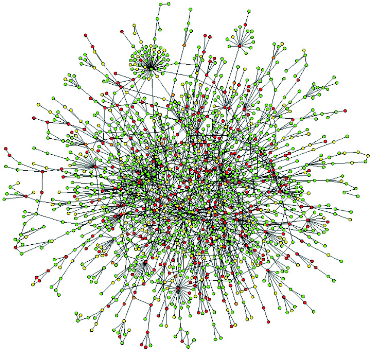
degrees: $k_i^{\rm in}(\mathbf{c}) = \sum_j c_{ij}, \quad k_i^{\rm out}(\mathbf{c}) = \sum_j c_{ji}$

$$\mathbf{k}^{\rm in}(\mathbf{c}) = (k_1^{\rm in}(\mathbf{c}), \ldots, k_N^{\rm in}(\mathbf{c})), \;\; \mathbf{k}^{\rm out}(\mathbf{c}) = \ldots$$

Networks in cellular biology

- **protein interaction networks**: (nondirected)

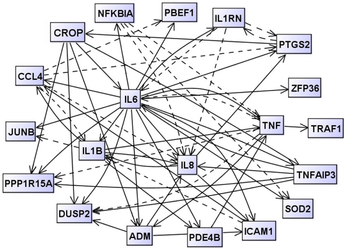
  nodes: proteins $i, j = 1 \ldots N$
  links: $c_{ij} = c_{ji} = 1$ if i can bind to j
  $c_{ij} = c_{ji} = 0$ otherwise
  $N \sim 10^4$, about 7 links/node



- **gene regulation networks**: (directed)

  nodes: proteins $i, j = 1 \ldots N$
  links: $c_{ij} = c_{ji} = 1$ if j is transcription factor of i
  $c_{ij} = c_{ji} = 0$ otherwise

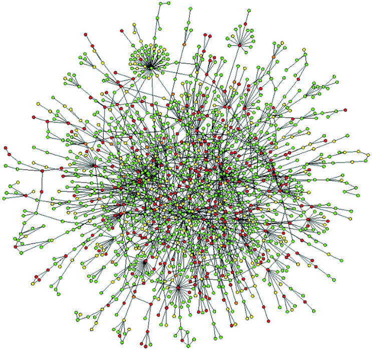
  $N \sim 10^4$, about 5 links/node



Networks in cellular biology

- **protein interaction networks**:
  (nondirected)

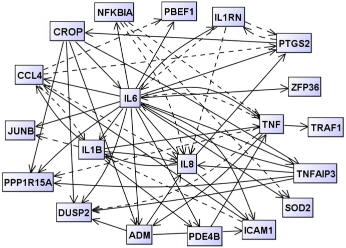
  nodes: proteins $i, j = 1 \ldots N$
  links: $c_{ij} = c_{ji} = 1$ if i can bind to j
  $c_{ij} = c_{ji} = 0$ otherwise
  $N \sim 10^4$, about 7 links/node



- **gene regulation networks**:
  (directed)

  nodes: proteins $i, j = 1 \ldots N$
  links: $c_{ij} = c_{ji} = 1$ if j is transcription factor of i
  $c_{ij} = c_{ji} = 0$ otherwise

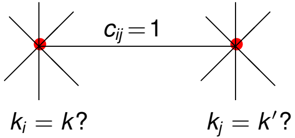
  $N \sim 10^4$, about 5 links/node



Quantify graph topology beyond degrees

joint degree statistics
of connected nodes

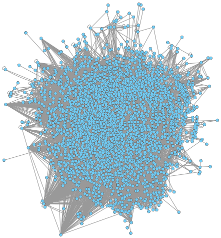
$$W(k,k'|\mathbf{c}) = \frac{1}{N\langle k\rangle}\sum_{ij} c_{ij}\delta_{k,k_i(\mathbf{c})}\delta_{k',k_j(\mathbf{c})}$$

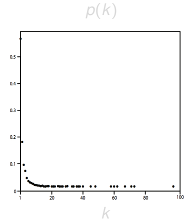


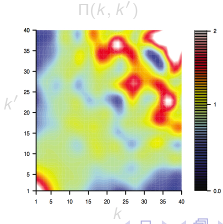
- $W(k|\mathbf{c}) = p(k|\mathbf{c})k/\langle k\rangle$
  so focus on:

$$\Pi(k,k'|\mathbf{c}) = \frac{W(k,k'|\mathbf{c})}{W(k|\mathbf{c})W(k'|\mathbf{c})}$$

$\Pi(k,k'|\mathbf{c}) \neq 1$:
structural information in degree correlations





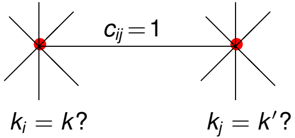


H sapiens PIN
$N = 9306$
$\langle k\rangle = 7.53$

Quantify graph topology beyond degrees

joint degree statistics
of connected nodes

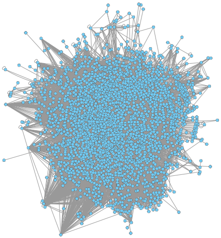
$$W(k,k'|\mathbf{c}) = \frac{1}{N\langle k\rangle}\sum_{ij} c_{ij}\,\delta_{k,k_i(\mathbf{c})}\,\delta_{k',k_j(\mathbf{c})}$$

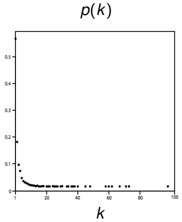


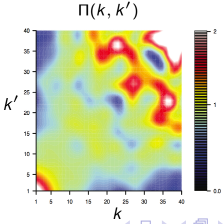
- $W(k|\mathbf{c}) = p(k|\mathbf{c})k/\langle k\rangle$
  so focus on:

$$\Pi(k,k'|\mathbf{c}) = \frac{W(k,k'|\mathbf{c})}{W(k|\mathbf{c})W(k'|\mathbf{c})}$$

$\Pi(k,k'|\mathbf{c}) \neq 1$:
structural information in degree correlations







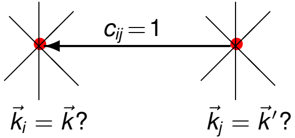
H sapiens PIN
$N = 9306$
$\langle k\rangle = 7.53$

for directed graphs:

joint in-out degree statistics
of connected nodes

$k_i \;\to\; \vec{k}_i = (k_i^{\rm in}, k_i^{\rm out})$

$$W(\vec{k},\vec{k}'|\mathbf{c}) = \frac{1}{N\langle k\rangle}\sum_{ij} c_{ij}\delta_{\vec{k},\vec{k}_i(\mathbf{c})}\delta_{\vec{k}',\vec{k}_j(\mathbf{c})}$$



- $W(\vec{k}|\mathbf{c}) \;\equiv \sum_{\vec{k}'} W(\vec{k},\vec{k}') = p(\vec{k}|\mathbf{c})k^{\rm in}/\langle k\rangle$
  $W(\vec{k}'|\mathbf{c}) \equiv \sum_{\vec{k}} W(\vec{k},\vec{k}') \;= p(\vec{k}'|\mathbf{c})k^{\rm out}/\langle k\rangle$

  so focus on:

$$\Pi(\vec{k},\vec{k}'|\mathbf{c}) = \frac{W(\vec{k},\vec{k}'|\mathbf{c})}{W_1(\vec{k}|\mathbf{c})W_2(\vec{k}'|\mathbf{c})}$$

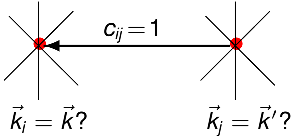
  $\Pi(\vec{k},\vec{k}'|\mathbf{c}) \neq 1$:
  structural information in degree correlations

for directed graphs:

joint in-out degree statistics
of connected nodes

$k_i \ \rightarrow \ \vec{k}_i = (k_i^{\rm in}, k_i^{\rm out})$

$$W(\vec{k},\vec{k}'|\mathbf{c}) = \frac{1}{N\langle k\rangle}\sum_{ij} c_{ij}\delta_{\vec{k},\vec{k}_i(\mathbf{c})}\delta_{\vec{k}',\vec{k}_j(\mathbf{c})}$$



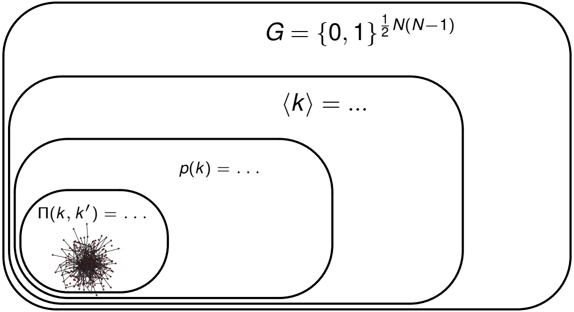
- $W(\vec{k}|\mathbf{c}) \ \equiv \sum_{\vec{k}'} W(\vec{k},\vec{k}') = p(\vec{k}|\mathbf{c})k^{\rm in}/\langle k\rangle$
  $W(\vec{k}'|\mathbf{c}) \equiv \sum_{\vec{k}} W(\vec{k},\vec{k}') \ = p(\vec{k}'|\mathbf{c})k^{\rm out}/\langle k\rangle$

  so focus on:

$$\Pi(\vec{k},\vec{k}'|\mathbf{c}) = \frac{W(\vec{k},\vec{k}'|\mathbf{c})}{W_1(\vec{k}|\mathbf{c})W_2(\vec{k}'|\mathbf{c})}$$

  $\Pi(\vec{k},\vec{k}'|\mathbf{c}) \neq 1$:
  structural information in degree correlations

Graph classification
via increasingly detailed
measurements



we are led to study:

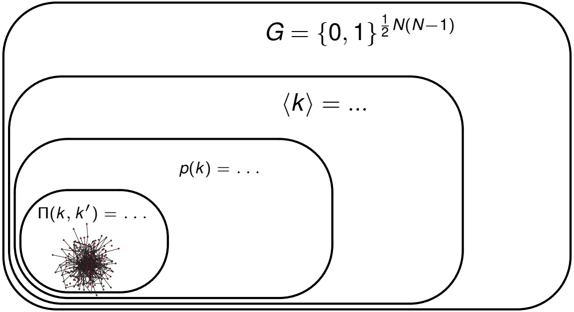
Tailored random graph ensembles

maximum entropy
random graph ensembles,
with prescribed values for $\langle k \rangle$, $p(k)$, $\Pi(k,k')$, ...

– proxies for real networks in stat mech process modelling
– complexity: how many networks exist with same features as **c**?
– hypothesis testing: graphs with controlled features as null models
– quantifying network dissimilarity

analytically in leading order in N

Graph classification
via increasingly detailed
measurements



we are led to study:

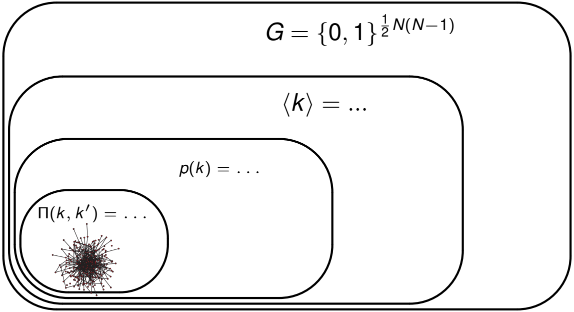
Tailored random graph ensembles

maximum entropy
random graph ensembles,
with prescribed values for $\langle k \rangle$, $p(k)$, $\Pi(k,k')$, ...

