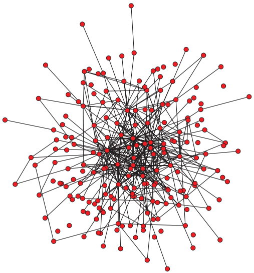
Tailoring, counting, and generating structured random graphs

A detour ...

ACC Coolen

Dept of Mathematics and Randall Division
King's College London



Outline

Outline

1. Background and origin of the problem
   - Protein interaction and gene regulation networks
   - Nonequilibrium stat mech of cellular signalling
   - Quantify graph topology beyond degrees
2. Analysis of tailored random graph ensembles
   - Canonical graph ensemble with controlled $\{p, \Pi\}$
   - Statistical mechanical analysis - entropy and beyond
   - Application to real PIN data
3. Generating tailored random graphs numerically
   - Degree-constrained 'shuffling'
   - Formulation as a stochastic process
   - Graph mobility
4. Ongoing work
5. Summary

Outline

1. Background and origin of the problem
   - Protein interaction and gene regulation networks
   - Nonequilibrium stat mech of cellular signalling
   - Quantify graph topology beyond degrees
2. Analysis of tailored random graph ensembles
   - Canonical graph ensemble with controlled $\{p, \Pi\}$
   - Statistical mechanical analysis - entropy and beyond
   - Application to real PIN data
3. Generating tailored random graphs numerically
   - Degree-constrained 'shuffling'
   - Formulation as a stochastic process
   - Graph mobility
4. Ongoing work
5. Summary

Outline

1. Background and origin of the problem
   - Protein interaction and gene regulation networks
   - Nonequilibrium stat mech of cellular signalling
   - Quantify graph topology beyond degrees
2. Analysis of tailored random graph ensembles
   - Canonical graph ensemble with controlled $\{p, \Pi\}$
   - Statistical mechanical analysis - entropy and beyond
   - Application to real PIN data
3. Generating tailored random graphs numerically
   - Degree-constrained 'shuffling'
   - Formulation as a stochastic process
   - Graph mobility
4. Ongoing work
5. Summary

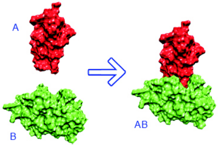
Outline

Outline

1. Background and origin of the problem
   - Protein interaction and gene regulation networks
   - Nonequilibrium stat mech of cellular signalling
   - Quantify graph topology beyond degrees
2. Analysis of tailored random graph ensembles
   - Canonical graph ensemble with controlled $\{p, \Pi\}$
   - Statistical mechanical analysis - entropy and beyond
   - Application to real PIN data
3. Generating tailored random graphs numerically
   - Degree-constrained 'shuffling'
   - Formulation as a stochastic process
   - Graph mobility
4. Ongoing work
5. Summary

- **proteins**:

  hetero-polymers that interact via (temporary) *complexes*



reaction eqns:

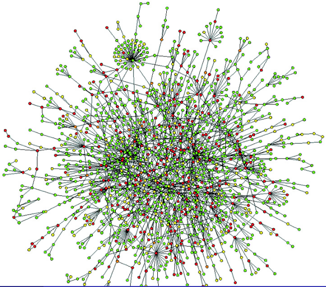
$$\frac{\mathrm{d}}{\mathrm{d}t}x_i^\alpha = \sum_j c_{ij}\sum_\beta [k_{ij}^{\alpha\beta-}x_{ij} - k_{ij}^{\alpha\beta+}x_i^\alpha x_j^\beta] + \theta_i^\alpha - \gamma_i^\alpha x_i^\alpha$$

$$\frac{\mathrm{d}}{\mathrm{d}t}x_{ij} = c_{ij}\sum_{\alpha\beta}[k_{ij}^{\alpha\beta+}x_i^\alpha x_j^\beta - k_{ij}^{\alpha\beta-}x_{ij}]$$

- **protein interaction network**:

  nodes: proteins $i, j = 1 \ldots N$
  links: $c_{ij} = c_{ji} = 1$ if $(i \asymp j)$ possible
  $c_{ij} = c_{ji} = 0$ otherwise

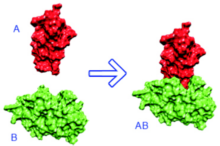
  $N \sim 10^4$, about 7 links/node



Introduction - protein interaction networks (PIN)

- **proteins**:

  hetero-polymers that interact
  via (temporary) *complexes*



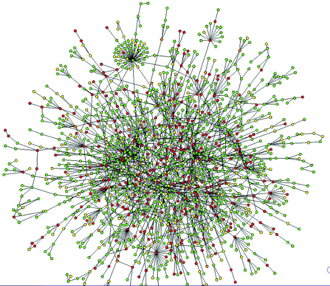
  reaction eqns:

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} x_i^\alpha &= \sum_j c_{ij} \sum_\beta [k_{ij}^{\alpha\beta-} x_{ij} - k_{ij}^{\alpha\beta+} x_i^\alpha x_j^\beta] + \theta_i^\alpha - \gamma_i^\alpha x_i^\alpha \\
\frac{\mathrm{d}}{\mathrm{d}t} x_{ij} &= c_{ij} \sum_{\alpha\beta} [k_{ij}^{\alpha\beta+} x_i^\alpha x_j^\beta - k_{ij}^{\alpha\beta-} x_{ij}]
\end{aligned}
$$

- **protein interaction network**:

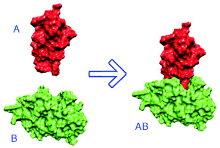
  nodes: proteins $i, j = 1 \ldots N$
  links: $c_{ij} = c_{ji} = 1$ if $(i \asymp j)$ possible
  $c_{ij} = c_{ji} = 0$ otherwise

  $N \sim 10^4$, about 7 links/node



- **proteins**:

  hetero-polymers that interact via (temporary) *complexes*



  reaction eqns:

$$\frac{\mathrm{d}}{\mathrm{d}t}x_i^\alpha = \sum_j c_{ij}\sum_\beta [k_{ij}^{\alpha\beta-}x_{ij} - k_{ij}^{\alpha\beta+}x_i^\alpha x_j^\beta] + \theta_i^\alpha - \gamma_i^\alpha x_i^\alpha$$

$$\frac{\mathrm{d}}{\mathrm{d}t}x_{ij} = c_{ij}\sum_{\alpha\beta}[k_{ij}^{\alpha\beta+}x_i^\alpha x_j^\beta - k_{ij}^{\alpha\beta-}x_{ij}]$$

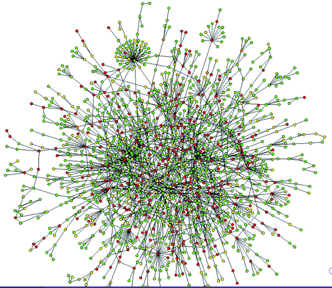
- **protein interaction network**:

  nodes: proteins $i, j = 1 \ldots N$

  links: $c_{ij} = c_{ji} = 1$ if ($i \asymp j$) possible

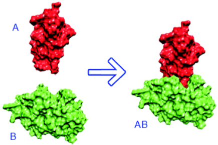
  $c_{ij} = c_{ji} = 0$ otherwise

  $N \sim 10^4$, about 7 links/node



- **proteins**:

  hetero-polymers that interact via (temporary) *complexes*



  reaction eqns:

$$\frac{\mathrm{d}}{\mathrm{d}t}x_i^\alpha = \sum_j c_{ij} \sum_\beta [k_{ij}^{\alpha\beta-} x_{ij} - k_{ij}^{\alpha\beta+} x_i^\alpha x_j^\beta] + \theta_i^\alpha - \gamma_i^\alpha x_i^\alpha$$

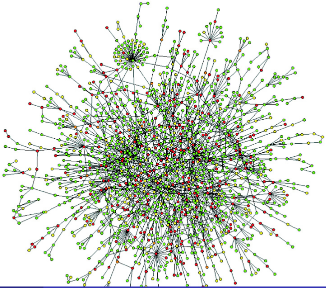
$$\frac{\mathrm{d}}{\mathrm{d}t}x_{ij} = c_{ij} \sum_{\alpha\beta} [k_{ij}^{\alpha\beta+} x_i^\alpha x_j^\beta - k_{ij}^{\alpha\beta-} x_{ij}]$$

- **protein interaction network**:

  nodes: proteins $i, j = 1 \ldots N$

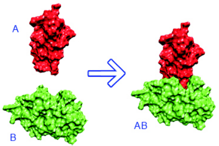
  links: $c_{ij} = c_{ji} = 1$ if $(i \asymp j)$ possible
  $c_{ij} = c_{ji} = 0$ otherwise

  $N \sim 10^4$, about 7 links/node



- **proteins**:

  hetero-polymers that interact
  via (temporary) *complexes*



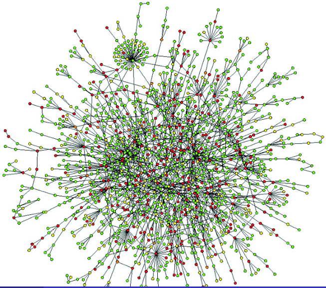
  reaction eqns:

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} x_i^\alpha &= \sum_j c_{ij} \sum_\beta [k_{ij}^{\alpha\beta-} x_{ij} - k_{ij}^{\alpha\beta+} x_i^\alpha x_j^\beta] + \theta_i^\alpha - \gamma_i^\alpha x_i^\alpha \\
\frac{\mathrm{d}}{\mathrm{d}t} x_{ij} &= c_{ij} \sum_{\alpha\beta} [k_{ij}^{\alpha\beta+} x_i^\alpha x_j^\beta - k_{ij}^{\alpha\beta-} x_{ij}]
\end{aligned}
$$

- **protein interaction network**:

  nodes: proteins $i, j = 1 \ldots N$
  links: $c_{ij} = c_{ji} = 1$ if $(i \asymp j)$ possible
  $c_{ij} = c_{ji} = 0$ otherwise

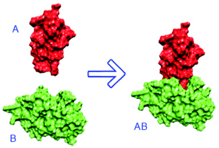
  $N \sim 10^4$, about 7 links/node



Introduction - protein interaction networks (PIN)

- **proteins**:

  hetero-polymers that interact via (temporary) *complexes*



  reaction eqns:

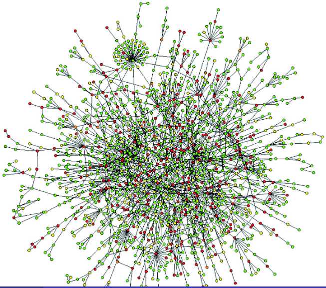
$$\frac{\mathrm{d}}{\mathrm{d}t}x_i^\alpha = \sum_j c_{ij} \sum_\beta [k_{ij}^{\alpha\beta-} x_{ij} - k_{ij}^{\alpha\beta+} x_i^\alpha x_j^\beta] + \theta_i^\alpha - \gamma_i^\alpha x_i^\alpha$$

$$\frac{\mathrm{d}}{\mathrm{d}t}x_{ij} = c_{ij} \sum_{\alpha\beta} [k_{ij}^{\alpha\beta+} x_i^\alpha x_j^\beta - k_{ij}^{\alpha\beta-} x_{ij}]$$

- **protein interaction network**:

  nodes: proteins $i, j = 1 \ldots N$
  links: $c_{ij} = c_{ji} = 1$ if $(i \asymp j)$ possible
  $c_{ij} = c_{ji} = 0$ otherwise

  $N \sim 10^4$, about 7 links/node



Introduction - gene regulation networks (GRN)

- gene **expression level** σ_i:

  degree to which gene is 'switched on'

  Boolean models:
  $\sigma_i \in \{-1, 1\}$,
  discrete time

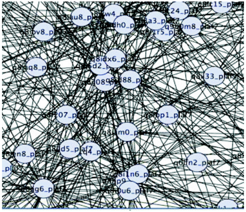
  $$\sigma_i(t+1) = F_i(\sigma_1(t), \ldots, \sigma_N(t))$$

  $$F_i(\sigma_1, \ldots, \sigma_N) = \theta[h_i(\sigma_1, \ldots, \sigma_N) + noise]$$
  $$h_i(\sigma_1, \ldots, \sigma_N) = \sum_j J_{ij}\sigma_j + \sum_{jk} J_{ijk}\sigma_j\sigma_k + \ldots$$

- **gene regulation network** (directed!)

  $$c_{ij} = 1 \quad \text{if } \sigma_j \text{ appears in } F_i(\sigma_1, \ldots, \sigma_N)$$
  $$c_{ij} = 0 \quad \text{otherwise}$$

  $N \sim 5000$, about 5 links/node



Introduction - gene regulation networks (GRN)

- gene **expression level** σ_i:

  degree to which gene is 'switched on'

  Boolean models:
  $\sigma_i \in \{-1, 1\}$,
  discrete time

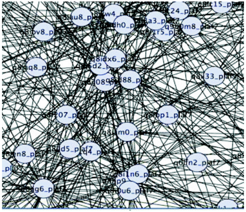
  $$\sigma_i(t+1) = F_i(\sigma_1(t), \ldots, \sigma_N(t))$$

  $$F_i(\sigma_1, \ldots, \sigma_N) = \theta[h_i(\sigma_1, \ldots, \sigma_N) + noise]$$
  $$h_i(\sigma_1, \ldots, \sigma_N) = \sum_j J_{ij}\sigma_j + \sum_{jk} J_{ijk}\sigma_j\sigma_k + \ldots$$