– proxies for real networks in stat mech process modelling
– complexity: how many networks exist with same features as **c**?
– hypothesis testing: graphs with controlled features as null models
– quantifying network dissimilarity

analytically in leading order in N

Graph classification
via increasingly detailed
measurements



we are led to study:

Tailored random graph ensembles

maximum entropy
random graph ensembles,
with prescribed values for $\langle k \rangle$, $p(k)$, $\Pi(k,k')$, ...

– proxies for real networks in stat mech process modelling
– complexity: how many networks exist with same features as **c**?
– hypothesis testing: graphs with controlled features as null models
– quantifying network dissimilarity

analytically in leading order in N

effective nr of graphs in ensemble $p(\mathbf{c})$,
complexity of tailored random graphs

$$\mathcal{N} = e^{N\langle k\rangle S}, \qquad S = -\frac{1}{N\langle k\rangle}\sum_{\mathbf{c}} p(\mathbf{c})\log p(\mathbf{c})$$

- nondirected graphs:

$$p(\mathbf{c}) = \frac{\prod_i \delta_{k_i,k_i(\mathbf{c})}}{Z}\prod_{i<j}\Big[\frac{\langle k\rangle}{N}\frac{W(k_i,k_j)}{p(k_i)p(k_j)}\delta_{c_{ij},1} + \Big(1-\frac{\langle k\rangle}{N}\frac{W(k_i,k_j)}{p(k_i)p(k_j)}\Big)\delta_{c_{ij},0}\Big]$$

– write $\delta_{k_i,k_i(\mathbf{c})} = (2\pi)^{-1}\int_0^{2\pi}\mathrm{d}\omega_i\ \mathrm{e}^{\mathrm{i}\omega_i[k_i-k_i(\mathbf{c})]}$
– factorisation over bonds, sum over graphs
– write as path integral over $P(k,\omega) = N^{-1}\sum_i \delta_{k,k_i}\delta(\omega-\omega_i)$
– integration via steepest descent
– solve saddle-point equation analytically

$$S = \frac{1}{2}[1+\log(\frac{N}{\langle k\rangle})] - \Big\{\frac{1}{\langle k\rangle}\sum_k p(k)\log[\frac{1}{\pi(k)}] + \frac{1}{2}\sum_{k,k'} W(k,k')\log\Big[\frac{W(k,k')}{W(k)W(k')}\Big]\Big\} + \epsilon_N$$

effective nr of graphs in ensemble $p(\mathbf{c})$,
complexity of tailored random graphs

$$\mathcal{N} = e^{N\langle k\rangle S}, \qquad S = -\frac{1}{N\langle k\rangle}\sum_{\mathbf{c}} p(\mathbf{c})\log p(\mathbf{c})$$

- nondirected graphs:

$$p(\mathbf{c}) = \frac{\prod_i \delta_{k_i,k_i(\mathbf{c})}}{Z}\prod_{i<j}\Big[\frac{\langle k\rangle}{N}\frac{W(k_i,k_j)}{p(k_i)p(k_j)}\delta_{c_{ij},1} + \Big(1-\frac{\langle k\rangle}{N}\frac{W(k_i,k_j)}{p(k_i)p(k_j)}\Big)\delta_{c_{ij},0}\Big]$$

– write $\delta_{k_i,k_i(\mathbf{c})} = (2\pi)^{-1}\int_0^{2\pi}\mathrm{d}\omega_i\ \mathrm{e}^{\mathrm{i}\omega_i[k_i-k_i(\mathbf{c})]}$

– factorisation over bonds, sum over graphs

– write as path integral over $P(k,\omega) = N^{-1}\sum_i \delta_{k,k_i}\delta(\omega-\omega_i)$

– integration via steepest descent

– solve saddle-point equation analytically

$$S = \frac{1}{2}[1+\log(\frac{N}{\langle k\rangle})] - \Big\{\frac{1}{\langle k\rangle}\sum_k p(k)\log[\frac{1}{\pi(k)}] + \frac{1}{2}\sum_{k,k'} W(k,k')\log\Big[\frac{W(k,k')}{W(k)W(k')}\Big]\Big\}$$

$$+\ \epsilon_N$$

effective nr of graphs in ensemble $p(\mathbf{c})$,
complexity of tailored random graphs

$$\mathcal{N} = e^{N\langle k\rangle S}, \qquad S = -\frac{1}{N\langle k\rangle}\sum_{\mathbf{c}} p(\mathbf{c})\log p(\mathbf{c})$$

- nondirected graphs:

$$p(\mathbf{c}) = \frac{\prod_i \delta_{k_i,k_i(\mathbf{c})}}{Z}\prod_{i<j}\Big[\frac{\langle k\rangle}{N}\frac{W(k_i,k_j)}{p(k_i)p(k_j)}\delta_{c_{ij},1} + \Big(1-\frac{\langle k\rangle}{N}\frac{W(k_i,k_j)}{p(k_i)p(k_j)}\Big)\delta_{c_{ij},0}\Big]$$

– write $\delta_{k_i,k_i(\mathbf{c})} = (2\pi)^{-1}\int_0^{2\pi}\mathrm{d}\omega_i\ \mathrm{e}^{\mathrm{i}\omega_i[k_i-k_i(\mathbf{c})]}$

– factorisation over bonds, sum over graphs

– write as path integral over $P(k,\omega) = N^{-1}\sum_i \delta_{k,k_i}\delta(\omega-\omega_i)$

– integration via steepest descent

– solve saddle-point equation analytically

$$S = \frac{1}{2}[1+\log(\frac{N}{\langle k\rangle})] - \Big\{\frac{1}{\langle k\rangle}\sum_k p(k)\log[\frac{1}{\pi(k)}] + \frac{1}{2}\sum_{k,k'} W(k,k')\log\Big[\frac{W(k,k')}{W(k)W(k')}\Big]\Big\} + \epsilon_N$$

effective nr of graphs in ensemble $p(\mathbf{c})$,
complexity of tailored random graphs

$$\mathcal{N} = e^{N\langle k\rangle S}, \qquad S = -\frac{1}{N\langle k\rangle}\sum_{\mathbf{c}} p(\mathbf{c})\log p(\mathbf{c})$$

- directed graphs:

$$p(\mathbf{c}) = \frac{\prod_i \delta_{\vec{k},\vec{k}_i(\mathbf{c})}}{Z}\prod_{i<j}\Big[\frac{\langle k\rangle}{N}\frac{W(\vec{k}_i,\vec{k}_j)}{p(\vec{k}_i)p(\vec{k}_j)}\delta_{c_{ij},1} + \Big(1-\frac{\langle k\rangle}{N}\frac{W(\vec{k}_i,\vec{k}_j)}{p(\vec{k}_i)p(\vec{k}_j)}\Big)\delta_{c_{ij},0}\Big]$$

with $\vec{k}_i(\mathbf{c}) = (k_i^{\rm in}(\mathbf{c}), k_i^{\rm out}(\mathbf{c}))$

similar methods ...
final result:

$$S = 1+\log(\frac{N}{\langle k\rangle})$$
$$-\Big\{\frac{1}{\langle k\rangle}\sum_{\vec{k}} p(\vec{k})\log[\frac{1}{\pi(k^{\rm in})\pi(k^{\rm out})}] + \sum_{\vec{k},\vec{k}'} W(\vec{k},\vec{k}')\log\Big[\frac{W(\vec{k},\vec{k}')}{W(\vec{k})W(\vec{k}')}\Big]\Big\}$$
$$+\ \epsilon_N$$

effective nr of graphs in ensemble $p(\mathbf{c})$,
complexity of tailored random graphs

$$\mathcal{N} = e^{N\langle k\rangle S}, \qquad S = -\frac{1}{N\langle k\rangle}\sum_{\mathbf{c}} p(\mathbf{c})\log p(\mathbf{c})$$

- directed graphs:

$$p(\mathbf{c}) = \frac{\prod_i \delta_{\vec{k},\vec{k}_i(\mathbf{c})}}{Z}\prod_{i<j}\Big[\frac{\langle k\rangle}{N}\frac{W(\vec{k}_i,\vec{k}_j)}{p(\vec{k}_i)p(\vec{k}_j)}\delta_{c_{ij},1} + \Big(1-\frac{\langle k\rangle}{N}\frac{W(\vec{k}_i,\vec{k}_j)}{p(\vec{k}_i)p(\vec{k}_j)}\Big)\delta_{c_{ij},0}\Big]$$

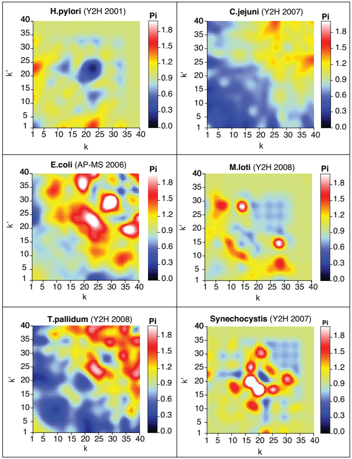
with $\vec{k}_i(\mathbf{c}) = (k_i^{\rm in}(\mathbf{c}), k_i^{\rm out}(\mathbf{c}))$

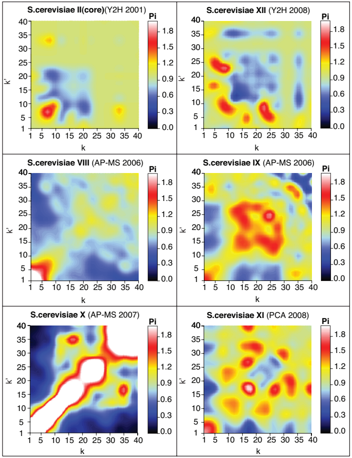
similar methods ...
final result:

$$\begin{aligned} S &= 1+\log(\frac{N}{\langle k\rangle}) \\ &\quad -\Big\{\frac{1}{\langle k\rangle}\sum_{\vec{k}} p(\vec{k})\log[\frac{1}{\pi(k^{\rm in})\pi(k^{\rm out})}] + \sum_{\vec{k},\vec{k}'} W(\vec{k},\vec{k}')\log\Big[\frac{W(\vec{k},\vec{k}')}{W(\vec{k})W(\vec{k}')}\Big]\Big\} \\ &\quad + \epsilon_N \end{aligned}$$

Information in degree correlations?

plot $\Pi(k,k') = W(k,k')/W(k)W(k')$
for protein interaction networks:
a problem ...