- **gene regulation network**
  (directed!)

  $$c_{ij} = 1 \quad \text{if } \sigma_j \text{ appears in } F_i(\sigma_1, \ldots, \sigma_N)$$
  $$c_{ij} = 0 \quad \text{otherwise}$$

  $N \sim 5000$, about 5 links/node



Introduction - gene regulation networks (GRN)

- gene **expression level** σ_i:

  degree to which gene is 'switched on'

  Boolean models:
  $\sigma_i \in \{-1, 1\}$,
  discrete time

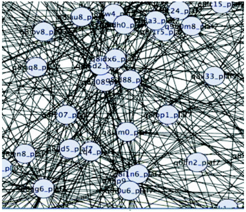
$$\sigma_i(t+1) = F_i(\sigma_1(t), \ldots, \sigma_N(t))$$

$$F_i(\sigma_1, \ldots, \sigma_N) = \theta[h_i(\sigma_1, \ldots, \sigma_N) + noise]$$
$$h_i(\sigma_1, \ldots, \sigma_N) = \sum_j J_{ij}\sigma_j + \sum_{jk} J_{ijk}\sigma_j\sigma_k + \ldots$$

- **gene regulation network** (directed!)

$$c_{ij} = 1 \quad \text{if } \sigma_j \text{ appears in } F_i(\sigma_1, \ldots, \sigma_N)$$
$$c_{ij} = 0 \quad \text{otherwise}$$

$N \sim 5000$, about 5 links/node



Introduction - gene regulation networks (GRN)

- gene **expression level** σ_i:

  degree to which gene is 'switched on'

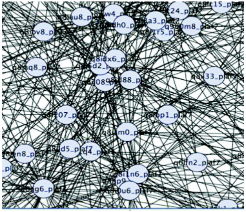
  Boolean models:
  $\sigma_i \in \{-1, 1\}$,
  discrete time

$$\sigma_i(t+1) = F_i(\sigma_1(t), \ldots, \sigma_N(t))$$

$$F_i(\sigma_1, \ldots, \sigma_N) = \theta[h_i(\sigma_1, \ldots, \sigma_N) + noise]$$
$$h_i(\sigma_1, \ldots, \sigma_N) = \sum_j J_{ij}\sigma_j + \sum_{jk} J_{ijk}\sigma_j\sigma_k + \ldots$$

- **gene regulation network**
  (directed!)

$$c_{ij} = 1 \quad \text{if } \sigma_j \text{ appears in } F_i(\sigma_1, \ldots, \sigma_N)$$
$$c_{ij} = 0 \quad \text{otherwise}$$



$N \sim 5000$, about 5 links/node

Introduction - gene regulation networks (GRN)

- gene **expression level** σ_i:

  degree to which gene is 'switched on'

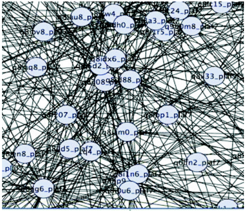
  Boolean models:
  $\sigma_i \in \{-1, 1\}$,
  discrete time

  $$\sigma_i(t+1) = F_i(\sigma_1(t), \ldots, \sigma_N(t))$$

  $$F_i(\sigma_1, \ldots, \sigma_N) = \theta[h_i(\sigma_1, \ldots, \sigma_N) + noise]$$
  $$h_i(\sigma_1, \ldots, \sigma_N) = \sum_j J_{ij}\sigma_j + \sum_{jk} J_{ijk}\sigma_j\sigma_k + \ldots$$

- **gene regulation network**
  (directed!)

  $$c_{ij} = 1 \quad \text{if } \sigma_j \text{ appears in } F_i(\sigma_1, \ldots, \sigma_N)$$
  $$c_{ij} = 0 \quad \text{otherwise}$$



$N \sim 5000$, about 5 links/node

Introduction - gene regulation networks (GRN)

- gene **expression level** σ_i:

  degree to which gene is 'switched on'

  Boolean models:
  $\sigma_i \in \{-1, 1\}$,
  discrete time

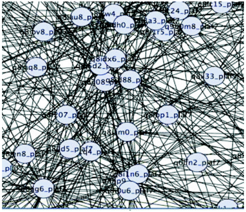
$$\sigma_i(t+1) = F_i(\sigma_1(t), \ldots, \sigma_N(t))$$

$$F_i(\sigma_1, \ldots, \sigma_N) = \theta[h_i(\sigma_1, \ldots, \sigma_N) + noise]$$
$$h_i(\sigma_1, \ldots, \sigma_N) = \sum_j J_{ij}\sigma_j + \sum_{jk} J_{ijk}\sigma_j\sigma_k + \ldots$$

- **gene regulation network**
  (directed!)

$$c_{ij} = 1 \quad \text{if } \sigma_j \text{ appears in } F_i(\sigma_1, \ldots, \sigma_N)$$
$$c_{ij} = 0 \quad \text{otherwise}$$

$N \sim 5000$, about 5 links/node



Outline

1. Background and origin of the problem
   - Protein interaction and gene regulation networks
   - Nonequilibrium stat mech of cellular signalling
   - Quantify graph topology beyond degrees
2. Analysis of tailored random graph ensembles
   - Canonical graph ensemble with controlled $\{p, \Pi\}$
   - Statistical mechanical analysis - entropy and beyond
   - Application to real PIN data
3. Generating tailored random graphs numerically
   - Degree-constrained 'shuffling'
   - Formulation as a stochastic process
   - Graph mobility
4. Ongoing work
5. Summary

Nonequil stat mech of cellular signalling

stochastic processes, **no detailed balance**
disorder: random graph, **finitely connected**

exact tool: generating functional analysis (GFA)

- **Gene regulation:**

$$P_{t+1}(\boldsymbol{\sigma}) = \sum_{\boldsymbol{\sigma}'} \Big(\prod_i \frac{e^{\beta\sigma_i h_i(\boldsymbol{\sigma}',t)}}{2\cosh[\beta h_i(\boldsymbol{\sigma}',t)]} \Big) P_t(\boldsymbol{\sigma}'), \qquad h_i(\boldsymbol{\sigma},t) = \sum_j c_{ij} J_{ij} \sigma_j + \theta_i(t)$$

generating functional:

$$\overline{Z[\psi]} = \overline{\left\langle \mathrm{e}^{-\mathrm{i}\sum_i \sum_t \psi_i(t)\sigma_i(t)} \right\rangle} \qquad P[\boldsymbol{\sigma}(0),\cdots,\boldsymbol{\sigma}(t_m)] = p_0[\boldsymbol{\sigma}(0)] \prod_{t<t_m} W_t[\boldsymbol{\sigma}(t+1)|\boldsymbol{\sigma}(t)]$$

random connectivity:

$$p(\mathbf{c}) = \frac{\prod_i \delta_{k_i,\sum_j c_{ij}}}{Z} \Big(\prod_{i<j} \hat{p}(c_{ij}) \hat{p}(c_{ji}|c_{ij}) \Big) \qquad \begin{array}{l} \hat{p}(c_{ij}) = \frac{\overline{k}}{N}\delta_{c_{ij},1} + (1-\frac{\overline{k}}{N})\delta_{c_{ij},0} \\ \hat{p}(c_{ji}|c_{ij}) = \varepsilon\delta_{c_{ji},c_{ij}} + (1-\varepsilon)\hat{p}(c_{ji}) \end{array}$$

Nonequil stat mech of cellular signalling

stochastic processes, **no detailed balance**
disorder: random graph, **finitely connected**

exact tool: generating functional analysis (GFA)

- **Gene regulation:**

$$P_{t+1}(\sigma)=\sum_{\sigma'}\Big(\prod_i \frac{e^{\beta\sigma_i h_i(\sigma',t)}}{2\cosh[\beta h_i(\sigma',t)]}\Big)P_t(\sigma'),\qquad h_i(\sigma,t)=\sum_j c_{ij}J_{ij}\sigma_j+\theta_i(t)$$

generating functional:

$$\overline{Z[\psi]}=\overline{\Big\langle e^{-i\sum_i\sum_t \psi_i(t)\sigma_i(t)}\Big\rangle}\qquad P[\sigma(0),\cdots,\sigma(t_m)]=p_0[\sigma(0)]\prod_{t<t_m}W_t[\sigma(t+1)|\sigma(t)]$$

random connectivity:

$$p(\mathbf{c})=\frac{\prod_i \delta_{k_i,\sum_j c_{ij}}}{Z}\Big(\prod_{i<j}\hat{p}(c_{ij})\hat{p}(c_{ji}|c_{ij})\Big)\qquad \begin{aligned}\hat{p}(c_{ij})&=\frac{\overline{k}}{N}\delta_{c_{ij},1}+(1-\frac{\overline{k}}{N})\delta_{c_{ij},0}\\ \hat{p}(c_{ji}|c_{ij})&=\varepsilon\delta_{c_{ji},c_{ij}}+(1-\varepsilon)\hat{p}(c_{ji})\end{aligned}$$

Nonequil stat mech of cellular signalling

stochastic processes, **no detailed balance**
disorder: random graph, **finitely connected**

exact tool: generating functional analysis (GFA)

- **Gene regulation**:

$$P_{t+1}(\boldsymbol{\sigma}) = \sum_{\boldsymbol{\sigma}'} \Big(\prod_i \frac{e^{\beta\sigma_i h_i(\boldsymbol{\sigma}',t)}}{2\cosh[\beta h_i(\boldsymbol{\sigma}',t)]} \Big) P_t(\boldsymbol{\sigma}'), \qquad h_i(\boldsymbol{\sigma},t) = \sum_j c_{ij} J_{ij} \sigma_j + \theta_i(t)$$

generating functional:

$$\overline{Z[\psi]} = \overline{\Big\langle e^{-\mathrm{i}\sum_i \sum_t \psi_i(t)\sigma_i(t)} \Big\rangle} \qquad P[\boldsymbol{\sigma}(0),\cdots,\boldsymbol{\sigma}(t_m)] = p_0[\boldsymbol{\sigma}(0)] \prod_{t<t_m} W_t[\boldsymbol{\sigma}(t+1)|\boldsymbol{\sigma}(t)]$$

random connectivity:

$$p(\mathbf{c}) = \frac{\prod_i \delta_{k_i,\sum_j c_{ij}}}{Z} \Big(\prod_{i<j} \hat{p}(c_{ij}) \hat{p}(c_{ji}|c_{ij}) \Big) \qquad \begin{aligned} \hat{p}(c_{ij}) &= \frac{\bar{k}}{N}\delta_{c_{ij},1} + (1-\frac{\bar{k}}{N})\delta_{c_{ij},0} \\ \hat{p}(c_{ji}|c_{ij}) &= \varepsilon\delta_{c_{ji},c_{ij}} + (1-\varepsilon)\hat{p}(c_{ji}) \end{aligned}$$

stochastic processes, **no detailed balance**
disorder: random graph, **finitely connected**

exact tool: generating functional analysis (GFA)

- **Gene regulation**:

$$P_{t+1}(\boldsymbol{\sigma}) = \sum_{\boldsymbol{\sigma}'} \Big(\prod_i \frac{e^{\beta\sigma_i h_i(\boldsymbol{\sigma}',t)}}{2\cosh[\beta h_i(\boldsymbol{\sigma}',t)]} \Big) P_t(\boldsymbol{\sigma}'), \qquad h_i(\boldsymbol{\sigma},t) = \sum_j c_{ij} J_{ij} \sigma_j + \theta_i(t)$$

generating functional:

$$\overline{Z[\psi]} = \overline{\left\langle \mathrm{e}^{-\mathrm{i}\sum_i \sum_t \psi_i(t)\sigma_i(t)} \right\rangle} \qquad P[\boldsymbol{\sigma}(0),\cdots,\boldsymbol{\sigma}(t_m)] = p_0[\boldsymbol{\sigma}(0)] \prod_{t<t_m} W_t[\boldsymbol{\sigma}(t+1)|\boldsymbol{\sigma}(t)]$$

random connectivity:

$$p(\mathbf{c}) = \frac{\prod_i \delta_{k_i,\sum_j c_{ij}}}{Z} \Big(\prod_{i<j} \hat{p}(c_{ij}) \hat{p}(c_{ji}|c_{ij}) \Big) \qquad \begin{aligned} \hat{p}(c_{ij}) &= \frac{\overline{k}}{N}\delta_{c_{ij},1} + (1-\frac{\overline{k}}{N})\delta_{c_{ij},0} \\ \hat{p}(c_{ji}|c_{ij}) &= \varepsilon\delta_{c_{ji},c_{ij}} + (1-\varepsilon)\hat{p}(c_{ji}) \end{aligned}$$

Nonequil stat mech of cellular signalling

stochastic processes, **no detailed balance**
disorder: random graph, **finitely connected**

exact tool: generating functional analysis (GFA)

- **Gene regulation**:

$$P_{t+1}(\boldsymbol{\sigma})=\sum_{\boldsymbol{\sigma}'}\Big(\prod_i \frac{e^{\beta\sigma_i h_i(\boldsymbol{\sigma}',t)}}{2\cosh[\beta h_i(\boldsymbol{\sigma}',t)]}\Big)P_t(\boldsymbol{\sigma}'),\qquad h_i(\boldsymbol{\sigma},t)=\sum_j c_{ij}J_{ij}\sigma_j+\theta_i(t)$$

generating functional:

$$\overline{Z[\psi]}=\overline{\Big\langle \mathrm{e}^{-\mathrm{i}\sum_i\sum_t \psi_i(t)\sigma_i(t)}\Big\rangle}\qquad P[\boldsymbol{\sigma}(0),\cdots,\boldsymbol{\sigma}(t_m)]=p_0[\boldsymbol{\sigma}(0)]\prod_{t<t_m}W_t[\boldsymbol{\sigma}(t+1)|\boldsymbol{\sigma}(t)]$$

random connectivity:

$$p(\mathbf{c})=\frac{\prod_i \delta_{k_i,\sum_j c_{ij}}}{Z}\Big(\prod_{i<j}\hat{p}(c_{ij})\hat{p}(c_{ji}|c_{ij})\Big)\qquad \begin{aligned}\hat{p}(c_{ij})&=\tfrac{\overline{k}}{N}\delta_{c_{ij},1}+(1-\tfrac{\overline{k}}{N})\delta_{c_{ij},0}\\ \hat{p}(c_{ji}|c_{ij})&=\varepsilon\delta_{c_{ji},c_{ij}}+(1-\varepsilon)\hat{p}(c_{ji})\end{aligned}$$

standard GFA steps ...

$\overline{Z[\psi]}$ in terms of integrals over all local fields $h_i(t)$ at all times, δ-functions in integral form, average over graphs, identify order parameters, use path integrals to isolate, steepest descent for $N \to \infty$

order parameters:

$$P[\boldsymbol{\sigma}|\boldsymbol{\theta}'] = \frac{1}{N}\sum_i \langle \delta_{\boldsymbol{\sigma},\boldsymbol{\sigma}_i} \rangle \Big|_{\boldsymbol{\theta}_i \to \boldsymbol{\theta}_i + \boldsymbol{\theta}'} \qquad Q[\boldsymbol{\sigma}|\boldsymbol{\theta}'] = \frac{1}{N}\sum_i \langle \delta_{\boldsymbol{\sigma},\boldsymbol{\sigma}_i} \rangle \Big|_{k_i \to k_i - 1, \boldsymbol{\theta}_i \to \boldsymbol{\theta}_i + \boldsymbol{\theta}'}$$

$$\boldsymbol{\sigma} = (\sigma(0), \sigma(1), \ldots)$$

order parameter eqns:

$$Q[\boldsymbol{\sigma}|\boldsymbol{\theta}'] = \sum_k \frac{k}{\langle k \rangle} p(k) \Big(\prod_{\ell<k} \int \mathrm{d}J_\ell \, \tilde{p}(J_\ell) \sum_{\boldsymbol{\sigma}_\ell} \Big[\varepsilon Q[\boldsymbol{\sigma}_\ell | J_\ell \boldsymbol{\sigma}] + (1-\varepsilon) P[\boldsymbol{\sigma}_\ell|\mathbf{0}] \Big] \Big)$$
$$\times \, p_0[\sigma(0)] \prod_{t<t_m} \frac{e^{\beta\sigma(t+1)[\theta(t)+\theta'(t)+\frac{1}{c}\sum_{\ell<k} J_\ell \sigma_\ell(t)]}}{2\cosh(\beta[\theta(t)+\theta'(t)+\sum_{\ell=1}^{k} J_\ell \sigma_\ell(t)])}$$

eqn for $P[\boldsymbol{\sigma}|\boldsymbol{\theta}]$: replace $p(k)k/\langle k \rangle \to p(k)$ in RHS

standard GFA steps ...

$\overline{Z[\psi]}$ in terms of integrals over all local fields $h_i(t)$ at all times,
δ-functions in integral form, average over graphs, identify order parameters,
use path integrals to isolate, steepest descent for $N \to \infty$

order parameters:

$$P[\boldsymbol{\sigma}|\boldsymbol{\theta}'] = \frac{1}{N}\sum_i \langle \delta_{\boldsymbol{\sigma},\boldsymbol{\sigma}_i}\rangle\Big|_{\boldsymbol{\theta}_i \to \boldsymbol{\theta}_i + \boldsymbol{\theta}'} \qquad Q[\boldsymbol{\sigma}|\boldsymbol{\theta}'] = \frac{1}{N}\sum_i \langle \delta_{\boldsymbol{\sigma},\boldsymbol{\sigma}_i}\rangle\Big|_{k_i \to k_i - 1, \boldsymbol{\theta}_i \to \boldsymbol{\theta}_i + \boldsymbol{\theta}'}$$

$\boldsymbol{\sigma} = (\sigma(0), \sigma(1), \ldots)$

order parameter eqns:

$$Q[\boldsymbol{\sigma}|\boldsymbol{\theta}'] = \sum_k \frac{k}{\langle k\rangle} p(k) \Big(\prod_{\ell<k} \int \mathrm{d}J_\ell\, \tilde{p}(J_\ell) \sum_{\boldsymbol{\sigma}_\ell} \Big[\varepsilon Q[\boldsymbol{\sigma}_\ell | J_\ell \boldsymbol{\sigma}] + (1-\varepsilon) P[\boldsymbol{\sigma}_\ell|\mathbf{0}]\Big]\Big)$$
$$\times\, p_0[\sigma(0)] \prod_{t<t_m} \frac{\mathrm{e}^{\beta\sigma(t+1)[\theta(t)+\theta'(t)+\frac{1}{c}\sum_{\ell<k} J_\ell \sigma_\ell(t)]}}{2\cosh(\beta[\theta(t)+\theta'(t)+\sum_{\ell=1}^{k} J_\ell \sigma_\ell(t)])}$$

eqn for $P[\boldsymbol{\sigma}|\boldsymbol{\theta}]$: replace $p(k)k/\langle k\rangle \to p(k)$ in RHS

standard GFA steps ...

$\overline{Z[\psi]}$ in terms of integrals over all local fields $h_i(t)$ at all times,
δ-functions in integral form, average over graphs, identify order parameters,
use path integrals to isolate, steepest descent for $N \to \infty$

order parameters:

$$P[\boldsymbol{\sigma}|\boldsymbol{\theta}'] = \frac{1}{N}\sum_i \langle \delta_{\boldsymbol{\sigma},\boldsymbol{\sigma}_i}\rangle\Big|_{\boldsymbol{\theta}_i\to\boldsymbol{\theta}_i+\boldsymbol{\theta}'} \qquad Q[\boldsymbol{\sigma}|\boldsymbol{\theta}'] = \frac{1}{N}\sum_i \langle \delta_{\boldsymbol{\sigma},\boldsymbol{\sigma}_i}\rangle\Big|_{k_i\to k_i-1,\boldsymbol{\theta}_i\to\boldsymbol{\theta}_i+\boldsymbol{\theta}'}$$

$$\boldsymbol{\sigma} = (\sigma(0), \sigma(1), \ldots)$$

order parameter eqns:

$$Q[\boldsymbol{\sigma}|\boldsymbol{\theta}'] = \sum_k \frac{k}{\langle k\rangle} p(k) \Big(\prod_{\ell<k} \int \mathrm{d}J_\ell\, \tilde{p}(J_\ell) \sum_{\boldsymbol{\sigma}_\ell} \Big[\varepsilon Q[\boldsymbol{\sigma}_\ell | J_\ell \boldsymbol{\sigma}] + (1-\varepsilon) P[\boldsymbol{\sigma}_\ell|\mathbf{0}]\Big]\Big)$$
$$\times\, p_0[\sigma(0)] \prod_{t<t_m} \frac{e^{\beta\sigma(t+1)[\theta(t)+\theta'(t)+\frac{1}{c}\sum_{\ell<k} J_\ell \sigma_\ell(t)]}}{2\cosh(\beta[\theta(t)+\theta'(t)+\sum_{\ell=1}^{k} J_\ell\sigma_\ell(t)])}$$

eqn for $P[\boldsymbol{\sigma}|\boldsymbol{\theta}]$: replace $p(k)k/\langle k\rangle \to p(k)$ in RHS

standard GFA steps ...

$\overline{Z[\psi]}$ in terms of integrals over all local fields $h_i(t)$ at all times, δ-functions in integral form, average over graphs, identify order parameters, use path integrals to isolate, steepest descent for $N \to \infty$

order parameters:

$$P[\boldsymbol{\sigma}|\boldsymbol{\theta}'] = \frac{1}{N}\sum_i \langle \delta_{\boldsymbol{\sigma},\boldsymbol{\sigma}_i}\rangle\Big|_{\boldsymbol{\theta}_i \to \boldsymbol{\theta}_i + \boldsymbol{\theta}'} \qquad Q[\boldsymbol{\sigma}|\boldsymbol{\theta}'] = \frac{1}{N}\sum_i \langle \delta_{\boldsymbol{\sigma},\boldsymbol{\sigma}_i}\rangle\Big|_{k_i \to k_i - 1, \boldsymbol{\theta}_i \to \boldsymbol{\theta}_i + \boldsymbol{\theta}'}$$

$$\boldsymbol{\sigma} = (\sigma(0), \sigma(1), \ldots)$$

order parameter eqns:

$$Q[\boldsymbol{\sigma}|\boldsymbol{\theta}'] = \sum_k \frac{k}{\langle k\rangle} p(k) \Big(\prod_{\ell<k} \int \mathrm{d}J_\ell\, \tilde{p}(J_\ell) \sum_{\boldsymbol{\sigma}_\ell} \Big[\varepsilon Q[\boldsymbol{\sigma}_\ell | J_\ell \boldsymbol{\sigma}] + (1-\varepsilon) P[\boldsymbol{\sigma}_\ell|\mathbf{0}]\Big]\Big)$$
$$\times\, p_0[\sigma(0)] \prod_{t<t_m} \frac{\mathrm{e}^{\beta\sigma(t+1)[\theta(t)+\theta'(t)+\frac{1}{c}\sum_{\ell<k} J_\ell \sigma_\ell(t)]}}{2\cosh(\beta[\theta(t)+\theta'(t)+\sum_{\ell=1}^{k} J_\ell \sigma_\ell(t)])}$$

eqn for $P[\boldsymbol{\sigma}|\boldsymbol{\theta}]$: replace $p(k)k/\langle k\rangle \to p(k)$ in RHS

- **Protein-protein interaction**:

solve eqns for $\{x_{ij}\}$:

$$\frac{\mathrm{d}}{\mathrm{d}t}x_i^\alpha(t) = \sum_j c_{ij}\int \mathrm{d}s \sum_{\rho\lambda} W_{\alpha;\rho\lambda}(t-s|\mathbf{k}_{ij})x_i^\rho(s)x_j^\lambda(s)+\theta_i^\alpha(t)-\gamma_i^\alpha(t)x_i^\alpha(t)$$

effective retarded free-protein interaction

$$W_{\alpha;\rho\lambda}(\tau|\mathbf{k}) = k^{\rho\lambda+}\Big[\sum_\beta k^{\alpha\beta-}\theta[\tau]\mathrm{e}^{-k^-\tau}-\delta_{\alpha\rho}\delta(\tau)\Big]$$

generating functional:

$$Z[\psi] = \int\Big[\prod_{i\alpha t}\mathrm{d}x_i^\alpha(t)\Big]\,\mathrm{e}^{\mathrm{i}\sum_{i\alpha}\int \mathrm{d}t\;\psi_i^\alpha(t)x_i^\alpha(t)}\prod_{i\alpha t}\delta\Big[x_i^\alpha(t+\mathrm{d}t)-x_i^\alpha(t)-F_i^\alpha[t,\{x\}]\mathrm{d}t\Big]$$

random connectivity:

$$p(\mathbf{c}) = \frac{\prod_i \delta_{k_i,\sum_{j\neq i}c_{ij}}}{Z}\prod_i\Big[c_0\delta_{c_{ii},1}+(1-c_0)\delta_{c_{ii},0}\Big]$$

- **Protein-protein interaction**:

  solve eqns for $\{x_{ij}\}$:

$$\frac{\mathrm{d}}{\mathrm{d}t}x_i^\alpha(t) \;=\; \sum_j c_{ij}\int \mathrm{d}s\sum_{\rho\lambda} W_{\alpha;\rho\lambda}(t-s|\mathbf{k}_{ij})x_i^\rho(s)x_j^\lambda(s)+\theta_i^\alpha(t)-\gamma_i^\alpha(t)x_i^\alpha(t)$$

  effective retarded free-protein interaction

$$W_{\alpha;\rho\lambda}(\tau|\mathbf{k}) = k^{\rho\lambda+}\Big[\sum_\beta k^{\alpha\beta-}\theta[\tau]\mathrm{e}^{-k^-\tau}-\delta_{\alpha\rho}\delta(\tau)\Big]$$

  generating functional:

$$Z[\psi] = \int\Big[\prod_{i\alpha t}\mathrm{d}x_i^\alpha(t)\Big]\,\mathrm{e}^{\mathrm{i}\sum_{i\alpha}\int\mathrm{d}t\;\psi_i^\alpha(t)x_i^\alpha(t)}\prod_{i\alpha t}\delta\Big[x_i^\alpha(t+\mathrm{d}t)-x_i^\alpha(t)-F_i^\alpha[t,\{x\}]\mathrm{d}t\Big]$$

  random connectivity:

$$p(\mathbf{c}) = \frac{\prod_i \delta_{k_i,\sum_{j\neq i}c_{ij}}}{Z}\;\prod_i\Big[c_0\delta_{c_{ii},1}+(1-c_0)\delta_{c_{ii},0}\Big]$$

standard GFA steps ...