structural dissimilarity of graphs $\mathbf{c}_A$ and $\mathbf{c}_B$, based on
Information-theoretic distance between associated ensembles

$$D_{AB} = \frac{1}{2N} \sum_{\mathbf{c} \in G} \left\{ p(\mathbf{c}|p_A, W_A) \log \left[\frac{p(\mathbf{c}|p_A, W_A)}{p(\mathbf{c}|p_B, W_B)} \right] + p(\mathbf{c}|p_B, W_B) \log \left[\frac{p(\mathbf{c}|p_B, W_B)}{p(\mathbf{c}|p_A, W_A)} \right] \right\}$$

same methods as in calculating
Shannon entropy:

$$\begin{aligned} D_{AB} = {} & \frac{1}{2} \sum_k p_A(k) \log \left[\frac{p_A(k)}{p_B(k)} \right] + \frac{1}{2} \sum_k p_B(k) \log \left[\frac{p_B(k)}{p_A(k)} \right] \\ & + \frac{1}{4\langle k \rangle_A} \sum_{kk'} p_A(k) p_A(k') kk' \Pi_A(k,k') \log \left[\frac{\Pi_A(k,k')}{\Pi_B(k,k')} \right] \\ & + \frac{1}{4\langle k \rangle_B} \sum_{kk'} p_B(k) p_B(k') kk' \Pi_B(k,k') \log \left[\frac{\Pi_B(k,k')}{\Pi_A(k,k')} \right] \\ & + \frac{1}{2} \sum_k p_A(k) k \log \rho_{AB}(k) + \frac{1}{2} \sum_k p_B(k) k \log \rho_{BA}(k) \end{aligned}$$

with

$$\Pi(k,k') = W(k,k)/W(k)W(k'), \qquad \rho_{AB}(k) = \sum_{k'} \Pi_B(k,k') W_A(k') \rho_{AB}^{-1}(k')$$

structural dissimilarity of graphs $\mathbf{c}_A$ and $\mathbf{c}_B$, based on
Information-theoretic distance between associated ensembles

$$D_{AB} = \frac{1}{2N} \sum_{\mathbf{c}\in G} \left\{ p(\mathbf{c}|p_A, W_A) \log \left[\frac{p(\mathbf{c}|p_A, W_A)}{p(\mathbf{c}|p_B, W_B)} \right] + p(\mathbf{c}|p_B, W_B) \log \left[\frac{p(\mathbf{c}|p_B, W_B)}{p(\mathbf{c}|p_A, W_A)} \right] \right\}$$

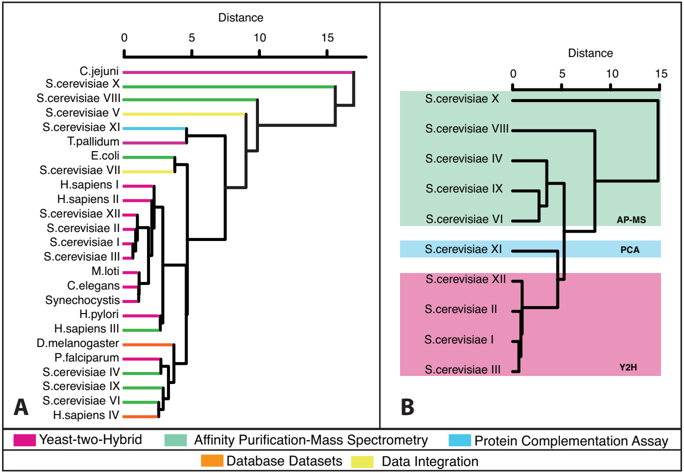
same methods as in calculating
Shannon entropy:

$$\begin{aligned} D_{AB} &= \frac{1}{2} \sum_k p_A(k) \log \left[\frac{p_A(k)}{p_B(k)} \right] + \frac{1}{2} \sum_k p_B(k) \log \left[\frac{p_B(k)}{p_A(k)} \right] \\ &+ \frac{1}{4\langle k \rangle_A} \sum_{kk'} p_A(k) p_A(k') kk' \Pi_A(k,k') \log \left[\frac{\Pi_A(k,k')}{\Pi_B(k,k')} \right] \\ &+ \frac{1}{4\langle k \rangle_B} \sum_{kk'} p_B(k) p_B(k') kk' \Pi_B(k,k') \log \left[\frac{\Pi_B(k,k')}{\Pi_A(k,k')} \right] \\ &+ \frac{1}{2} \sum_k p_A(k) k \log \rho_{AB}(k) + \frac{1}{2} \sum_k p_B(k) k \log \rho_{BA}(k) \end{aligned}$$

with

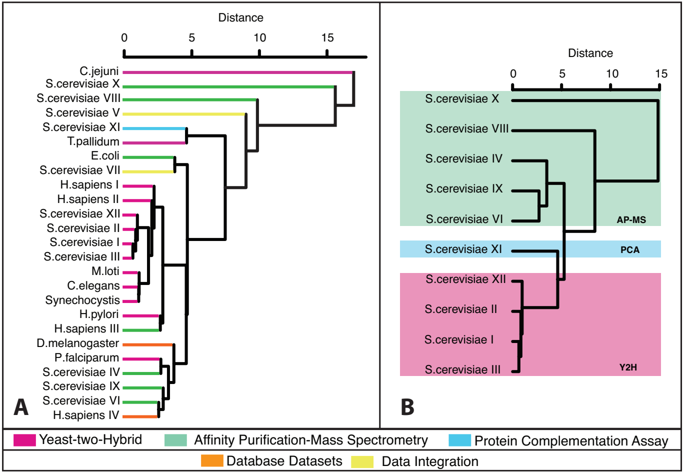
$$\Pi(k,k') = W(k,k)/W(k)W(k'), \qquad \rho_{AB}(k) = \sum_{k'} \Pi_B(k,k') W_A(k') \rho_{AB}^{-1}(k')$$

clustering of protein interaction networks with information-theoretic distance measure



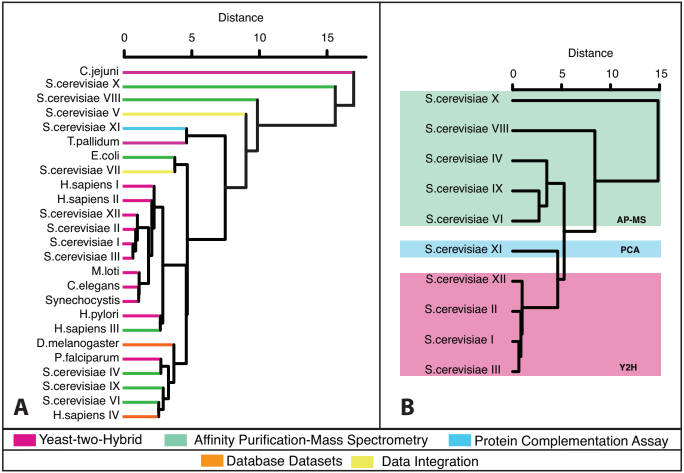
- PINs of same species and measured via same experimental method are statistically similar (in spite of limited overlap)
- PINs measured via same method cluster together, revealing **bias introduced by experimental method** that overrules species information

clustering of protein interaction networks with information-theoretic distance measure

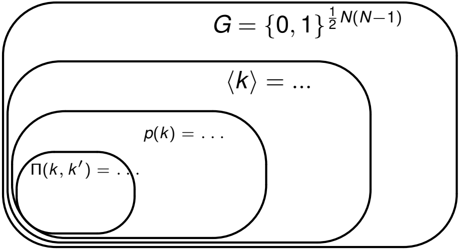


- PINs of same species and measured via same experimental method are statistically similar (in spite of limited overlap)
- PINs measured via same method cluster together, revealing **bias introduced by experimental method** that overrules species information

clustering of protein interaction networks with information-theoretic distance measure



- PINs of same species and measured via same experimental method are statistically similar (in spite of limited overlap)
- PINs measured via same method cluster together, revealing **bias introduced by experimental method** that overrules species information



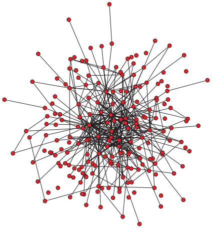
finally: let us

generate tailored random graphs

from the above families
numerically ...

Generating tailored random graphs numerically

$G[\mathbf{k}]$: all nondirected graphs $\mathbf{c}$ with degrees $\mathbf{k}$



how to generate

- random $\mathbf{c} \in G[\mathbf{k}]$, with uniform probability
- random $\mathbf{c} \in G[\mathbf{k}]$, with specified probability $p(\mathbf{c})$ (e.g. tailored graphs)

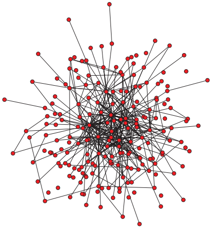
available approaches

- matching algorithm (Bender & Canfield, 1978)
  builds one random graph $\mathbf{c}$ with specified degrees $\mathbf{k}$
  (assign k_i 'stubs' to each node i, then randomly connect pairs of 'stubs')
- edge switching algorithm (Seidel, 1976; Taylor, 1981)
  ergodic Markov process in $G[\mathbf{k}]$
  $\mathbf{c} \rightarrow \mathbf{c}' \rightarrow \mathbf{c}'' \rightarrow \ldots$

– sampling all graphs in $G[\mathbf{k}]$: in principle easy
– main problem: sampling with **correct probabilities**
– matching and edge switching: both **biased** (yet widely used ...)

Generating tailored random graphs numerically

$G[\mathbf{k}]$: all nondirected graphs $\mathbf{c}$ with degrees $\mathbf{k}$



how to generate

- random $\mathbf{c} \in G[\mathbf{k}]$, with uniform probability
- random $\mathbf{c} \in G[\mathbf{k}]$, with specified probability $p(\mathbf{c})$ (e.g. tailored graphs)

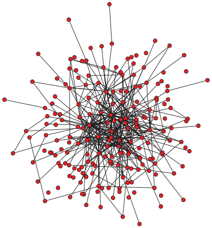
available approaches

- matching algorithm (Bender & Canfield, 1978)
  builds one random graph $\mathbf{c}$ with specified degrees $\mathbf{k}$
  (assign k_i 'stubs' to each node i, then randomly connect pairs of 'stubs')
- edge switching algorithm (Seidel, 1976; Taylor, 1981)
  ergodic Markov process in $G[\mathbf{k}]$
  $\mathbf{c} \to \mathbf{c}' \to \mathbf{c}'' \to \ldots$

– sampling all graphs in $G[\mathbf{k}]$: in principle easy
– main problem: sampling with **correct probabilities**
– matching and edge switching: both **biased** (yet widely used ...)

Generating tailored random graphs numerically

$G[\mathbf{k}]$: all nondirected graphs **c** with degrees **k**



how to generate

- random $\mathbf{c} \in G[\mathbf{k}]$, with uniform probability
- random $\mathbf{c} \in G[\mathbf{k}]$, with specified probability $p(\mathbf{c})$
  (e.g. tailored graphs)

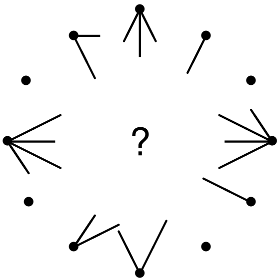
available approaches

- matching algorithm (Bender & Canfield, 1978)
  builds one random graph **c** with specified degrees **k**
  (assign k_i 'stubs' to each node i, then randomly connect pairs of 'stubs')
- edge switching algorithm (Seidel, 1976; Taylor, 1981)
  ergodic Markov process in $G[\mathbf{k}]$
  $\mathbf{c} \to \mathbf{c}' \to \mathbf{c}'' \to \ldots$

– sampling all graphs in $G[\mathbf{k}]$: in principle easy
– main problem: sampling with **correct probabilities**
– matching and edge switching: both **biased** (yet widely used ...)

Matching algorithm

- stochastic growth dynamics
- start with graph without any links
- pick at random two nodes whose in- and out degrees have not yet reached required values, and connect these if possible (Bender & Canfield, 1978)



bond creation as elementary moves:
$c_{ij} = 0 \;\rightarrow\; c_{ij} = 1$
if $k_i^{\rm in}(\mathbf{c}) < k_i^{\rm in}$ and $k_j^{\rm out}(\mathbf{c}) < k_j^{\rm out}$

origin of sampling bias:

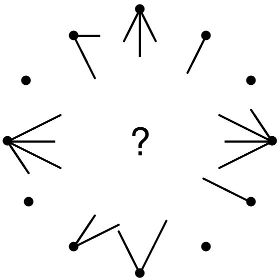
– process can terminate before $k_i(\mathbf{c}) = k_i$ for all i
(e.g. if remaining 'stubs' require self-loops)

– requires 'backtracking' which creates correlations
between graph realisations

dangerous for scale-free graphs ...

Matching algorithm

- stochastic growth dynamics
- start with graph without any links
- pick at random two nodes whose in- and out degrees have not yet reached required values, and connect these if possible (Bender & Canfield, 1978)



bond creation as elementary moves:
$c_{ij} = 0 \ \rightarrow \ c_{ij} = 1$
if $k_i^{\rm in}(\mathbf{c}) < k_i^{\rm in}$ and $k_j^{\rm out}(\mathbf{c}) < k_j^{\rm out}$

origin of sampling bias:

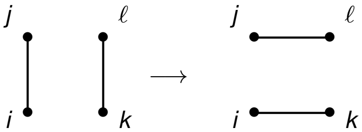
– process can terminate before $k_i(\mathbf{c}) = k_i$ for all i
(e.g. if remaining 'stubs' require self-loops)

– requires 'backtracking' which creates correlations
between graph realisations

dangerous for scale-free graphs ...

Edge switching

- construct arbitrary graph with degrees **k**
- shuffle links repeatedly via randomly drawn ‘edge swaps’ (Seidel, 1976)



edge swaps as elementary moves:

– preserve the degrees of all nodes
– are ergodic on $G[\mathbf{k}]$ (Taylor, 1981)

origin of sampling bias:

nr of possible moves
depends on state **c**!

result:
stationary state of Markov chain favours high-mobility graphs

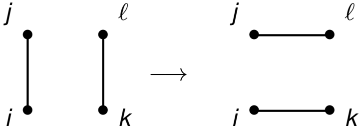
many possible moves

only one move ...

dangerous for scale-free graphs ...

Edge switching

- construct arbitrary graph with degrees **k**
- shuffle links repeatedly via randomly drawn 'edge swaps' (Seidel, 1976)



edge swaps as elementary moves:

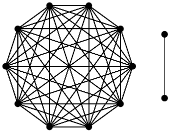
– preserve the degrees of all nodes
– are ergodic on $G[\mathbf{k}]$ (Taylor, 1981)

origin of sampling bias:

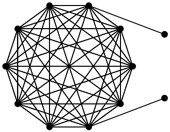
nr of possible moves
depends on state **c**!

result:
stationary state of Markov chain
favours high-mobility graphs

many possible moves



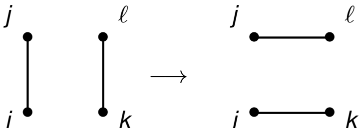
only one move ...



dangerous for scale-free graphs ...

Edge switching

- construct arbitrary graph with degrees **k**
- shuffle links repeatedly via randomly drawn ‘edge swaps’ (Seidel, 1976)



edge swaps as elementary moves:

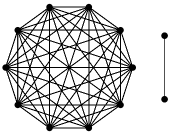
– preserve the degrees of all nodes
– are ergodic on $G[\mathbf{k}]$ (Taylor, 1981)

origin of sampling bias:

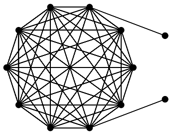
nr of possible moves
depends on state **c**!

result:
stationary state of Markov chain favours high-mobility graphs

many possible moves



only one move ...



dangerous for scale-free graphs ...

Constrained Markovian graph dynamics

need to study graph dynamics more systematically ...

- constraints:
  $G[\star] \subseteq G$: all $\mathbf{c} \in G$ that satisfy constraints $\star$

- stochastic graph dynamics as a Markov chain,
  transition probabilities $W(\mathbf{c}|\mathbf{c}')$

$$\forall \mathbf{c} \in G[\star]: \qquad p_{t+1}(\mathbf{c}) = \sum_{\mathbf{c}' \in G[\star]} W(\mathbf{c}|\mathbf{c}')p_t(\mathbf{c}')$$

- allowed moves (exclude identity):

  Φ: set of allowed moves $F: G_F[\star] \to G[\star]$
  $G_F[\star]$: those $\mathbf{c} \in G[\star]$ on which F can act

  all moves are auto-invertible: $(\forall F \in \Phi): F^2 = \mathbb{I}$
  Φ is ergodic on $G[\star]$

- graph mobility $n(\mathbf{c})$:

$$n(\mathbf{c}) = \sum_{F \in \Phi} I_F(\mathbf{c}), \qquad I_F(\mathbf{c}) = \begin{cases} 1 & \text{if } \mathbf{c} \in G_F[\star] \\ 0 & \text{if } \mathbf{c} \notin G_F[\star] \end{cases}$$

Constrained Markovian graph dynamics

need to study graph dynamics more systematically ...

- constraints:
  $G[\star] \subseteq G$: all $\mathbf{c} \in G$ that satisfy constraints $\star$

- stochastic graph dynamics as a Markov chain,
  transition probabilities $W(\mathbf{c}|\mathbf{c}')$

$$\forall \mathbf{c} \in G[\star]: \qquad p_{t+1}(\mathbf{c}) = \sum_{\mathbf{c}' \in G[\star]} W(\mathbf{c}|\mathbf{c}') p_t(\mathbf{c}')$$

- allowed moves (exclude identity):

  Φ: set of allowed moves $F: G_F[\star] \to G[\star]$
  $G_F[\star]$: those $\mathbf{c} \in G[\star]$ on which F can act

  all moves are auto-invertible: $(\forall F \in \Phi): \; F^2 = \mathbb{I}$
  Φ is ergodic on $G[\star]$

- graph mobility $n(\mathbf{c})$:

$$n(\mathbf{c}) = \sum_{F \in \Phi} I_F(\mathbf{c}), \qquad I_F(\mathbf{c}) = \begin{cases} 1 & \text{if } \mathbf{c} \in G_F[\star] \\ 0 & \text{if } \mathbf{c} \notin G_F[\star] \end{cases}$$

Constrained Markovian graph dynamics

need to study graph dynamics more systematically ...

- constraints:
  $G[\star] \subseteq G$: all $\mathbf{c} \in G$ that satisfy constraints $\star$
- stochastic graph dynamics as a Markov chain,
  transition probabilities $W(\mathbf{c}|\mathbf{c}')$

$$\forall \mathbf{c} \in G[\star]: \qquad p_{t+1}(\mathbf{c}) = \sum_{\mathbf{c}' \in G[\star]} W(\mathbf{c}|\mathbf{c}') p_t(\mathbf{c}')$$

- allowed moves (exclude identity):

  Φ: set of allowed moves $F: G_F[\star] \to G[\star]$
  $G_F[\star]$: those $\mathbf{c} \in G[\star]$ on which F can act

  all moves are auto-invertible: $(\forall F \in \Phi): \; F^2 = \mathbb{I}$
  Φ is ergodic on $G[\star]$

- graph mobility $n(\mathbf{c})$:

$$n(\mathbf{c}) = \sum_{F \in \Phi} I_F(\mathbf{c}), \qquad I_F(\mathbf{c}) = \begin{cases} 1 & \text{if } \mathbf{c} \in G_F[\star] \\ 0 & \text{if } \mathbf{c} \notin G_F[\star] \end{cases}$$

Constrained Markovian graph dynamics

need to study graph dynamics more systematically ...

- constraints:
  $G[\star] \subseteq G$: all $\mathbf{c} \in G$ that satisfy constraints $\star$

- stochastic graph dynamics as a Markov chain,
  transition probabilities $W(\mathbf{c}|\mathbf{c}')$

$$\forall \mathbf{c} \in G[\star] : \qquad p_{t+1}(\mathbf{c}) = \sum_{\mathbf{c}' \in G[\star]} W(\mathbf{c}|\mathbf{c}')p_t(\mathbf{c}')$$

- allowed moves (exclude identity):

  Φ: set of allowed moves $F : G_F[\star] \to G[\star]$
  $G_F[\star]$: those $\mathbf{c} \in G[\star]$ on which F can act

  all moves are auto-invertible: $(\forall F \in \Phi) : \ F^2 = \mathbb{I}$
  Φ is ergodic on $G[\star]$

- graph mobility $n(\mathbf{c})$:

$$n(\mathbf{c}) = \sum_{F \in \Phi} I_F(\mathbf{c}), \qquad I_F(\mathbf{c}) = \begin{cases} 1 & \textit{if } \mathbf{c} \in G_F[\star] \\ 0 & \textit{if } \mathbf{c} \notin G_F[\star] \end{cases}$$

MCMC objective

construct transition probs $W(\mathbf{c}|\mathbf{c}')$, based on moves $F \in \Phi$, such that process converges to $p_\infty(\mathbf{c}) = Z^{-1} e^{-H(\mathbf{c})}$ on $G[\star]$

- structure:

$$W(\mathbf{c}|\mathbf{c}') = \sum_{F\in\Phi} q(F|\mathbf{c}')\Big[\delta_{\mathbf{c},F\mathbf{c}'} A(F\mathbf{c}'|\mathbf{c}') + \delta_{\mathbf{c},\mathbf{c}'}[1 - A(F\mathbf{c}'|\mathbf{c}')]\Big]$$

$q(F|\mathbf{c})$: *move proposal probability*

$A(\mathbf{c}|\mathbf{c}')$: *move acceptance probability*

- detailed balance condition:

$$(\forall F \in \Phi)(\forall \mathbf{c} \in G[\star]) : \qquad q(F|\mathbf{c})A(F\mathbf{c}|\mathbf{c})\mathrm{e}^{-H(\mathbf{c})} = q(F|F\mathbf{c})A(\mathbf{c}|F\mathbf{c})\mathrm{e}^{-H(F\mathbf{c})}$$

allowed F equally probable:
$q(F|\mathbf{c}) = I_F(\mathbf{c})/n(\mathbf{c})$

$$(\forall F \in \Phi)(\forall \mathbf{c} \in G_F[\star]) : \qquad \frac{1}{n(\mathbf{c})} A(F\mathbf{c}|\mathbf{c})\mathrm{e}^{-H(\mathbf{c})} = \frac{1}{n(F\mathbf{c})} A(\mathbf{c}|F\mathbf{c})\mathrm{e}^{-H(F\mathbf{c})}$$

MCMC objective

construct transition probs $W(\mathbf{c}|\mathbf{c}')$, based on moves $F \in \Phi$, such that process converges to $p_\infty(\mathbf{c}) = Z^{-1} e^{-H(\mathbf{c})}$ on $G[\star]$

- structure:

$$W(\mathbf{c}|\mathbf{c}') = \sum_{F\in\Phi} q(F|\mathbf{c}')\Big[\delta_{\mathbf{c},F\mathbf{c}'} A(F\mathbf{c}'|\mathbf{c}') + \delta_{\mathbf{c},\mathbf{c}'}[1 - A(F\mathbf{c}'|\mathbf{c}')]\Big]$$