order parameters:

$$D[\{x\}|\{y\}] = \frac{1}{N}\sum_j \overline{\langle \delta[\{x\}-\{x_j\}]\rangle}\big|_{\theta_j^\alpha(t)\to\theta_j^\alpha(t)+y_\alpha(t)\ \forall\alpha}$$

$$W[\{x\}|\{y\}] = \frac{1}{N}\sum_j \overline{\langle \delta[\{x\}-\{x_j\}]\rangle}\big|_{k_j\to k_j-1,\ \ \theta_j^\alpha(t)\to\theta_j^\alpha(t)+y_\alpha(t)\ \forall\alpha}$$

$\{x\} = \{x_\alpha(0), x_\alpha(1), \ldots\}$

order parameter eqns:

$$\begin{aligned}
W[\{x\}|\{y\}] \;\;=\;\; & \sum_k \frac{p(k)k}{\langle k\rangle}\Big\langle \int \prod_{\ell<k}\Big\{\{\mathrm{d}x_\ell \mathrm{d}y_\ell\}W[\{x_\ell\}|\{y_\ell\}] \\
& \times \prod_{\alpha t}\delta\Big[y_{\ell\alpha}(t) - \int \mathrm{d}\tau \sum_{\rho\lambda} W_{\alpha;\rho\lambda}(\tau|S\mathbf{k}_\ell)x_{\ell\rho}(t-\tau)x_\lambda(t-\tau)\Big]\Big\} \\
& \times \prod_{\alpha t}\delta\Big\{\mathrm{d}x_\alpha(t) - \mathrm{d}t\Big[\theta_\alpha(t)+y_\alpha(t)-\gamma_\alpha(t)x_\alpha(t)+\sum_{\rho\lambda}\int \mathrm{d}\tau\; x_\rho(t-\tau) \\
& \times \Big(sW_{\alpha;\rho\lambda}(\tau|\mathbf{k})x_\lambda(t-\tau)+\sum_{\ell<k}W_{\alpha;\rho\lambda}(\tau|\mathbf{k}_\ell)x_{\ell\lambda}(t-\tau)\Big)\Big]\Big\}\Big\rangle_{\mathbf{k},s;\{\mathbf{k}_\ell\},\{\theta,\gamma|k\}}
\end{aligned}$$

eqn for $D[\{x\}|\{y\}]$: replace $p(k)k/\langle k\rangle \to p(k)$ in RHS

standard GFA steps ...

order parameters:

$$D[\{x\}|\{y\}] = \frac{1}{N}\sum_j \overline{\langle\delta[\{x\}-\{x_j\}]\rangle}\Big|_{\theta_j^\alpha(t)\to\theta_j^\alpha(t)+y_\alpha(t)\ \forall\alpha}$$

$$W[\{x\}|\{y\}] = \frac{1}{N}\sum_j \overline{\langle\delta[\{x\}-\{x_j\}]\rangle}\Big|_{k_j\to k_j-1,\ \ \theta_j^\alpha(t)\to\theta_j^\alpha(t)+y_\alpha(t)\ \forall\alpha}$$

$\{x\} = \{x_\alpha(0), x_\alpha(1), \ldots\}$

order parameter eqns:

$$W[\{x\}|\{y\}] = \sum_k \frac{p(k)k}{\langle k\rangle}\Big\langle \int \prod_{\ell<k}\Big\{\{\mathrm{d}x_\ell \mathrm{d}y_\ell\} W[\{x_\ell\}|\{y_\ell\}] \times \prod_{\alpha t}\delta\Big[y_{\ell\alpha}(t) - \int \mathrm{d}\tau \sum_{\rho\lambda} W_{\alpha;\rho\lambda}(\tau|S\mathbf{k}_\ell)x_{\ell\rho}(t-\tau)x_\lambda(t-\tau)\Big]\Big\}$$
$$\times \prod_{\alpha t}\delta\Big\{\mathrm{d}x_\alpha(t) - \mathrm{d}t\Big[\theta_\alpha(t) + y_\alpha(t) - \gamma_\alpha(t)x_\alpha(t) + \sum_{\rho\lambda}\int \mathrm{d}\tau\ x_\rho(t-\tau)$$
$$\times\Big(sW_{\alpha;\rho\lambda}(\tau|\mathbf{k})x_\lambda(t-\tau) + \sum_{\ell<k} W_{\alpha;\rho\lambda}(\tau|\mathbf{k}_\ell)x_{\ell\lambda}(t-\tau)\Big)\Big]\Big\}\Big\rangle_{\mathbf{k},s;\{\mathbf{k}_\ell\},\{\theta,\gamma|k\}}$$

eqn for $D[\{x\}|\{y\}]$: replace $p(k)k/\langle k\rangle \to p(k)$ in RHS

standard GFA steps ...
order parameters:

$$D[\{x\}|\{y\}] = \frac{1}{N}\sum_j \overline{\langle \delta[\{x\} - \{x_j\}]\rangle}\big|_{\theta_j^\alpha(t)\to\theta_j^\alpha(t)+y_\alpha(t)\ \forall\alpha}$$
$$W[\{x\}|\{y\}] = \frac{1}{N}\sum_j \overline{\langle \delta[\{x\} - \{x_j\}]\rangle}\big|_{k_j\to k_j-1,\ \ \theta_j^\alpha(t)\to\theta_j^\alpha(t)+y_\alpha(t)\ \forall\alpha}$$

$$\{x\} = \{x_\alpha(0), x_\alpha(1), \ldots\}$$

order parameter eqns:

$$\begin{aligned}W[\{x\}|\{y\}] \;=\; & \sum_k \frac{p(k)k}{\langle k\rangle}\Big\langle \int \prod_{\ell<k}\Big\{\{\mathrm{d}x_\ell \mathrm{d}y_\ell\} W[\{x_\ell\}|\{y_\ell\}] \\ & \times \prod_{\alpha t}\delta\Big[y_{\ell\alpha}(t) - \int \mathrm{d}\tau \sum_{\rho\lambda} W_{\alpha;\rho\lambda}(\tau|S\mathbf{k}_\ell)x_{\ell\rho}(t-\tau)x_\lambda(t-\tau)\Big]\Big\} \\ & \times \prod_{\alpha t}\delta\Big\{\mathrm{d}x_\alpha(t) - \mathrm{d}t\Big[\theta_\alpha(t) + y_\alpha(t) - \gamma_\alpha(t)x_\alpha(t) + \sum_{\rho\lambda}\int \mathrm{d}\tau\ x_\rho(t-\tau) \\ & \times\Big(sW_{\alpha;\rho\lambda}(\tau|\mathbf{k})x_\lambda(t-\tau) + \sum_{\ell<k} W_{\alpha;\rho\lambda}(\tau|\mathbf{k}_\ell)x_{\ell\lambda}(t-\tau)\Big)\Big]\Big\}\Big\rangle_{\mathbf{k},s;\{\mathbf{k}_\ell\},\{\theta,\gamma|k\}}\end{aligned}$$

eqn for $D[\{x\}|\{y\}]$: replace $p(k)k/\langle k\rangle \to p(k)$ in RHS

Random graphs based on degree information only?

		impose $\langle k\rangle$	impose $p(k)$	exact soln
1D Ising:		$\langle k\rangle=2$ $T_c\approx 1.82$	$p(k)=\delta_{k,2}$ $T_c=0$	$T_c=0$
2D Ising:		$\langle k\rangle=4$ $T_c\approx 3.92$	$p(k)=\delta_{k,4}$ $T_c\approx 2.89$	$T_c\approx 2.27$
3D Ising:		$\langle k\rangle=6$ $T_c\approx 5.94$	$p(k)=\delta_{k,6}$ $T_c\approx 4.93$	$T_c\approx 4.51?$
'small world':		$\langle k\rangle=2+c$	$p(k>1)=\frac{e^{-c}c^{k-2}}{(k-2)!}$	
	$c=1:$	$T_c\approx 2.89$	$T_c\approx 2.18$	$T_c\approx 2.27$
	$c=2:$	$T_c\approx 3.92$	$T_c\approx 3.40$	$T_c\approx 3.47$

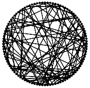
Random graphs based on degree information only?

	impose $\langle k \rangle$	impose $p(k)$	exact soln
1D Ising:	$\langle k \rangle = 2$ $T_c \approx 1.82$	$p(k) = \delta_{k,2}$ $T_c = 0$	$T_c = 0$
2D Ising:	$\langle k \rangle = 4$ $T_c \approx 3.92$	$p(k) = \delta_{k,4}$ $T_c \approx 2.89$	$T_c \approx 2.27$
3D Ising:	$\langle k \rangle = 6$ $T_c \approx 5.94$	$p(k) = \delta_{k,6}$ $T_c \approx 4.93$	$T_c \approx 4.51?$

		impose $\langle k \rangle$	impose $p(k)$	exact soln
'small world':		$\langle k \rangle = 2 + c$	$p(k\!>\!1) = \frac{e^{-c} c^{k-2}}{(k-2)!}$	
	$c = 1:$	$T_c \approx 2.89$	$T_c \approx 2.18$	$T_c \approx 2.27$
	$c = 2:$	$T_c \approx 3.92$	$T_c \approx 3.40$	$T_c \approx 3.47$

Random graphs based on degree information only?

	impose $\langle k \rangle$	impose $p(k)$	exact soln
1D Ising:	$\langle k \rangle = 2$ $T_c \approx 1.82$	$p(k) = \delta_{k,2}$ $T_c = 0$	$T_c = 0$
2D Ising:	$\langle k \rangle = 4$ $T_c \approx 3.92$	$p(k) = \delta_{k,4}$ $T_c \approx 2.89$	$T_c \approx 2.27$
3D Ising:	$\langle k \rangle = 6$ $T_c \approx 5.94$	$p(k) = \delta_{k,6}$ $T_c \approx 4.93$	$T_c \approx 4.51?$
'small world':	$\langle k \rangle = 2 + c$	$p(k > 1) = \frac{e^{-c} c^{k-2}}{(k-2)!}$	
$c = 1$:	$T_c \approx 2.89$	$T_c \approx 2.18$	$T_c \approx 2.27$
$c = 2$:	$T_c \approx 3.92$	$T_c \approx 3.40$	$T_c \approx 3.47$



we wish to code more information into graphs,
for which graph ensembles $p(\mathbf{c})$
is a spin model still **solvable**?

e.g.

$$H(\sigma) = -\sum_{i<j} c_{ij} J_{ij} \sigma_i \sigma_j, \qquad p(\mathbf{c}) = \frac{\delta_{\mathbf{\Omega},\mathbf{\Omega}(\mathbf{c})}}{Z(\mathbf{\Omega})}, \qquad \mathbf{\Omega}(\mathbf{c}) = (\Omega_1(\mathbf{c}), \ldots, \Omega_L(\mathbf{c}))$$

when can we do the average over graphs **analytically**?

$$\text{statics:} \qquad \overline{\mathrm{e}^{-\beta \sum_\alpha H(\boldsymbol{\sigma}^\alpha)}} = \sum_{\mathbf{c}} \frac{\delta_{\mathbf{\Omega},\mathbf{\Omega}(\mathbf{c})}}{Z(\mathbf{\Omega})} \mathrm{e}^{\beta \sum_{i<j} c_{ij} J_{ij} \sum_\alpha \sigma_i^\alpha \sigma_j^\alpha}$$

$$\text{dynamics:} \qquad \overline{\mathrm{e}^{-\mathrm{i} \sum_{it} \hat{h}_i(t) \sum_j c_{ij} J_{ij} \sigma_j(t)}} = \sum_{\mathbf{c}} \frac{\delta_{\mathbf{\Omega},\mathbf{\Omega}(\mathbf{c})}}{Z(\mathbf{\Omega})} \mathrm{e}^{-\mathrm{i} \sum_{i<j} c_{ij} J_{ij} \sum_t [\hat{h}_i(t)\sigma_j(t) + \hat{h}_j(t)\sigma_i(t)]}$$

must do analytically: $\sum_{\mathbf{c}} \delta_{\mathbf{\Omega},\mathbf{\Omega}(\mathbf{c})} \prod_{i<j} \mathrm{e}^{c_{ij} A_{ij}}$

even simpler ... $\sum_{\mathbf{c}} \delta_{\mathbf{\Omega},\mathbf{\Omega}(\mathbf{c})}$

if we can't calculate graph ensemble entropy,
we certainly can't solve the spin models ...

we wish to code more information into graphs,
for which graph ensembles $p(\mathbf{c})$
is a spin model still **solvable**?
e.g.