$q(F|\mathbf{c})$: *move proposal probability*

$A(\mathbf{c}|\mathbf{c}')$: *move acceptance probability*

- detailed balance condition:

$$(\forall F\in\Phi)(\forall \mathbf{c}\in G[\star]): \qquad q(F|\mathbf{c})A(F\mathbf{c}|\mathbf{c})\mathrm{e}^{-H(\mathbf{c})} = q(F|F\mathbf{c})A(\mathbf{c}|F\mathbf{c})\mathrm{e}^{-H(F\mathbf{c})}$$

allowed F equally probable:
$q(F|\mathbf{c}) = I_F(\mathbf{c})/n(\mathbf{c})$

$$(\forall F\in\Phi)(\forall \mathbf{c}\in G_F[\star]): \qquad \frac{1}{n(\mathbf{c})}A(F\mathbf{c}|\mathbf{c})\mathrm{e}^{-H(\mathbf{c})} = \frac{1}{n(F\mathbf{c})}A(\mathbf{c}|F\mathbf{c})\mathrm{e}^{-H(F\mathbf{c})}$$

MCMC objective

construct transition probs $W(\mathbf{c}|\mathbf{c}')$, based on moves $F \in \Phi$, such that process converges to $p_\infty(\mathbf{c}) = Z^{-1} e^{-H(\mathbf{c})}$ on $G[\star]$

- structure:

$$W(\mathbf{c}|\mathbf{c}') = \sum_{F\in\Phi} q(F|\mathbf{c}') \Big[\delta_{\mathbf{c},F\mathbf{c}'} A(F\mathbf{c}'|\mathbf{c}') + \delta_{\mathbf{c},\mathbf{c}'} [1 - A(F\mathbf{c}'|\mathbf{c}')]\Big]$$

$q(F|\mathbf{c})$: *move proposal probability*

$A(\mathbf{c}|\mathbf{c}')$: *move acceptance probability*

- detailed balance condition:

$$(\forall F \in \Phi)(\forall \mathbf{c} \in G[\star]) : \qquad q(F|\mathbf{c})A(F\mathbf{c}|\mathbf{c})\mathrm{e}^{-H(\mathbf{c})} = q(F|F\mathbf{c})A(\mathbf{c}|F\mathbf{c})\mathrm{e}^{-H(F\mathbf{c})}$$

allowed F equally probable:
$q(F|\mathbf{c}) = I_F(\mathbf{c})/n(\mathbf{c})$

$$(\forall F \in \Phi)(\forall \mathbf{c} \in G_F[\star]) : \qquad \frac{1}{n(\mathbf{c})} A(F\mathbf{c}|\mathbf{c})\mathrm{e}^{-H(\mathbf{c})} = \frac{1}{n(F\mathbf{c})} A(\mathbf{c}|F\mathbf{c})\mathrm{e}^{-H(F\mathbf{c})}$$

canonical Markov chain

ergodic auto-invertible moves $F \in \Phi$,
convergence to $p_\infty(\mathbf{c}) = Z^{-1} e^{-H(\mathbf{c})}$ on $G[\star]$
for acceptance probabilities

$$A(\mathbf{c}|\mathbf{c}') = \frac{n(\mathbf{c}')\mathrm{e}^{-\frac{1}{2}[H(\mathbf{c})-H(\mathbf{c}')]}}{n(\mathbf{c}')e^{-\frac{1}{2}[H(\mathbf{c})-H(\mathbf{c}')]} + n(\mathbf{c})\mathrm{e}^{\frac{1}{2}[H(\mathbf{c})-H(\mathbf{c}')]}}$$

corollary:

naive edge-swapping, $A(\mathbf{c}|\mathbf{c}') = const$,
corresponds to $H(\mathbf{c}) = -\log n(\mathbf{c})$,
so would give

$$\textit{sampling bias}: \quad p_\infty(\mathbf{c}) = \frac{n(\mathbf{c})}{\sum_{\mathbf{c}' \in G[\star]} n(\mathbf{c}')}$$

canonical Markov chain

ergodic auto-invertible moves $F \in \Phi$,
convergence to $p_\infty(\mathbf{c}) = Z^{-1} e^{-H(\mathbf{c})}$ on $G[\star]$
for acceptance probabilities

$$A(\mathbf{c}|\mathbf{c}') = \frac{n(\mathbf{c}')\mathrm{e}^{-\frac{1}{2}[H(\mathbf{c})-H(\mathbf{c}')]}}{n(\mathbf{c}')e^{-\frac{1}{2}[H(\mathbf{c})-H(\mathbf{c}')]} + n(\mathbf{c})\mathrm{e}^{\frac{1}{2}[H(\mathbf{c})-H(\mathbf{c}')]}}$$

corollary:

naive edge-swapping, $A(\mathbf{c}|\mathbf{c}') = const$,
corresponds to $H(\mathbf{c}) = -\log n(\mathbf{c})$,
so would give

$$\textit{sampling bias}: \quad p_\infty(\mathbf{c}) = \frac{n(\mathbf{c})}{\sum_{\mathbf{c}' \in G[\star]} n(\mathbf{c}')}$$

Master equation representation of the process

- soln of Markov chain: $p_n(\mathbf{c})$, $n = 0, 1, 2, \ldots$

  continuous time process, $p_t(\mathbf{c})$, $t \in [0, \infty\rangle$
  via *random durations* of MC steps

  $\pi_m(t) = (t/\tau)^m \mathrm{e}^{-t/\tau}/m!$:
  prob that at time t precisely m MC steps have been made

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} p_t(\mathbf{c}) = \sum_{\mathbf{c}' \in G[\star]} W(\mathbf{c}|\mathbf{c}') p_t(\mathbf{c}') - p_t(\mathbf{c})$$

- work out details,
  using $\Delta_F U(\mathbf{c}) = U(F\mathbf{c}) - U(\mathbf{c})$

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} p_t(\mathbf{c}) = \sum_{F \in \Phi} I_F(\mathbf{c}) \left\{ \frac{w_F^+(\mathbf{c})}{n(F\mathbf{c})} p_t(F\mathbf{c}) - \frac{w_F^-(\mathbf{c})}{n(\mathbf{c})} p_t(\mathbf{c}) \right\}$$

$$w_F^{\pm}(\mathbf{c}) = \frac{1}{2} \pm \frac{1}{2} \tanh \left[\frac{1}{2} \Delta_F [H(\mathbf{c}) + \log n(\mathbf{c})] \right]$$

- Convergence:

$$let \quad \hat{p}(\mathbf{c}) = Z^{-1}\mathrm{e}^{-H(\mathbf{c})}, \qquad F(t) = \sum_{\mathbf{c}\in G[\star]} p_t(\mathbf{c}) \log[p_t(\mathbf{c})/\hat{p}(\mathbf{c})]$$

$F(t)$ is Lyapunov function $\quad \forall t \geq 0: \quad F(t) \geq 0, \quad \frac{\mathrm{d}}{\mathrm{d}t}F(t) \leq 0$

- Proof (standard):

use detailed balance

$$\frac{\mathrm{d}}{\mathrm{d}t}F(t) = -\frac{1}{2\tau} \sum_{\mathbf{c},\mathbf{c}'\in G[\star]} W(\mathbf{c}|\mathbf{c}')\mathrm{e}^{-H(\mathbf{c}')} \Big[[H(\mathbf{c}) + \log p_t(\mathbf{c})] - [H(\mathbf{c}') + \log p_t(\mathbf{c}')]\Big] \times \Big[\mathrm{e}^{H(\mathbf{c}) + \log p_t(\mathbf{c})} - \mathrm{e}^{H(\mathbf{c}') + \log p_t(\mathbf{c}')}\Big] \leq 0$$

$(e^x - e^y)(x - y) \geq 0$, equality only if $x = y$

- stationarity: $\frac{\mathrm{d}}{\mathrm{d}t}F(t) = 0$,
write $p(\mathbf{c}) = \chi(\mathbf{c})\mathrm{e}^{-H(\mathbf{c})}$,

$$(\forall \mathbf{c}, \mathbf{c}' \in G[\star]): \quad W(\mathbf{c}|\mathbf{c}') = 0 \quad \text{or} \quad \chi(\mathbf{c}) = \chi(\mathbf{c}')$$

$$ergodic \;\Rightarrow\; \chi(\mathbf{c}) = const \;\Rightarrow\; p(\mathbf{c}) = Z^{-1}\mathrm{e}^{-H(\mathbf{c})} = \hat{p}(\mathbf{c})$$

- Convergence:

$$\textit{let}\quad \hat{p}(\mathbf{c}) = Z^{-1}\mathrm{e}^{-H(\mathbf{c})}, \qquad F(t) = \sum_{\mathbf{c}\in G[\star]} p_t(\mathbf{c})\log[p_t(\mathbf{c})/\hat{p}(\mathbf{c})]$$

$F(t)$ is Lyapunov function $\quad \forall t \geq 0: \quad F(t) \geq 0, \quad \frac{\mathrm{d}}{\mathrm{d}t}F(t) \leq 0$

- Proof (standard):

use detailed balance

$$\frac{\mathrm{d}}{\mathrm{d}t}F(t) = -\frac{1}{2\tau}\sum_{\mathbf{c},\mathbf{c}'\in G[\star]} W(\mathbf{c}|\mathbf{c}')\mathrm{e}^{-H(\mathbf{c}')}\Big[[H(\mathbf{c})+\log p_t(\mathbf{c})] - [H(\mathbf{c}')+\log p_t(\mathbf{c}')]\Big]$$
$$\times\Big[\mathrm{e}^{H(\mathbf{c})+\log p_t(\mathbf{c})} - \mathrm{e}^{H(\mathbf{c}')+\log p_t(\mathbf{c}')}\Big] \leq 0$$

$(e^x - e^y)(x-y) \geq 0$, equality only if $x = y$

- stationarity: $\frac{\mathrm{d}}{\mathrm{d}t}F(t) = 0$,
write $p(\mathbf{c}) = \chi(\mathbf{c})\mathrm{e}^{-H(\mathbf{c})}$,

$$(\forall \mathbf{c},\mathbf{c}' \in G[\star]): \quad W(\mathbf{c}|\mathbf{c}') = 0 \;\; \text{or} \;\; \chi(\mathbf{c}) = \chi(\mathbf{c}')$$
$$\textit{ergodic} \;\Rightarrow\; \chi(\mathbf{c}) = \textit{const} \;\Rightarrow\; p(\mathbf{c}) = Z^{-1}\mathrm{e}^{-H(\mathbf{c})} = \hat{p}(\mathbf{c})$$

- Convergence:

$$let \quad \hat{p}(\mathbf{c}) = Z^{-1}\mathrm{e}^{-H(\mathbf{c})}, \qquad F(t) = \sum_{\mathbf{c}\in G[\star]} p_t(\mathbf{c}) \log[p_t(\mathbf{c})/\hat{p}(\mathbf{c})]$$

$F(t)$ is Lyapunov function $\quad \forall t \geq 0: \quad F(t) \geq 0, \quad \frac{\mathrm{d}}{\mathrm{d}t}F(t) \leq 0$