$$H(\boldsymbol{\sigma}) = -\sum_{i<j} c_{ij} J_{ij} \sigma_i \sigma_j, \qquad p(\mathbf{c}) = \frac{\delta_{\boldsymbol{\Omega},\boldsymbol{\Omega}(\mathbf{c})}}{Z(\boldsymbol{\Omega})}, \qquad \boldsymbol{\Omega}(\mathbf{c}) = (\Omega_1(\mathbf{c}), \ldots, \Omega_L(\mathbf{c}))$$

when can we do the average over graphs **analytically**?

$$\text{statics:} \qquad \overline{\mathrm{e}^{-\beta \sum_\alpha H(\boldsymbol{\sigma}^\alpha)}} = \sum_{\mathbf{c}} \frac{\delta_{\boldsymbol{\Omega},\boldsymbol{\Omega}(\mathbf{c})}}{Z(\boldsymbol{\Omega})} \mathrm{e}^{\beta \sum_{i<j} c_{ij} J_{ij} \sum_\alpha \sigma_i^\alpha \sigma_j^\alpha}$$

$$\text{dynamics:} \qquad \overline{\mathrm{e}^{-\mathrm{i} \sum_{it} \hat{h}_i(t) \sum_j c_{ij} J_{ij} \sigma_j(t)}} = \sum_{\mathbf{c}} \frac{\delta_{\boldsymbol{\Omega},\boldsymbol{\Omega}(\mathbf{c})}}{Z(\boldsymbol{\Omega})} \mathrm{e}^{-\mathrm{i} \sum_{i<j} c_{ij} J_{ij} \sum_t [\hat{h}_i(t)\sigma_j(t) + \hat{h}_j(t)\sigma_i(t)]}$$

must do analytically: $\sum_{\mathbf{c}} \delta_{\boldsymbol{\Omega},\boldsymbol{\Omega}(\mathbf{c})} \prod_{i<j} \mathrm{e}^{c_{ij} A_{ij}}$
even simpler ... $\sum_{\mathbf{c}} \delta_{\boldsymbol{\Omega},\boldsymbol{\Omega}(\mathbf{c})}$

if we can't calculate graph ensemble entropy,
we certainly can't solve the spin models ...

we wish to code more information into graphs,
for which graph ensembles $p(\mathbf{c})$
is a spin model still **solvable**?
e.g.

$$H(\boldsymbol{\sigma}) = -\sum_{i<j} c_{ij} J_{ij} \sigma_i \sigma_j, \qquad p(\mathbf{c}) = \frac{\delta_{\boldsymbol{\Omega},\boldsymbol{\Omega}(\mathbf{c})}}{Z(\boldsymbol{\Omega})}, \qquad \boldsymbol{\Omega}(\mathbf{c}) = (\Omega_1(\mathbf{c}), \ldots, \Omega_L(\mathbf{c}))$$

when can we do the average over graphs **analytically**?

$$\text{statics:} \quad \overline{e^{-\beta \sum_\alpha H(\boldsymbol{\sigma}^\alpha)}} = \sum_{\mathbf{c}} \frac{\delta_{\boldsymbol{\Omega},\boldsymbol{\Omega}(\mathbf{c})}}{Z(\boldsymbol{\Omega})} e^{\beta \sum_{i<j} c_{ij} J_{ij} \sum_\alpha \sigma_i^\alpha \sigma_j^\alpha}$$

$$\text{dynamics:} \quad \overline{e^{-i \sum_{it} \hat{h}_i(t) \sum_j c_{ij} J_{ij} \sigma_j(t)}} = \sum_{\mathbf{c}} \frac{\delta_{\boldsymbol{\Omega},\boldsymbol{\Omega}(\mathbf{c})}}{Z(\boldsymbol{\Omega})} e^{-i \sum_{i<j} c_{ij} J_{ij} \sum_t [\hat{h}_i(t)\sigma_j(t) + \hat{h}_j(t)\sigma_i(t)]}$$

must do analytically: $\sum_{\mathbf{c}} \delta_{\boldsymbol{\Omega},\boldsymbol{\Omega}(\mathbf{c})} \prod_{i<j} e^{c_{ij} A_{ij}}$
even simpler ... $\sum_{\mathbf{c}} \delta_{\boldsymbol{\Omega},\boldsymbol{\Omega}(\mathbf{c})}$

if we can't calculate graph ensemble entropy,
we certainly can't solve the spin models ...

we wish to code more information into graphs,
for which graph ensembles $p(\mathbf{c})$
is a spin model still **solvable**?
e.g.

$$H(\boldsymbol{\sigma}) = -\sum_{i<j} c_{ij} J_{ij} \sigma_i \sigma_j, \qquad p(\mathbf{c}) = \frac{\delta_{\boldsymbol{\Omega},\boldsymbol{\Omega}(\mathbf{c})}}{Z(\boldsymbol{\Omega})}, \qquad \boldsymbol{\Omega}(\mathbf{c}) = (\Omega_1(\mathbf{c}), \ldots, \Omega_L(\mathbf{c}))$$

when can we do the average over graphs **analytically**?

$$\text{statics:} \qquad \overline{e^{-\beta \sum_\alpha H(\boldsymbol{\sigma}^\alpha)}} = \sum_{\mathbf{c}} \frac{\delta_{\boldsymbol{\Omega},\boldsymbol{\Omega}(\mathbf{c})}}{Z(\boldsymbol{\Omega})} e^{\beta \sum_{i<j} c_{ij} J_{ij} \sum_\alpha \sigma_i^\alpha \sigma_j^\alpha}$$

$$\text{dynamics:} \qquad \overline{e^{-i \sum_{it} \hat{h}_i(t) \sum_j c_{ij} J_{ij} \sigma_j(t)}} = \sum_{\mathbf{c}} \frac{\delta_{\boldsymbol{\Omega},\boldsymbol{\Omega}(\mathbf{c})}}{Z(\boldsymbol{\Omega})} e^{-i \sum_{i<j} c_{ij} J_{ij} \sum_t [\hat{h}_i(t)\sigma_j(t) + \hat{h}_j(t)\sigma_i(t)]}$$

must do analytically: $\sum_{\mathbf{c}} \delta_{\boldsymbol{\Omega},\boldsymbol{\Omega}(\mathbf{c})} \prod_{i<j} e^{c_{ij} A_{ij}}$
even simpler ... $\sum_{\mathbf{c}} \delta_{\boldsymbol{\Omega},\boldsymbol{\Omega}(\mathbf{c})}$

if we can't calculate graph ensemble entropy,
we certainly can't solve the spin models ...

we wish to code more information into graphs,
for which graph ensembles $p(\mathbf{c})$
is a spin model still **solvable**?
e.g.

$$H(\boldsymbol{\sigma}) = -\sum_{i<j} c_{ij} J_{ij} \sigma_i \sigma_j, \qquad p(\mathbf{c}) = \frac{\delta_{\boldsymbol{\Omega},\boldsymbol{\Omega}(\mathbf{c})}}{Z(\boldsymbol{\Omega})}, \qquad \boldsymbol{\Omega}(\mathbf{c}) = (\Omega_1(\mathbf{c}), \ldots, \Omega_L(\mathbf{c}))$$

when can we do the average over graphs **analytically**?

$$\text{statics:} \qquad \overline{\mathrm{e}^{-\beta \sum_\alpha H(\boldsymbol{\sigma}^\alpha)}} = \sum_{\mathbf{c}} \frac{\delta_{\boldsymbol{\Omega},\boldsymbol{\Omega}(\mathbf{c})}}{Z(\boldsymbol{\Omega})} \mathrm{e}^{\beta \sum_{i<j} c_{ij} J_{ij} \sum_\alpha \sigma_i^\alpha \sigma_j^\alpha}$$

$$\text{dynamics:} \qquad \overline{\mathrm{e}^{-\mathrm{i} \sum_{it} \hat{h}_i(t) \sum_j c_{ij} J_{ij} \sigma_j(t)}} = \sum_{\mathbf{c}} \frac{\delta_{\boldsymbol{\Omega},\boldsymbol{\Omega}(\mathbf{c})}}{Z(\boldsymbol{\Omega})} \mathrm{e}^{-\mathrm{i} \sum_{i<j} c_{ij} J_{ij} \sum_t [\hat{h}_i(t)\sigma_j(t) + \hat{h}_j(t)\sigma_i(t)]}$$

must do analytically: $\sum_{\mathbf{c}} \delta_{\boldsymbol{\Omega},\boldsymbol{\Omega}(\mathbf{c})} \prod_{i<j} \mathrm{e}^{c_{ij} A_{ij}}$
even simpler ... $\sum_{\mathbf{c}} \delta_{\boldsymbol{\Omega},\boldsymbol{\Omega}(\mathbf{c})}$

if we can't calculate graph ensemble entropy,
we certainly can't solve the spin models ...

we wish to code more information into graphs,
for which graph ensembles $p(\mathbf{c})$
is a spin model still **solvable**?
e.g.

$$H(\boldsymbol{\sigma}) = -\sum_{i<j} c_{ij} J_{ij} \sigma_i \sigma_j, \qquad p(\mathbf{c}) = \frac{\delta_{\boldsymbol{\Omega},\boldsymbol{\Omega}(\mathbf{c})}}{Z(\boldsymbol{\Omega})}, \qquad \boldsymbol{\Omega}(\mathbf{c}) = (\Omega_1(\mathbf{c}), \ldots, \Omega_L(\mathbf{c}))$$

when can we do the average over graphs **analytically**?

$$\text{statics:} \quad \overline{\mathrm{e}^{-\beta \sum_\alpha H(\boldsymbol{\sigma}^\alpha)}} = \sum_{\mathbf{c}} \frac{\delta_{\boldsymbol{\Omega},\boldsymbol{\Omega}(\mathbf{c})}}{Z(\boldsymbol{\Omega})} \mathrm{e}^{\beta \sum_{i<j} c_{ij} J_{ij} \sum_\alpha \sigma_i^\alpha \sigma_j^\alpha}$$

$$\text{dynamics:} \quad \overline{\mathrm{e}^{-\mathrm{i} \sum_{it} \hat{h}_i(t) \sum_j c_{ij} J_{ij} \sigma_j(t)}} = \sum_{\mathbf{c}} \frac{\delta_{\boldsymbol{\Omega},\boldsymbol{\Omega}(\mathbf{c})}}{Z(\boldsymbol{\Omega})} \mathrm{e}^{-\mathrm{i} \sum_{i<j} c_{ij} J_{ij} \sum_t [\hat{h}_i(t)\sigma_j(t) + \hat{h}_j(t)\sigma_i(t)]}$$

must do analytically: $\sum_{\mathbf{c}} \delta_{\boldsymbol{\Omega},\boldsymbol{\Omega}(\mathbf{c})} \prod_{i<j} \mathrm{e}^{c_{ij} A_{ij}}$
even simpler ... $\sum_{\mathbf{c}} \delta_{\boldsymbol{\Omega},\boldsymbol{\Omega}(\mathbf{c})}$

if we can't calculate graph ensemble entropy,
we certainly can't solve the spin models ...

we wish to code more information into graphs,
for which graph ensembles $p(\mathbf{c})$
is a spin model still **solvable**?
e.g.

$$H(\boldsymbol{\sigma}) = -\sum_{i<j} c_{ij} J_{ij} \sigma_i \sigma_j, \qquad p(\mathbf{c}) = \frac{\delta_{\mathbf{\Omega},\mathbf{\Omega}(\mathbf{c})}}{Z(\mathbf{\Omega})}, \qquad \mathbf{\Omega}(\mathbf{c}) = (\Omega_1(\mathbf{c}), \ldots, \Omega_L(\mathbf{c}))$$

when can we do the average over graphs **analytically**?

$$\text{statics}: \quad \overline{\mathrm{e}^{-\beta \sum_\alpha H(\boldsymbol{\sigma}^\alpha)}} = \sum_{\mathbf{c}} \frac{\delta_{\mathbf{\Omega},\mathbf{\Omega}(\mathbf{c})}}{Z(\mathbf{\Omega})} \mathrm{e}^{\beta \sum_{i<j} c_{ij} J_{ij} \sum_\alpha \sigma_i^\alpha \sigma_j^\alpha}$$

$$\text{dynamics}: \quad \overline{\mathrm{e}^{-\mathrm{i} \sum_{it} \hat{h}_i(t) \sum_j c_{ij} J_{ij} \sigma_j(t)}} = \sum_{\mathbf{c}} \frac{\delta_{\mathbf{\Omega},\mathbf{\Omega}(\mathbf{c})}}{Z(\mathbf{\Omega})} \mathrm{e}^{-\mathrm{i} \sum_{i<j} c_{ij} J_{ij} \sum_t [\hat{h}_i(t)\sigma_j(t) + \hat{h}_j(t)\sigma_i(t)]}$$

must do analytically: $\sum_{\mathbf{c}} \delta_{\mathbf{\Omega},\mathbf{\Omega}(\mathbf{c})} \prod_{i<j} \mathrm{e}^{c_{ij} A_{ij}}$
even simpler ... $\sum_{\mathbf{c}} \delta_{\mathbf{\Omega},\mathbf{\Omega}(\mathbf{c})}$

if we can't calculate graph ensemble entropy,
we certainly can't solve the spin models ...

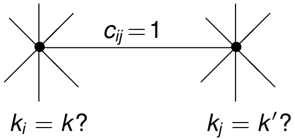
Outline

1. Background and origin of the problem
   - Protein interaction and gene regulation networks
   - Nonequilibrium stat mech of cellular signalling
   - Quantify graph topology beyond degrees
2. Analysis of tailored random graph ensembles
   - Canonical graph ensemble with controlled $\{p, \Pi\}$
   - Statistical mechanical analysis - entropy and beyond
   - Application to real PIN data
3. Generating tailored random graphs numerically
   - Degree-constrained 'shuffling'
   - Formulation as a stochastic process
   - Graph mobility
4. Ongoing work
5. Summary

Quantify graph topology beyond degrees

degree stats of
connected nodes

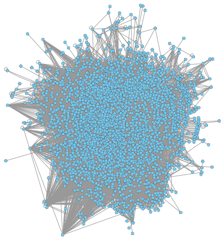
$$W(k,k'|\mathbf{c}) = \frac{1}{N\langle k\rangle}\sum_{ij} c_{ij}\delta_{k,k_i(\mathbf{c})}\delta_{k',k_j(\mathbf{c})}$$

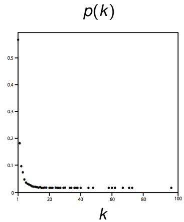


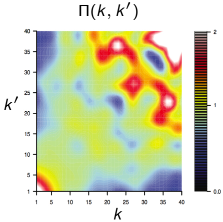
- $W(k|\mathbf{c}) = p(k|\mathbf{c})k/\langle k\rangle$
  so focus on

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

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



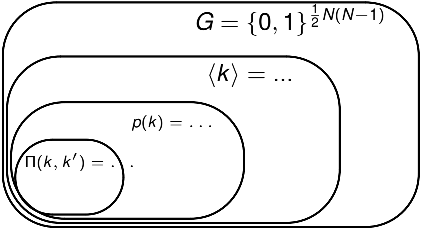




H sapiens PIN
$N = 9306$
$\langle k\rangle = 7.53$

Tailored random graph ensembles

Graph classification
via increasingly detailed
measurements



we are led to study:

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

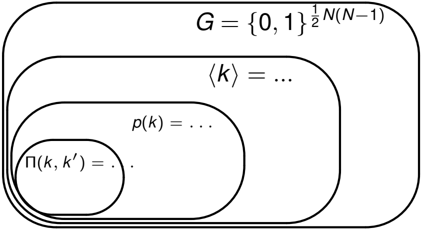
– proxies for networks in stat mech process modelling
– apply information theory and stat mech!

if you have bio-informatics friends:

– complexity: how many networks exist with same features as **c**?
– hypothesis testing: graphs with controlled features as null models
– quantifying network dissimilarity

Tailored random graph ensembles

Graph classification
via increasingly detailed
measurements



we are led to study:

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

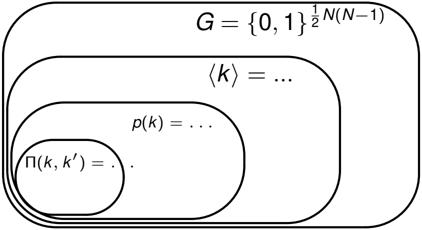
– proxies for networks in stat mech process modelling
– apply information theory and stat mech!

if you have bio-informatics friends:

– complexity: how many networks exist with same features as **c**?
– hypothesis testing: graphs with controlled features as null models
– quantifying network dissimilarity

Tailored random graph ensembles

Graph classification
via increasingly detailed
measurements



we are led to study:

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

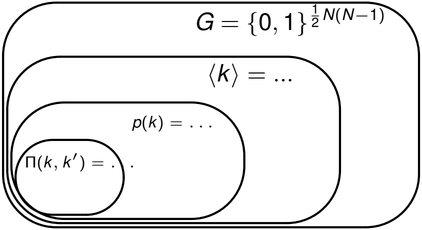
– proxies for networks in stat mech process modelling
– apply information theory and stat mech!

if you have bio-informatics friends:

– complexity: how many networks exist with same features as **c**?
– hypothesis testing: graphs with controlled features as null models
– quantifying network dissimilarity

Tailored random graph ensembles

Graph classification
via increasingly detailed
measurements



we are led to study:

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

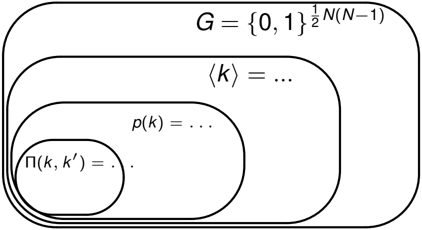
– proxies for networks in stat mech process modelling
– apply information theory and stat mech!

if you have bio-informatics friends:

– complexity: how many networks exist with same features as **c**?
– hypothesis testing: graphs with controlled features as null models
– quantifying network dissimilarity

Tailored random graph ensembles

Graph classification
via increasingly detailed
measurements



we are led to study:

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

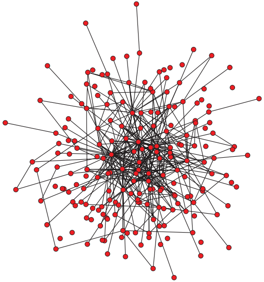
– proxies for networks in stat mech process modelling
– apply information theory and stat mech!

if you have bio-informatics friends:

– complexity: how many networks exist with same features as **c**?
– hypothesis testing: graphs with controlled features as null models
– quantifying network dissimilarity

Outline

Associate with each PIN a random graph ensemble:
(maximum entropy, <u>h</u>ard/<u>s</u>oft constraints)



- impose $\langle k\rangle$:

$$p_{\rm h}(\mathbf{c}|\langle k\rangle) = Z_{\rm h}^{-1}(\langle k\rangle)\,\delta_{\sum_{ij}c_{ij},N\langle k\rangle}$$

$$p_{\rm s}(\mathbf{c}|\langle k\rangle) = \prod_{i<j}\Big[\frac{\langle k\rangle}{N}\delta_{c_{ij},1}+(1-\frac{\langle k\rangle}{N})\delta_{c_{ij},0}\Big]$$

(Erdös-Rényi)

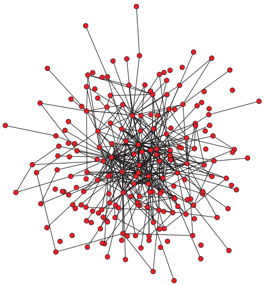
- impose degrees $\mathbf{k}=(k_1,\ldots,k_N)$:

$$p_{\rm h}(\mathbf{c}|\mathbf{k}) = Z_{\rm h}^{-1}(\mathbf{k})\,\prod_i \delta_{\sum_j c_{ij},k_i}$$

$$p_{\rm s}(\mathbf{c}|\mathbf{k}) = \prod_{i<j}\Big[\frac{e^{\omega_i+\omega_j}}{1+e^{\omega_i+\omega_j}}\delta_{c_{ij},1}+\frac{1}{1+e^{\omega_i+\omega_j}}\delta_{c_{ij},0}\Big]$$

Tailored random graph ensembles

Associate with each PIN a random graph ensemble:
(maximum entropy, hard/soft constraints)



- impose $\langle k \rangle$:

$$p_{\rm h}(\mathbf{c}|\langle k\rangle) = Z_{\rm h}^{-1}(\langle k\rangle)\ \delta_{\sum_{ij} c_{ij}, N\langle k\rangle}$$

$$p_{\rm s}(\mathbf{c}|\langle k\rangle) = \prod_{i<j}\Big[\frac{\langle k\rangle}{N}\delta_{c_{ij},1} + (1-\frac{\langle k\rangle}{N})\delta_{c_{ij},0}\Big]$$

(Erdös-Rényi)

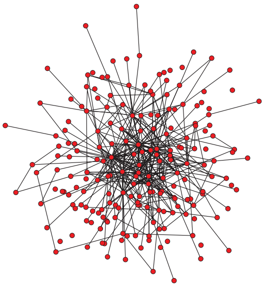
- impose degrees $\mathbf{k} = (k_1, \ldots, k_N)$:

$$p_{\rm h}(\mathbf{c}|\mathbf{k}) = Z_{\rm h}^{-1}(\mathbf{k})\ \prod_i \delta_{\sum_j c_{ij}, k_i}$$

$$p_{\rm s}(\mathbf{c}|\mathbf{k}) = \prod_{i<j}\Big[\frac{e^{\omega_i+\omega_j}}{1+e^{\omega_i+\omega_j}}\delta_{c_{ij},1} + \frac{1}{1+e^{\omega_i+\omega_j}}\delta_{c_{ij},0}\Big]$$

Tailored random graph ensembles

Associate with each PIN a random graph ensemble:
(maximum entropy, hard/soft constraints)



- impose $\langle k \rangle$:

$$
\begin{aligned}
p_{\mathrm{h}}(\mathbf{c}|\langle k\rangle) &= Z_{\mathrm{h}}^{-1}(\langle k\rangle)\,\delta_{\sum_{ij}c_{ij},N\langle k\rangle} \\
p_{\mathrm{s}}(\mathbf{c}|\langle k\rangle) &= \prod_{i<j}\Big[\frac{\langle k\rangle}{N}\delta_{c_{ij},1}+(1-\frac{\langle k\rangle}{N})\delta_{c_{ij},0}\Big]
\end{aligned}
$$

(Erdös-Rényi)

- impose degrees $\mathbf{k}=(k_1,\ldots,k_N)$:

$$
\begin{aligned}
p_{\mathrm{h}}(\mathbf{c}|\mathbf{k}) &= Z_{\mathrm{h}}^{-1}(\mathbf{k})\prod_i \delta_{\sum_j c_{ij},k_i} \\
p_{\mathrm{s}}(\mathbf{c}|\mathbf{k}) &= \prod_{i<j}\Big[\frac{\mathrm{e}^{\omega_i+\omega_j}}{1+\mathrm{e}^{\omega_i+\omega_j}}\delta_{c_{ij},1}+\frac{1}{1+\mathrm{e}^{\omega_i+\omega_j}}\delta_{c_{ij},0}\Big]
\end{aligned}
$$

- impose degrees $\mathbf{k}$ and $W(k,k')$:

$$p_{\rm h}(\mathbf{c}|\mathbf{k},W)=\frac{\delta_{\mathbf{k},\mathbf{k}(\mathbf{c})}}{Z_{\rm h}(\mathbf{k},W)}\prod_{kk'}\delta_{\sum_{ij}\delta_{k,k_i(\mathbf{c})}c_{ij}\delta_{k',k_j(\mathbf{c})},N\langle k\rangle W(k,k')}$$

$$p_{\rm s}(\mathbf{c}|\mathbf{k},W)=\frac{1}{Z_{\rm s}(\mathbf{k},W)}\,{\rm e}^{\sum_{i<j}c_{ij}[\omega_i+\omega_j+\psi(k_i(\mathbf{c}),k_j(\mathbf{c})+\psi(k_i(\mathbf{c}),k_j(\mathbf{c})]}$$

$\{\omega_i,\psi(k,k')\}$ to be solved from

$$\forall i:\qquad \sum_{\mathbf{c}\in G}p_{\rm s}(\mathbf{c}|\mathbf{k},W)\sum_j c_{ij}=k_i$$

$$\forall k,k':\qquad \sum_{\mathbf{c}\in G}p_{\rm s}(\mathbf{c}|\mathbf{k},W)\frac{1}{N\langle k\rangle}\sum_{ij}c_{ij}\delta_{k,k_i(\mathbf{c})}\delta_{k',k_j(\mathbf{c})}=W(k,k')$$

graphs share more features with the biological networks,
price paid: mathematical and computational complexity

struggle to solve eqns for Lagrange parameters by numerical sampling,
e.g. $N=1000$: $\quad 2^{\frac{1}{2}N(N-1)}\approx 10^{150,364}$ graphs ...

- impose degrees $\mathbf{k}$ and $W(k,k')$:

$$p_{\rm h}(\mathbf{c}|\mathbf{k},W)=\frac{\delta_{\mathbf{k},\mathbf{k}(\mathbf{c})}}{Z_{\rm h}(\mathbf{k},W)}\prod_{kk'}\delta_{\sum_{ij}\delta_{k,k_i(\mathbf{c})}c_{ij}\delta_{k',k_j(\mathbf{c})},N\langle k\rangle W(k,k')}$$

$$p_{\rm s}(\mathbf{c}|\mathbf{k},W)=\frac{1}{Z_{\rm s}(\mathbf{k},W)}\,{\rm e}^{\sum_{i<j}c_{ij}[\omega_i+\omega_j+\psi(k_i(\mathbf{c}),k_j(\mathbf{c})+\psi(k_i(\mathbf{c}),k_j(\mathbf{c})]}$$

$\{\omega_i,\psi(k,k')\}$ to be solved from

$$\forall i:\qquad \sum_{\mathbf{c}\in G}p_{\rm s}(\mathbf{c}|\mathbf{k},W)\sum_j c_{ij}=k_i$$

$$\forall k,k':\qquad \sum_{\mathbf{c}\in G}p_{\rm s}(\mathbf{c}|\mathbf{k},W)\frac{1}{N\langle k\rangle}\sum_{ij}c_{ij}\delta_{k,k_i(\mathbf{c})}\delta_{k',k_j(\mathbf{c})}=W(k,k')$$

graphs share more features with the biological networks,
price paid: mathematical and computational complexity

struggle to solve eqns for Lagrange parameters by numerical sampling,
e.g. $N=1000$: $2^{\frac{1}{2}N(N-1)}\approx 10^{150,364}$ graphs ...