- Proof (standard):

use detailed balance

$$\frac{\mathrm{d}}{\mathrm{d}t}F(t) = -\frac{1}{2\tau} \sum_{\mathbf{c},\mathbf{c}'\in G[\star]} W(\mathbf{c}|\mathbf{c}')\mathrm{e}^{-H(\mathbf{c}')} \Big[[H(\mathbf{c})+\log p_t(\mathbf{c})] - [H(\mathbf{c}')+\log p_t(\mathbf{c}')]\Big] \times \Big[\mathrm{e}^{H(\mathbf{c})+\log p_t(\mathbf{c})} - \mathrm{e}^{H(\mathbf{c}')+\log p_t(\mathbf{c}')}\Big] \leq 0$$

$(e^x - e^y)(x-y) \geq 0$, equality only if $x = y$

- stationarity: $\frac{\mathrm{d}}{\mathrm{d}t}F(t) = 0$,
write $p(\mathbf{c}) = \chi(\mathbf{c})\mathrm{e}^{-H(\mathbf{c})}$,

$$(\forall \mathbf{c},\mathbf{c}' \in G[\star]): \quad W(\mathbf{c}|\mathbf{c}') = 0 \;\; \text{or} \;\; \chi(\mathbf{c}) = \chi(\mathbf{c}')$$

$$ergodic \;\Rightarrow\; \chi(\mathbf{c}) = const \;\Rightarrow\; p(\mathbf{c}) = Z^{-1}\mathrm{e}^{-H(\mathbf{c})} = \hat{p}(\mathbf{c})$$

Degree-constrained dynamics of nondirected graphs

- constraints: imposed degrees, so graph set is $G[\mathbf{k}]$

  ergodic set Φ of admissible moves:
  edge swaps $F : G_F[\mathbf{k}] \to G[\mathbf{k}]$

  $Q = \{(i,j,k,\ell) \in \{1,\ldots,N\}^4 |\ i<j<k<\ell\}$, ordered node quadruplets

  possible edge swaps to act on (i,j,k,ℓ):

- group into pairs (I,IV), (II,V), and (III,VI)
  auto-invertible swaps: $F_{ijk\ell;\alpha}$, with $i<j<k<\ell$ and $\alpha \in \{1,2,3\}$

  $I_{ijk\ell;\alpha}(\mathbf{c}) = 1$:

$$F_{ijk\ell;\alpha}(\mathbf{c})_{qr} = 1 - c_{qr} \quad \text{for } (q,r) \in S_{ijk\ell;\alpha}$$
$$F_{ijk\ell;\alpha}(\mathbf{c})_{qr} = c_{qr} \quad \text{for } (q,r) \notin S_{ijk\ell;\alpha}$$

$$S_{ijk\ell;1} = \{(i,j),(k,\ell),(i,\ell),(j,k)\}, \quad S_{ijk\ell;2} = \{(i,j),(k,\ell),(i,k),(j,\ell)\}$$
$$S_{ijk\ell;3} = \{(i,k),(j,\ell),(i,\ell),(j,k)\}$$

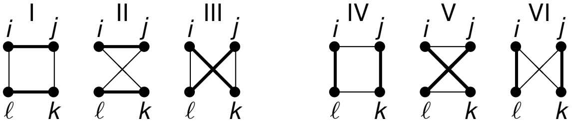
Degree-constrained dynamics of nondirected graphs

- constraints: imposed degrees, so graph set is $G[\mathbf{k}]$

  ergodic set Φ of admissible moves:
  edge swaps $F: G_F[\mathbf{k}] \to G[\mathbf{k}]$

  $Q = \{(i,j,k,\ell) \in \{1,\ldots,N\}^4 |\ i<j<k<\ell\}$, ordered node quadruplets

  possible edge swaps to act on (i,j,k,ℓ):



- group into pairs (I,IV), (II,V), and (III,VI)
  auto-invertible swaps: $F_{ijk\ell;\alpha}$, with $i<j<k<\ell$ and $\alpha \in \{1,2,3\}$

  $I_{ijk\ell;\alpha}(\mathbf{c}) = 1$:

$$F_{ijk\ell;\alpha}(\mathbf{c})_{qr} = 1 - c_{qr} \quad \text{for } (q,r) \in S_{ijk\ell;\alpha}$$
$$F_{ijk\ell;\alpha}(\mathbf{c})_{qr} = c_{qr} \quad \text{for } (q,r) \notin S_{ijk\ell;\alpha}$$

$$S_{ijk\ell;1} = \{(i,j),(k,\ell),(i,\ell),(j,k)\}, \quad S_{ijk\ell;2} = \{(i,j),(k,\ell),(i,k),(j,\ell)\}$$
$$S_{ijk\ell;3} = \{(i,k),(j,\ell),(i,\ell),(j,k)\}$$

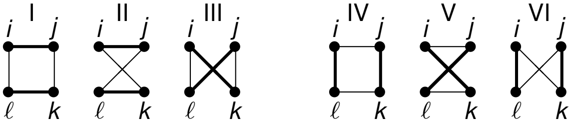
Degree-constrained dynamics of nondirected graphs

- constraints: imposed degrees, so graph set is $G[\mathbf{k}]$

  ergodic set Φ of admissible moves:
  edge swaps $F : G_F[\mathbf{k}] \to G[\mathbf{k}]$

  $Q = \{(i,j,k,\ell) \in \{1,\ldots,N\}^4 |\ i<j<k<\ell\}$, ordered node quadruplets

  possible edge swaps
  to act on (i,j,k,ℓ):



- group into pairs (I,IV), (II,V), and (III,VI)
  auto-invertible swaps: $F_{ijk\ell;\alpha}$, with $i<j<k<\ell$ and $\alpha \in \{1,2,3\}$

  $I_{ijk\ell;\alpha}(\mathbf{c}) = 1$:

$$
\begin{aligned}
F_{ijk\ell;\alpha}(\mathbf{c})_{qr} &= 1 - c_{qr} \quad \text{for } (q,r) \in \mathcal{S}_{ijk\ell;\alpha} \\
F_{ijk\ell;\alpha}(\mathbf{c})_{qr} &= c_{qr} \quad \text{for } (q,r) \notin \mathcal{S}_{ijk\ell;\alpha}
\end{aligned}
$$

$$
\mathcal{S}_{ijk\ell;1} = \{(i,j),(k,\ell),(i,\ell),(j,k)\}, \quad \mathcal{S}_{ijk\ell;2} = \{(i,j),(k,\ell),(i,k),(j,\ell)\}
$$
$$
\mathcal{S}_{ijk\ell;3} = \{(i,k),(j,\ell),(i,\ell),(j,k)\}
$$

to implement the Markov chain,
need to calculate graph mobility **analytically**

$n(\mathbf{c}) = \sum_{i<j<k<\ell}^{N} \sum_{\alpha=1}^{3} I_{ijk\ell;\alpha}(\mathbf{c})$ ...

$$
\begin{aligned}
I_{ijk\ell;1}(\mathbf{c}) &= c_{ij}c_{k\ell}(1-c_{i\ell})(1-c_{jk}) + (1-c_{ij})(1-c_{k\ell})c_{i\ell}c_{jk} \\
I_{ijk\ell;2}(\mathbf{c}) &= c_{ij}c_{k\ell}(1-c_{ik})(1-c_{j\ell}) + (1-c_{ij})(1-c_{k\ell})c_{ik}c_{j\ell} \\
I_{ijk\ell;3}(\mathbf{c}) &= c_{ik}c_{j\ell}(1-c_{i\ell})(1-c_{jk}) + (1-c_{ik})(1-c_{j\ell})c_{i\ell}c_{jk}
\end{aligned}
$$

combinatorial problem easily solved:

$$
n(\mathbf{c}) = \underbrace{\frac{1}{4}N^2\langle k\rangle^2 + \frac{1}{4}N\langle k\rangle - \frac{1}{2}N\langle k^2\rangle}_{invariant} + \underbrace{\frac{1}{4}\mathrm{Tr}(\mathbf{c}^4) + \frac{1}{2}\mathrm{Tr}(\mathbf{c}^3) - \frac{1}{2}\sum_{ij} k_i c_{ij} k_j}_{state\ dependent}
$$

– state-dependent part can be ignored if $\langle k^2\rangle k_{\max}/\langle k\rangle^2 \ll N$
(in which case naive edge swapping is harmless)

– computational feasibility:
calculate change $\Delta_{ijk\ell;\alpha} n(\mathbf{c})$ for each executed move $F_{ijk\ell;\alpha}$

to implement the Markov chain,
need to calculate graph mobility **analytically**

$n(\mathbf{c}) = \sum_{i<j<k<\ell}^{N} \sum_{\alpha=1}^{3} I_{ijk\ell;\alpha}(\mathbf{c})$ ...

$$
\begin{aligned}
I_{ijk\ell;1}(\mathbf{c}) &= c_{ij}c_{k\ell}(1-c_{i\ell})(1-c_{jk}) + (1-c_{ij})(1-c_{k\ell})c_{i\ell}c_{jk} \\
I_{ijk\ell;2}(\mathbf{c}) &= c_{ij}c_{k\ell}(1-c_{ik})(1-c_{j\ell}) + (1-c_{ij})(1-c_{k\ell})c_{ik}c_{j\ell} \\
I_{ijk\ell;3}(\mathbf{c}) &= c_{ik}c_{j\ell}(1-c_{i\ell})(1-c_{jk}) + (1-c_{ik})(1-c_{j\ell})c_{i\ell}c_{jk}
\end{aligned}
$$

combinatorial problem easily solved:

$$
n(\mathbf{c}) = \underbrace{\frac{1}{4}N^2\langle k\rangle^2 + \frac{1}{4}N\langle k\rangle - \frac{1}{2}N\langle k^2\rangle}_{invariant} + \underbrace{\frac{1}{4}\mathrm{Tr}(\mathbf{c}^4) + \frac{1}{2}\mathrm{Tr}(\mathbf{c}^3) - \frac{1}{2}\sum_{ij} k_i c_{ij} k_j}_{state\ dependent}
$$

– state-dependent part can be ignored if $\langle k^2\rangle k_{\max}/\langle k\rangle^2 \ll N$
(in which case naive edge swapping is harmless)

– computational feasibility:
calculate change $\Delta_{ijk\ell;\alpha} n(\mathbf{c})$ for each executed move $F_{ijk\ell;\alpha}$

to implement the Markov chain,
need to calculate graph mobility **analytically**

$n(\mathbf{c}) = \sum_{i<j<k<\ell}^{N} \sum_{\alpha=1}^{3} I_{ijk\ell;\alpha}(\mathbf{c})$ ...

$$\begin{aligned}
I_{ijk\ell;1}(\mathbf{c}) &= c_{ij}c_{k\ell}(1-c_{i\ell})(1-c_{jk}) + (1-c_{ij})(1-c_{k\ell})c_{i\ell}c_{jk} \\
I_{ijk\ell;2}(\mathbf{c}) &= c_{ij}c_{k\ell}(1-c_{ik})(1-c_{j\ell}) + (1-c_{ij})(1-c_{k\ell})c_{ik}c_{j\ell} \\
I_{ijk\ell;3}(\mathbf{c}) &= c_{ik}c_{j\ell}(1-c_{i\ell})(1-c_{jk}) + (1-c_{ik})(1-c_{j\ell})c_{i\ell}c_{jk}
\end{aligned}$$

combinatorial problem easily solved:

$$n(\mathbf{c}) = \underbrace{\frac{1}{4}N^2\langle k\rangle^2 + \frac{1}{4}N\langle k\rangle - \frac{1}{2}N\langle k^2\rangle}_{invariant} + \underbrace{\frac{1}{4}\mathrm{Tr}(\mathbf{c}^4) + \frac{1}{2}\mathrm{Tr}(\mathbf{c}^3) - \frac{1}{2}\sum_{ij} k_i c_{ij} k_j}_{state\ dependent}$$