- impose degrees $\mathbf{k}$ and $W(k,k')$:

$$p_{\rm h}(\mathbf{c}|\mathbf{k},W)=\frac{\delta_{\mathbf{k},\mathbf{k}(\mathbf{c})}}{Z_{\rm h}(\mathbf{k},W)}\prod_{kk'}\delta_{\sum_{ij}\delta_{k,k_i(\mathbf{c})}c_{ij}\delta_{k',k_j(\mathbf{c})},N\langle k\rangle W(k,k')}$$

$$p_{\rm s}(\mathbf{c}|\mathbf{k},W)=\frac{1}{Z_{\rm s}(\mathbf{k},W)}\,{\rm e}^{\sum_{i<j}c_{ij}[\omega_i+\omega_j+\psi(k_i(\mathbf{c}),k_j(\mathbf{c})+\psi(k_i(\mathbf{c}),k_j(\mathbf{c})]}$$

$\{\omega_i,\psi(k,k')\}$ to be solved from

$$\forall i:\qquad \sum_{\mathbf{c}\in G}p_{\rm s}(\mathbf{c}|\mathbf{k},W)\sum_j c_{ij}=k_i$$

$$\forall k,k':\qquad \sum_{\mathbf{c}\in G}p_{\rm s}(\mathbf{c}|\mathbf{k},W)\frac{1}{N\langle k\rangle}\sum_{ij}c_{ij}\delta_{k,k_i(\mathbf{c})}\delta_{k',k_j(\mathbf{c})}=W(k,k')$$

graphs share more features with the biological networks,
price paid: mathematical and computational complexity

struggle to solve eqns for Lagrange parameters by numerical sampling,
e.g. $N=1000$: $\;2^{\frac{1}{2}N(N-1)}\approx 10^{150,364}$ graphs ...

- impose degrees $\mathbf{k}$ and $W(k,k')$:

$$p_{\rm h}(\mathbf{c}|\mathbf{k},W)=\frac{\delta_{\mathbf{k},\mathbf{k}(\mathbf{c})}}{Z_{\rm h}(\mathbf{k},W)}\prod_{kk'}\delta_{\sum_{ij}\delta_{k,k_i(\mathbf{c})}c_{ij}\delta_{k',k_j(\mathbf{c})},N\langle k\rangle W(k,k')}$$

$$p_{\rm s}(\mathbf{c}|\mathbf{k},W)=\frac{1}{Z_{\rm s}(\mathbf{k},W)}\,{\rm e}^{\sum_{i<j}c_{ij}[\omega_i+\omega_j+\psi(k_i(\mathbf{c}),k_j(\mathbf{c})+\psi(k_i(\mathbf{c}),k_j(\mathbf{c})]}$$

$\{\omega_i,\psi(k,k')\}$ to be solved from

$$\forall i:\qquad \sum_{\mathbf{c}\in G}p_{\rm s}(\mathbf{c}|\mathbf{k},W)\sum_j c_{ij}=k_i$$

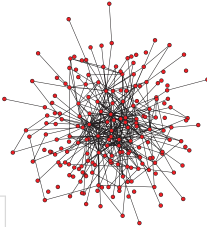
$$\forall k,k':\qquad \sum_{\mathbf{c}\in G}p_{\rm s}(\mathbf{c}|\mathbf{k},W)\frac{1}{N\langle k\rangle}\sum_{ij}c_{ij}\delta_{k,k_i(\mathbf{c})}\delta_{k',k_j(\mathbf{c})}=W(k,k')$$

graphs share more features with the biological networks,
price paid: mathematical and computational complexity

struggle to solve eqns for Lagrange parameters by numerical sampling,
e.g. $N=1000$: $2^{\frac{1}{2}N(N-1)}\approx 10^{150,364}$ graphs ...

Canonical graph ensemble with controlled $\{p, \Pi\}$

Need to proceed **analytically** if at all ...



$$p(\mathbf{c}|p,Q) = \sum_{\mathbf{k}} \Big(\prod_i p(k_i)\Big) p(\mathbf{c}|Q,\mathbf{k})$$

$$p(\mathbf{c}|\mathbf{k},Q) = \frac{\delta_{\mathbf{k},\mathbf{k}(\mathbf{c})}}{Z(\mathbf{k},\Pi)} \prod_{i<j} \Big[\frac{\overline{k}}{N} Q(k_i,k_j)\delta_{c_{ij},1} + \Big(1 - \frac{\overline{k}}{N} Q(k_i,k_j)\Big)\delta_{c_{ij},0}\Big]$$

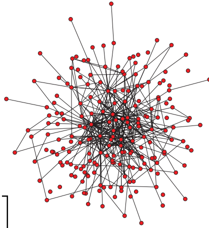
$\overline{k} = N^{-1}\sum_i k_i$,
$\sum_{kk'} p(k)Q(k,k')p(k') = 1$

For $Q(k,k') = kk'\Pi(k,k')/\overline{k}^2$:

- graphs have properties $p(k)$ and $\Pi(k,k')$ for $N \to \infty$
- $p(\mathbf{c}|\mathbf{k},\Pi)$ is maximum entropy ensemble, given constraints $\{\mathbf{k},\Pi\}$
- analytically tractable!

Canonical graph ensemble with controlled $\{p, \Pi\}$

Need to proceed **analytically** if at all ...



$$p(\mathbf{c}|p,Q) = \sum_{\mathbf{k}} \Big(\prod_i p(k_i) \Big) p(\mathbf{c}|Q,\mathbf{k})$$

$$p(\mathbf{c}|\mathbf{k},Q) = \frac{\delta_{\mathbf{k},\mathbf{k}(\mathbf{c})}}{Z(\mathbf{k},\Pi)} \prod_{i<j} \Big[\frac{\overline{k}}{N} Q(k_i,k_j)\delta_{c_{ij},1} + \Big(1 - \frac{\overline{k}}{N} Q(k_i,k_j)\Big)\delta_{c_{ij},0} \Big]$$

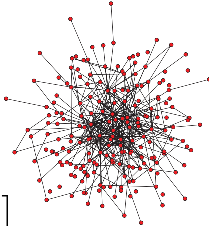
$\overline{k} = N^{-1}\sum_i k_i$,
$\sum_{kk'} p(k)Q(k,k')p(k') = 1$

For $Q(k,k') = kk'\Pi(k,k')/\overline{k}^2$:

- graphs have properties $p(k)$ and $\Pi(k,k')$ for $N \to \infty$
- $p(\mathbf{c}|\mathbf{k},\Pi)$ is maximum entropy ensemble, given constraints $\{\mathbf{k},\Pi\}$
- analytically tractable!

Canonical graph ensemble with controlled $\{p, \Pi\}$

Need to proceed **analytically** if at all ...



$$p(\mathbf{c}|p,Q) = \sum_{\mathbf{k}} \Big(\prod_i p(k_i)\Big) p(\mathbf{c}|Q,\mathbf{k})$$

$$p(\mathbf{c}|\mathbf{k},Q) = \frac{\delta_{\mathbf{k},\mathbf{k}(\mathbf{c})}}{Z(\mathbf{k},\Pi)} \prod_{i<j} \Big[\frac{\overline{k}}{N} Q(k_i,k_j)\delta_{c_{ij},1} + \Big(1 - \frac{\overline{k}}{N} Q(k_i,k_j)\Big)\delta_{c_{ij},0}\Big]$$

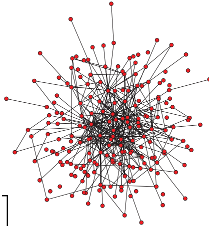
$\overline{k} = N^{-1}\sum_i k_i$,
$\sum_{kk'} p(k)Q(k,k')p(k') = 1$

For $Q(k,k') = kk'\Pi(k,k')/\overline{k}^2$:

- graphs have properties $p(k)$ and $\Pi(k,k')$ for $N \to \infty$
- $p(\mathbf{c}|\mathbf{k},\Pi)$ is maximum entropy ensemble, given constraints $\{\mathbf{k},\Pi\}$
- analytically tractable!

Canonical graph ensemble with controlled $\{p, \Pi\}$

Need to proceed **analytically** if at all ...



$$p(\mathbf{c}|p,Q) = \sum_{\mathbf{k}} \Big(\prod_i p(k_i)\Big) p(\mathbf{c}|Q,\mathbf{k})$$

$$p(\mathbf{c}|\mathbf{k},Q) = \frac{\delta_{\mathbf{k},\mathbf{k}(\mathbf{c})}}{Z(\mathbf{k},\Pi)} \prod_{i<j} \Big[\frac{\overline{k}}{N} Q(k_i,k_j)\delta_{c_{ij},1} + \Big(1-\frac{\overline{k}}{N} Q(k_i,k_j)\Big)\delta_{c_{ij},0}\Big]$$

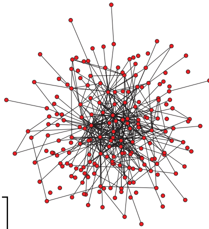
$\overline{k} = N^{-1}\sum_i k_i$,
$\sum_{kk'} p(k)Q(k,k')p(k') = 1$

For $Q(k,k') = kk'\Pi(k,k')/\overline{k}^2$:

- graphs have properties $p(k)$ and $\Pi(k,k')$ for $N \to \infty$
- $p(\mathbf{c}|\mathbf{k},\Pi)$ is maximum entropy ensemble, given constraints $\{\mathbf{k},\Pi\}$
- analytically tractable!

Canonical graph ensemble with controlled $\{p, \Pi\}$

Need to proceed **analytically** if at all ...



$$p(\mathbf{c}|p,Q) = \sum_{\mathbf{k}} \Big(\prod_i p(k_i)\Big) p(\mathbf{c}|Q,\mathbf{k})$$

$$p(\mathbf{c}|\mathbf{k},Q) = \frac{\delta_{\mathbf{k},\mathbf{k}(\mathbf{c})}}{Z(\mathbf{k},\Pi)} \prod_{i<j} \Big[\frac{\overline{k}}{N} Q(k_i,k_j)\delta_{c_{ij},1} + \Big(1 - \frac{\overline{k}}{N} Q(k_i,k_j)\Big)\delta_{c_{ij},0}\Big]$$

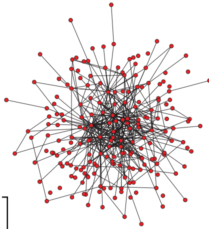
$\overline{k} = N^{-1}\sum_i k_i$,
$\sum_{kk'} p(k)Q(k,k')p(k') = 1$

For $Q(k,k') = kk'\Pi(k,k')/\overline{k}^2$:

- graphs have properties $p(k)$ and $\Pi(k,k')$ for $N \to \infty$
- $p(\mathbf{c}|\mathbf{k},\Pi)$ is maximum entropy ensemble, given constraints $\{\mathbf{k},\Pi\}$
- analytically tractable!

Canonical graph ensemble with controlled $\{p, \Pi\}$

Need to proceed **analytically** if at all ...



$$p(\mathbf{c}|p,Q) = \sum_{\mathbf{k}} \Big(\prod_i p(k_i)\Big) p(\mathbf{c}|Q,\mathbf{k})$$

$$p(\mathbf{c}|\mathbf{k},Q) = \frac{\delta_{\mathbf{k},\mathbf{k}(\mathbf{c})}}{Z(\mathbf{k},\Pi)} \prod_{i<j} \Big[\frac{\overline{k}}{N} Q(k_i,k_j)\delta_{c_{ij},1} + \Big(1-\frac{\overline{k}}{N}Q(k_i,k_j)\Big)\delta_{c_{ij},0}\Big]$$

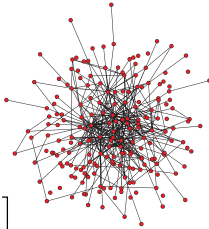
$\overline{k} = N^{-1}\sum_i k_i$,
$\sum_{kk'} p(k)Q(k,k')p(k') = 1$

For $Q(k,k') = kk'\Pi(k,k')/\overline{k}^2$:

- graphs have properties $p(k)$ and $\Pi(k,k')$ for $N \to \infty$
- $p(\mathbf{c}|\mathbf{k},\Pi)$ is maximum entropy ensemble, given constraints $\{\mathbf{k},\Pi\}$
- analytically tractable!

Canonical graph ensemble with controlled $\{p, \Pi\}$

Need to proceed **analytically** if at all ...



$$p(\mathbf{c}|p,Q) = \sum_{\mathbf{k}} \Big(\prod_i p(k_i)\Big) p(\mathbf{c}|Q,\mathbf{k})$$

$$p(\mathbf{c}|\mathbf{k},Q) = \frac{\delta_{\mathbf{k},\mathbf{k}(\mathbf{c})}}{Z(\mathbf{k},\Pi)} \prod_{i<j} \Big[\frac{\overline{k}}{N} Q(k_i,k_j)\delta_{c_{ij},1} + \Big(1 - \frac{\overline{k}}{N} Q(k_i,k_j)\Big)\delta_{c_{ij},0}\Big]$$

$\overline{k} = N^{-1} \sum_i k_i$,
$\sum_{kk'} p(k)Q(k,k')p(k') = 1$

For $Q(k,k') = kk'\Pi(k,k')/\overline{k}^2$:

- graphs have properties $p(k)$ and $\Pi(k,k')$ for $N \to \infty$
- $p(\mathbf{c}|\mathbf{k},\Pi)$ is maximum entropy ensemble, given constraints $\{\mathbf{k},\Pi\}$
- analytically tractable!