– state-dependent part can be ignored if $\langle k^2\rangle k_{\max}/\langle k\rangle^2 \ll N$
(in which case naive edge swapping is harmless)

– computational feasibility:
calculate change $\Delta_{ijk\ell;\alpha} n(\mathbf{c})$ for each executed move $F_{ijk\ell;\alpha}$

Example

many possible moves *only <u>one</u> move ...*

$N = 100$

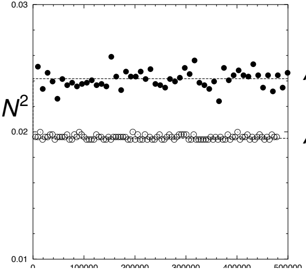
naive versus correct acceptance probabilities

predictions:

$p(\mathbf{c}) = \textit{constant}$:
$\overline{n(\mathbf{c})}/N^2 \approx 0.0195$

$p(\mathbf{c}) = n(\mathbf{c})/Z$:
$\overline{n(\mathbf{c})}/N^2 \approx 0.0242$

$\overline{n(\mathbf{c})}/N^2$

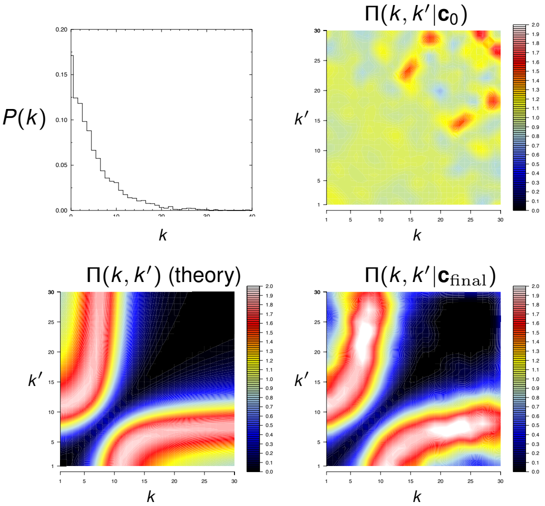


executed moves

$A(\mathbf{c}|\mathbf{c}') = 1$

$A(\mathbf{c}|\mathbf{c}') = [1 + \frac{n(\mathbf{c})}{n(\mathbf{c}')}]^{-1}$

Example



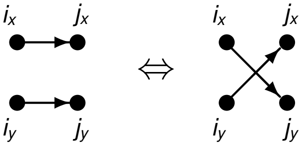
$N = 4000$,
$\overline{k} = 5$

$$\Pi(k, k') = \frac{(k - k')^2}{[\beta_1 - \beta_2 k + \beta_3 k^2][\beta_1 - \beta_2 k' + \beta_3 k'^2]}$$

Degree-constrained dynamics of directed graphs

- constraints: imposed in-out degrees, so graph set is $G[\mathbf{k}^{\rm in},\mathbf{k}^{\rm out}]$

  set Φ of admissible moves:
  directed edge swaps $\quad F: G_F[\mathbf{k}^{\rm in},\mathbf{k}^{\rm out}] \to G[\mathbf{k}^{\rm in},\mathbf{k}^{\rm out}]$



- auto-invertible edge-swaps:
  Let $\Lambda = \{(i,j)\in N^2 |\ c_{ij}=1\}$

$$I_{(i_x,j_x),(i_y,j_y);\square} = \begin{cases} 1 & \text{if } (i_x,j_x),(i_y,j_y)\in\Lambda \text{ and } (i_x,j_y),(i_y,j_x)\notin\Lambda \\ 0 & \text{otherwise}\end{cases}$$

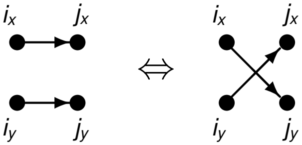
If $I_{(i_x,j_x),(i_y,j_y);\square}=1$:

$$\begin{aligned} F_{(i_x,j_x),(i_y,j_y);\square}(\mathbf{c})_{ij} &= 1-c_{ij} && \text{if } i\in\{i_x,i_y\} \text{ and } j\in\{j_x,j_y\} \\ F_{(i_x,j_x),(i_y,j_y);\square}(\mathbf{c})_{ij} &= c_{ij} && \text{otherwise}\end{aligned}$$

Degree-constrained dynamics of directed graphs

- constraints: imposed in-out degrees, so graph set is $G[\mathbf{k}^{\rm in},\mathbf{k}^{\rm out}]$

  set Φ of admissible moves:
  directed edge swaps $\quad F : G_F[\mathbf{k}^{\rm in},\mathbf{k}^{\rm out}] \to G[\mathbf{k}^{\rm in},\mathbf{k}^{\rm out}]$



- auto-invertible edge-swaps:
  Let $\Lambda = \{(i,j) \in N^2 |\ c_{ij} = 1\}$

$$I_{(i_x,j_x),(i_y,j_y);\Box} = \begin{cases} 1 & \textit{if } (i_x,j_x),(i_y,j_y) \in \Lambda \text{ and } (i_x,j_y),(i_y,j_x) \notin \Lambda \\ 0 & \text{otherwise} \end{cases}$$

If $I_{(i_x,j_x),(i_y,j_y);\Box} = 1$:

$$\begin{aligned} F_{(i_x,j_x),(i_y,j_y);\Box}(\mathbf{c})_{ij} &= 1 - c_{ij} \qquad \text{if } i \in \{i_x,i_y\} \text{ and } j \in \{j_x,j_y\} \\ F_{(i_x,j_x),(i_y,j_y);\Box}(\mathbf{c})_{ij} &= c_{ij} \qquad \text{otherwise} \end{aligned}$$

difference with nondirected graphs:

edge swaps *no longer ergodic* (Rao, 1996)
(unless self-interactions are allowed)

further move type required
to restore ergodicity:
3-loop reversal

$$I_{(i_x,j_x),(i_y,j_y);\triangle} = \begin{cases} 1 & \text{if } (i_x,j_x),(i_y,j_y),(j_y,i_x) \in \Lambda \text{ and } x_j = y_i \\ & \text{and } (j_x,i_x),(j_y,i_y),(i_x,j_y) \notin \Lambda \\ 0 & \text{otherwise} \end{cases}$$

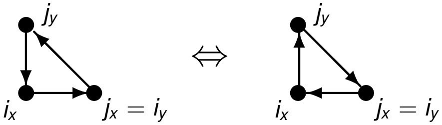
$$F_{(i_x,j_x),(i_y,j_y);\triangle}(\mathbf{c})_{ij} = 1 - c_{ij} \quad \text{for } (i,j) \in \mathcal{S}_{i_x,j_x,j_y}$$
$$F_{(i_x,j_x),(i_y,j_y);\triangle}(\mathbf{c})_{ij} = c_{ij} \quad \text{for } (i,j) \notin \mathcal{S}_{i_x,j_x,j_y}$$

$$\mathcal{S}_{abc} = \{(a,b),(b,c),(c,a),(b,a),(c,b),(a,c)\}$$

difference with nondirected graphs:

edge swaps *no longer ergodic* (Rao, 1996)
(unless self-interactions are allowed)

further move type required
to restore ergodicity:
3-loop reversal



$$I_{(i_x,j_x),(i_y,j_y);\triangle} = \begin{cases} 1 & \textit{if } (i_x,j_x),(i_y,j_y),(j_y,i_x) \in \Lambda \textit{ and } x_j = y_i \\ & \textit{and } (j_x,i_x),(j_y,i_y),(i_x,j_y) \notin \Lambda \\ 0 & \text{otherwise} \end{cases}$$

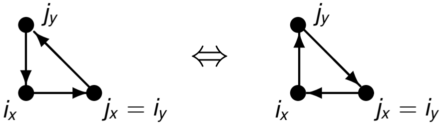
$$\begin{aligned} F_{(i_x,j_x),(i_y,j_y);\triangle}(\mathbf{c})_{ij} &= 1 - c_{ij} \quad \textit{for } (i,j) \in \mathcal{S}_{i_x,j_x,j_y} \\ F_{(i_x,j_x),(i_y,j_y);\triangle}(\mathbf{c})_{ij} &= c_{ij} \qquad \textit{for } (i,j) \notin \mathcal{S}_{i_x,j_x,j_y} \end{aligned}$$

$$\mathcal{S}_{abc} = \{(a,b),(b,c),(c,a),(b,a),(c,b),(a,c)\}$$

difference with nondirected graphs:

edge swaps *no longer ergodic* (Rao, 1996)
(unless self-interactions are allowed)

further move type required
to restore ergodicity:
3-loop reversal



$$I_{(i_x,j_x),(i_y,j_y);\triangle} = \begin{cases} 1 & \textit{if } (i_x,j_x),(i_y,j_y),(j_y,i_x) \in \Lambda \;\; \textit{and} \;\; x_j = y_i \\ & \textit{and} \;\; (j_x,i_x),(j_y,i_y),(i_x,j_y) \notin \Lambda \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{aligned} F_{(i_x,j_x),(i_y,j_y);\triangle}(\mathbf{c})_{ij} &= 1 - c_{ij} \quad \textit{for } (i,j) \in \mathcal{S}_{i_x,j_x,j_y} \\ F_{(i_x,j_x),(i_y,j_y);\triangle}(\mathbf{c})_{ij} &= c_{ij} \qquad\quad \textit{for } (i,j) \notin \mathcal{S}_{i_x,j_x,j_y} \end{aligned}$$

$$\mathcal{S}_{abc} = \{(a,b),(b,c),(c,a),(b,a),(c,b),(a,c)\}$$

to implement the Markov chain,
need to calculate graph mobility **analytically**:

$$
\begin{aligned}
n(\mathbf{c}) &= n_\square(\mathbf{c}) + n_\triangle(\mathbf{c}) \\
&= \sum_{(i_x,j_x),(i_y,j_y)\in\Lambda} I_{(i_x,j_x),(i_y,j_y);\square} + \sum_{(i_x,j_x),(i_y,j_y)\in\Lambda} I_{(i_x,j_x),(i_y,j_y);\triangle}
\end{aligned}
$$

$$
\begin{aligned}
I_{(i_x,j_x),(i_y,j_y);\square} &= c_{i_x,j_x} c_{i_y,j_y} (1-c_{i_x,j_y})(1-c_{i_y,j_x}) \\
I_{(i_x,j_x),(i_y,j_y);\triangle} &= \delta_{x_j,y_i}\; c_{i_x,j_x} c_{i_y,j_y} c_{j_y,i_x} (1-c_{j_x,i_x})(1-c_{j_y,i_y})(1-c_{i_x,j_y})
\end{aligned}
$$

combinatorial problem easily solved:

$$
n_\square(\mathbf{c}) = \underbrace{\frac{1}{2}N^2\langle k\rangle^2 - \sum_j k_j^{\rm in}k_j^{\rm out}}_{invariant} + \underbrace{\frac{1}{2}\mathrm{Tr}(\mathbf{c}^2) + \frac{1}{2}\mathrm{Tr}(\mathbf{c}^\dagger\mathbf{c}\mathbf{c}^\dagger\mathbf{c}) + \mathrm{Tr}(\mathbf{c}^2\mathbf{c}^\dagger) - \sum_{ij} k_i^{\rm in} c_{ij} k_j^{\rm out}}_{state\ dependent}
$$

$$
n_\triangle(\mathbf{c}) = \underbrace{\frac{1}{3}\mathrm{Tr}(\mathbf{c}^3) - \mathrm{Tr}(\hat{\mathbf{c}}\mathbf{c}^2) + \mathrm{Tr}(\hat{\mathbf{c}}^2\mathbf{c}) - \frac{1}{3}\mathrm{Tr}(\hat{\mathbf{c}}^3)}_{state\ dependent}
$$

with: $(\mathbf{c}^\dagger)_{ij} = c_{ji}$, $\hat{\mathbf{c}}_{ij} = c_{ij}c_{ji}$

to implement the Markov chain,
need to calculate graph mobility **analytically**:

$$
\begin{aligned}
n(\mathbf{c}) &= n_{\square}(\mathbf{c}) + n_{\triangle}(\mathbf{c}) \\
&= \sum_{(i_x,j_x),(i_y,j_y)\in\Lambda} I_{(i_x,j_x),(i_y,j_y);\square} + \sum_{(i_x,j_x),(i_y,j_y)\in\Lambda} I_{(i_x,j_x),(i_y,j_y);\triangle}
\end{aligned}
$$

$$
\begin{aligned}
I_{(i_x,j_x),(i_y,j_y);\square} &= c_{i_x,j_x} c_{i_y,j_y} (1-c_{i_x,j_y})(1-c_{i_y,j_x}) \\
I_{(i_x,j_x),(i_y,j_y);\triangle} &= \delta_{x_j,y_i}\, c_{i_x,j_x} c_{i_y,j_y} c_{j_y,i_x} (1-c_{j_x,i_x})(1-c_{j_y,i_y})(1-c_{i_x,j_y})
\end{aligned}
$$

combinatorial problem easily solved:

$$
n_{\square}(\mathbf{c}) = \underbrace{\frac{1}{2}N^2\langle k\rangle^2 - \sum_j k_j^{\rm in} k_j^{\rm out}}_{invariant} + \underbrace{\frac{1}{2}\mathrm{Tr}(\mathbf{c}^2) + \frac{1}{2}\mathrm{Tr}(\mathbf{c}^\dagger\mathbf{c}\mathbf{c}^\dagger\mathbf{c}) + \mathrm{Tr}(\mathbf{c}^2\mathbf{c}^\dagger) - \sum_{ij} k_i^{\rm in} c_{ij} k_j^{\rm out}}_{state\ dependent}
$$

$$
n_{\triangle}(\mathbf{c}) = \underbrace{\frac{1}{3}\mathrm{Tr}(\mathbf{c}^3) - \mathrm{Tr}(\hat{\mathbf{c}}\mathbf{c}^2) + \mathrm{Tr}(\hat{\mathbf{c}}^2\mathbf{c}) - \frac{1}{3}\mathrm{Tr}(\hat{\mathbf{c}}^3)}_{state\ dependent}
$$

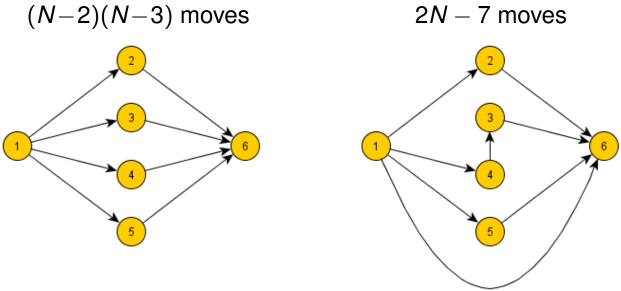
with: $(\mathbf{c}^\dagger)_{ij} = c_{ji}$, $\hat{\mathbf{c}}_{ij} = c_{ij}c_{ji}$

Example

$(k_1^{\rm in}, k_1^{\rm out}) = (0, N-2)$

$i = 2 \ldots N-1$:
$(k_i^{\rm in}, k_i^{\rm out}) = (1, 1)$

$(k_N^{\rm in}, k_N^{\rm out}) = (N-2, 0)$



predicted values versus
equilibrated dynamics for $\langle n(\mathbf{c})\rangle / N^2$:

	prediction for $p(\mathbf{c}) = const$	dynamics with $A(\mathbf{c}\vert\mathbf{c}') = 1$	dynamics with $A(\mathbf{c}\vert\mathbf{c}') = [1 + \frac{n(\mathbf{c})}{n(\mathbf{c}')}]^{-1}$
$N = 17$:	27.87	33.59	27.87
$N = 27$:	47.92	58.32	47.95

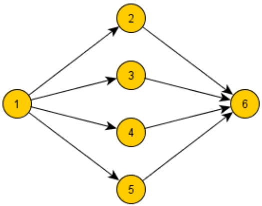
Example

$(k_1^{\rm in}, k_1^{\rm out}) = (0, N-2)$

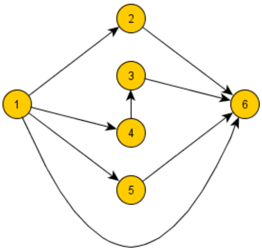
$i = 2 \ldots N-1$:
$(k_i^{\rm in}, k_i^{\rm out}) = (1, 1)$

$(k_N^{\rm in}, k_N^{\rm out}) = (N-2, 0)$

$(N-2)(N-3)$ moves



$2N-7$ moves



predicted values versus
equilibrated dynamics for $\langle n(\mathbf{c})\rangle / N^2$:

	prediction for $p(\mathbf{c}) = const$	dynamics with $A(\mathbf{c}\vert\mathbf{c}') = 1$	dynamics with $A(\mathbf{c}\vert\mathbf{c}') = [1 + \frac{n(\mathbf{c})}{n(\mathbf{c}')}]^{-1}$
$N = 17$:	27.87	33.59	27.87
$N = 27$:	47.92	58.32	47.95

Summary

- standard 'matching' end 'edge swapping' algorithms,
  for generating graphs **c** with prescribed degrees, are *both biased*
- need exact method for generating random graphs **c**
  with prescribed degrees and prescribed sampling probabilities $p(\mathbf{c})$
- exact degree constrained Markovian graph dynamics can be defined,
  guaranteed to evolve to any prescribed measure $p(\mathbf{c})$
- process requires nontrivial state acceptance probabilities,
  that involve the mobilities $n(\mathbf{c})$ of states
- nondirected graph: edge swaps only
  directed graphs: edge swaps and 3-cycle reversals
- mobilities can be calculated exactly
- theory worked out, implemented and tested
  for nondirected and directed graphs

Summary

- standard 'matching' end 'edge swapping' algorithms, for generating graphs **c** with prescribed degrees, are *both biased*
- need exact method for generating random graphs **c** with prescribed degrees and prescribed sampling probabilities $p(\mathbf{c})$

- exact degree constrained Markovian graph dynamics can be defined, guaranteed to evolve to any prescribed measure $p(\mathbf{c})$
- process requires nontrivial state acceptance probabilities, that involve the mobilities $n(\mathbf{c})$ of states
- nondirected graph: edge swaps only
  directed graphs: edge swaps and 3-cycle reversals

- mobilities can be calculated exactly
- theory worked out, implemented and tested for nondirected and directed graphs

Summary

- standard ‘matching’ end ‘edge swapping’ algorithms,
  for generating graphs **c** with prescribed degrees, are *both biased*
- need exact method for generating random graphs **c**
  with prescribed degrees and prescribed sampling probabilities $p(\mathbf{c})$

- exact degree constrained Markovian graph dynamics can be defined,
  guaranteed to evolve to any prescribed measure $p(\mathbf{c})$
- process requires nontrivial state acceptance probabilities,
  that involve the mobilities $n(\mathbf{c})$ of states
- nondirected graph: edge swaps only
  directed graphs: edge swaps and 3-cycle reversals

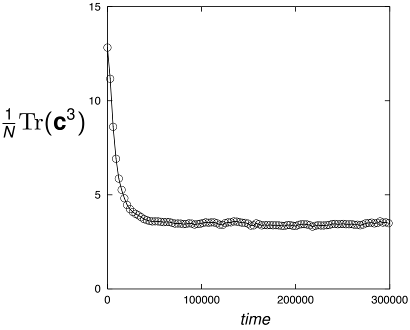
- mobilities can be calculated exactly
- theory worked out, implemented and tested
  for nondirected and directed graphs

Current and future work

Development of theory for tailored graph ensembles characterised by statistics of **loops** ...

(in addition to degrees and degree correlations)

- calculate Shannon entropies
- dynamics (constrained by degrees and loops)
- graph ensembles defined by eigenvalue spectrum
- loops versus closed paths (see talk by Clara Gracio)



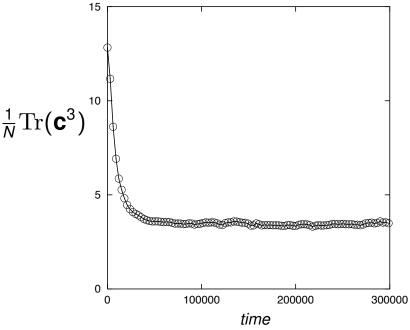
protein interaction networks

Current and future work

Development of theory for tailored graph ensembles characterised by statistics of **loops** ...

(in addition to degrees and degree correlations)

- calculate Shannon entropies
- dynamics (constrained by degrees and loops)
- graph ensembles defined by eigenvalue spectrum
- loops versus closed paths (see talk by Clara Gracio)



protein interaction networks

Thanks to

theory:

- Kate Roberts
- Alessia Annibale
- Ginestra Bianconi
- Andrea De Martino
- Conrad Pérez-Vicente
- Clara Gracio

bio-informatics applications:

- Luis Fernandes
- Franca Fraternali
- Jens Kleinjung
- Thomas Schlitt

grants:

BBSRC bioscience for the future

EPSRC Engineering and Physical Sciences Research Council

papers - theory

G Bianconi, ACC Coolen and CJ Perez-Vicente
Phys. Rev. E78 2008, 016114

ACC Coolen, A De Martino and A Annibale
J. Stat. Phys. 136 2009, 1035-1067

A Annibale, ACC Coolen, LP Fernandes, F Fraternali and J Kleinjung
J. Phys. A: Math. Theor. 42 2009, 485001

A Annibale, ACC Coolen and G Bianconi
J. Phys. A: Math. Theor. 43 2010, 395001

ES Roberts, T Schlitt and ACC Coolen
J. Phys. A: Math. Theor. 44 2011, 275002

papers - applications in biology

LP Fernandes, A Annibale, J Kleinjung, ACC Coolen and F Fraternali
PLoS ONE 5 2010, e12083

ACC Coolen, F Fraternali, A Annibale, LP Fernandes, and J Kleinjung
Handbook of Statistical Systems Biology (in press, 2011)

A Annibale and ACC Coolen
Interface Focus (Royal Society, in press 2011), preprint ArXiv:1106.0236