Outline

Stat mech calculation of Shannon entropy

Shannon entropy per node, effective nr of graphs $\mathcal{N}$

$$S[p,\Pi] = -\frac{1}{N}\sum_{\mathbf{c}} p(\mathbf{c}|p,\Pi)\ \log p(\mathbf{c}|p,\Pi), \qquad \mathcal{N}[p,\Pi] = e^{NS[p,\Pi]}$$

simple manipulations:

$$S[p,\Pi] = -\sum_k p(k)\log p(k) + \sum_{\mathbf{k}}\Big[\prod_i p(k_i)\Big]\Big[A(\mathbf{k}) - B(\mathbf{k})\Big]$$

with

$$A(\mathbf{k}) = \frac{1}{N}\log\langle\delta_{\mathbf{k},\mathbf{k}(\mathbf{c})}\rangle_w \qquad B(\mathbf{k}) = \frac{1}{N}\frac{\langle\delta_{\mathbf{k},\mathbf{k}(\mathbf{c})}\log W(\mathbf{c}|\mathbf{k})\rangle_w}{\langle\delta_{\mathbf{k},\mathbf{k}(\mathbf{c})}\rangle_w}$$

$$W(\mathbf{c}|\mathbf{k}) = \prod_{i<j}\Big[\frac{\overline{k}}{N}Q(k_i,k_j)\delta_{c_{ij},1} + \Big(1-\frac{\overline{k}}{N}Q(k_i,k_j)\Big)\delta_{c_{ij},0}\Big]$$

Stat mech calculation of Shannon entropy

Shannon entropy per node, effective nr of graphs $\mathcal{N}$

$$S[p,\Pi] = -\frac{1}{N}\sum_{\mathbf{c}} p(\mathbf{c}|p,\Pi)\ \log p(\mathbf{c}|p,\Pi), \qquad \mathcal{N}[p,\Pi] = e^{NS[p,\Pi]}$$

simple
manipulations:

$$S[p,\Pi] = -\sum_{k} p(k)\log p(k) + \sum_{\mathbf{k}}\Big[\prod_i p(k_i)\Big]\Big[A(\mathbf{k}) - B(\mathbf{k})\Big]$$

with

$$A(\mathbf{k}) = \frac{1}{N}\log\langle\delta_{\mathbf{k},\mathbf{k}(\mathbf{c})}\rangle_w \qquad B(\mathbf{k}) = \frac{1}{N}\frac{\langle\delta_{\mathbf{k},\mathbf{k}(\mathbf{c})}\log W(\mathbf{c}|\mathbf{k})\rangle_w}{\langle\delta_{\mathbf{k},\mathbf{k}(\mathbf{c})}\rangle_w}$$

$$W(\mathbf{c}|\mathbf{k}) = \prod_{i<j}\Big[\frac{\overline{k}}{N}Q(k_i,k_j)\delta_{c_{ij},1} + \Big(1-\frac{\overline{k}}{N}Q(k_i,k_j)\Big)\delta_{c_{ij},0}\Big]$$

Stat mech calculation of Shannon entropy

Shannon entropy per node, effective nr of graphs $\mathcal{N}$

$$S[p,\Pi] = -\frac{1}{N}\sum_{\mathbf{c}} p(\mathbf{c}|p,\Pi)\ \log p(\mathbf{c}|p,\Pi), \qquad \mathcal{N}[p,\Pi] = e^{NS[p,\Pi]}$$

simple
manipulations:

$$S[p,\Pi] = -\sum_k p(k)\log p(k) + \sum_{\mathbf{k}}\Big[\prod_i p(k_i)\Big]\Big[A(\mathbf{k}) - B(\mathbf{k})\Big]$$

with

$$A(\mathbf{k}) = \frac{1}{N}\log\langle\delta_{\mathbf{k},\mathbf{k}(\mathbf{c})}\rangle_w \qquad B(\mathbf{k}) = \frac{1}{N}\frac{\langle\delta_{\mathbf{k},\mathbf{k}(\mathbf{c})}\log W(\mathbf{c}|\mathbf{k})\rangle_w}{\langle\delta_{\mathbf{k},\mathbf{k}(\mathbf{c})}\rangle_w}$$

$$W(\mathbf{c}|\mathbf{k}) = \prod_{i<j}\Big[\frac{\overline{k}}{N}Q(k_i,k_j)\delta_{c_{ij},1} + \Big(1-\frac{\overline{k}}{N}Q(k_i,k_j)\Big)\delta_{c_{ij},0}\Big]$$

- degree constraints in integral form

$$\delta_{\mathbf{k},\mathbf{k}(\mathbf{c})} = \int_{-\pi}^{\pi} \prod_i \left[\frac{\mathrm{d}\omega_i}{2\pi} \mathrm{e}^{\mathrm{i}\omega_i(k_i - \sum_j c_{ij})}\right]$$

sum over graphs:

$$A(\mathbf{k}) = \frac{1}{N} \log \int \frac{\mathrm{d}\omega\, \mathrm{e}^{\mathrm{i}\boldsymbol{\omega}\cdot\mathbf{k}}}{(2\pi)^N} \prod_{i<j} \left[1 + \frac{\overline{k}}{N} Q(k_i, k_j)[\mathrm{e}^{-\mathrm{i}(\omega_i+\omega_j)} - 1]\right]$$

$$B(\mathbf{k}) = \mathrm{e}^{-NA(\mathbf{k})} \int \frac{\mathrm{d}\omega\, \mathrm{e}^{\mathrm{i}\boldsymbol{\omega}\cdot\mathbf{k}}}{(2\pi)^N} \prod_{i<j} \left[1 + \frac{\overline{k}}{N} Q(k_i, k_j)[\mathrm{e}^{-\mathrm{i}(\omega_i+\omega_j)} - 1]\right] \times \frac{\overline{k}}{N^2} \sum_{\ell<m} Q(k_\ell, k_m)\left[\mathrm{e}^{-\mathrm{i}(\omega_\ell+\omega_m)} \log[\frac{\overline{k}}{N} Q(k_\ell, k_m)] - 1 + \mathcal{O}(\frac{1}{N})\right]$$

- path integrals to get site factorization:
  order parameter: $P(k, \omega) = N^{-1} \sum_i \delta_{k,k_i} \delta[\omega - \omega_i]$
  (related to change in entropy upon single node removal)
- integral $\int \{\mathrm{d}P \mathrm{d}\hat{P}\} \ldots$ via steepest descent, functional saddle-point eqns

- degree constraints in integral form

$$\delta_{\mathbf{k},\mathbf{k}(\mathbf{c})} = \int_{-\pi}^{\pi} \prod_i \left[\frac{\mathrm{d}\omega_i}{2\pi} \mathrm{e}^{\mathrm{i}\omega_i(k_i - \sum_j c_{ij})}\right]$$

  sum over graphs:

$$A(\mathbf{k}) = \frac{1}{N} \log \int \frac{\mathrm{d}\omega\ \mathrm{e}^{\mathrm{i}\boldsymbol{\omega}\cdot\mathbf{k}}}{(2\pi)^N} \prod_{i<j} \left[1 + \frac{\overline{k}}{N} Q(k_i, k_j)[\mathrm{e}^{-\mathrm{i}(\omega_i+\omega_j)} - 1]\right]$$

$$B(\mathbf{k}) = \mathrm{e}^{-NA(\mathbf{k})} \int \frac{\mathrm{d}\omega\ \mathrm{e}^{\mathrm{i}\boldsymbol{\omega}\cdot\mathbf{k}}}{(2\pi)^N} \prod_{i<j} \left[1 + \frac{\overline{k}}{N} Q(k_i, k_j)[\mathrm{e}^{-\mathrm{i}(\omega_i+\omega_j)} - 1]\right]$$
$$\times \frac{\overline{k}}{N^2} \sum_{\ell<m} Q(k_\ell, k_m) \left[\mathrm{e}^{-\mathrm{i}(\omega_\ell+\omega_m)} \log[\frac{\overline{k}}{N} Q(k_\ell, k_m)] - 1 + \mathcal{O}(\frac{1}{N})\right]$$

- path integrals to get site factorization:
  order parameter: $P(k,\omega) = N^{-1} \sum_i \delta_{k,k_i} \delta[\omega - \omega_i]$
  (related to change in entropy upon single node removal)

- integral $\int \{\mathrm{d}P \mathrm{d}\hat{P}\} \ldots$ via steepest descent, functional saddle-point eqns

- degree constraints in integral form
$$\delta_{\mathbf{k},\mathbf{k}(\mathbf{c})}=\int_{-\pi}^{\pi}\prod_i\left[\frac{\mathrm{d}\omega_i}{2\pi}\mathrm{e}^{\mathrm{i}\omega_i(k_i-\sum_j c_{ij})}\right]$$
sum over graphs:
$$A(\mathbf{k})=\frac{1}{N}\log\int\frac{\mathrm{d}\boldsymbol{\omega}\;\mathrm{e}^{\mathrm{i}\boldsymbol{\omega}\cdot\mathbf{k}}}{(2\pi)^N}\prod_{i<j}\left[1+\frac{\overline{k}}{N}Q(k_i,k_j)[\mathrm{e}^{-\mathrm{i}(\omega_i+\omega_j)}-1]\right]$$
$$B(\mathbf{k})=\mathrm{e}^{-NA(\mathbf{k})}\int\frac{\mathrm{d}\boldsymbol{\omega}\;\mathrm{e}^{\mathrm{i}\boldsymbol{\omega}\cdot\mathbf{k}}}{(2\pi)^N}\prod_{i<j}\left[1+\frac{\overline{k}}{N}Q(k_i,k_j)[\mathrm{e}^{-\mathrm{i}(\omega_i+\omega_j)}-1]\right]$$
$$\times\frac{\overline{k}}{N^2}\sum_{\ell<m}Q(k_\ell,k_m)\left[\mathrm{e}^{-\mathrm{i}(\omega_\ell+\omega_m)}\log[\frac{\overline{k}}{N}Q(k_\ell,k_m)]-1+\mathcal{O}(\frac{1}{N})\right]$$
- path integrals to get site factorization:
order parameter: $P(k,\omega)=N^{-1}\sum_i\delta_{k,k_i}\delta[\omega-\omega_i]$
(related to change in entropy upon single node removal)
- integral $\int\{\mathrm{d}P\mathrm{d}\hat{P}\}\ldots$ via steepest descent, functional saddle-point eqns

- degree constraints in integral form
$$\delta_{\mathbf{k},\mathbf{k}(\mathbf{c})} = \int_{-\pi}^{\pi} \prod_i \left[\frac{\mathrm{d}\omega_i}{2\pi} \mathrm{e}^{\mathrm{i}\omega_i(k_i - \sum_j c_{ij})}\right]$$

  sum over graphs:

$$A(\mathbf{k}) = \frac{1}{N} \log \int \frac{\mathrm{d}\boldsymbol{\omega}\ \mathrm{e}^{\mathrm{i}\boldsymbol{\omega}\cdot\mathbf{k}}}{(2\pi)^N} \prod_{i<j} \left[1 + \frac{\overline{k}}{N} Q(k_i, k_j)[\mathrm{e}^{-\mathrm{i}(\omega_i+\omega_j)} - 1]\right]$$

$$B(\mathbf{k}) = \mathrm{e}^{-NA(\mathbf{k})} \int \frac{\mathrm{d}\boldsymbol{\omega}\ \mathrm{e}^{\mathrm{i}\boldsymbol{\omega}\cdot\mathbf{k}}}{(2\pi)^N} \prod_{i<j} \left[1 + \frac{\overline{k}}{N} Q(k_i, k_j)[\mathrm{e}^{-\mathrm{i}(\omega_i+\omega_j)} - 1]\right]$$
$$\times \frac{\overline{k}}{N^2} \sum_{\ell<m} Q(k_\ell, k_m) \left[\mathrm{e}^{-\mathrm{i}(\omega_\ell+\omega_m)} \log[\frac{\overline{k}}{N} Q(k_\ell, k_m)] - 1 + \mathcal{O}(\frac{1}{N})\right]$$

- path integrals to get site factorization:
  order parameter: $P(k,\omega) = N^{-1} \sum_i \delta_{k,k_i} \delta[\omega - \omega_i]$
  (related to change in entropy upon single node removal)
- integral $\int \{\mathrm{d}P \mathrm{d}\hat{P}\} \ldots$ via steepest descent, functional saddle-point eqns

Stat mech calculation – analysis of saddle point eqns

- $\delta\Psi/\delta P = \delta\Psi/\delta\hat{P} = 0$,

$$\Psi[P,\hat{P}] = \mathrm{i}\int\!\mathrm{d}\omega\sum_k \hat{P}(k,\omega)P(k,\omega) + \sum_k p(k)\log\int\frac{d\omega}{2\pi}\mathrm{e}^{\mathrm{i}\omega k-\mathrm{i}\hat{P}(k,\omega)}$$
$$+\frac{1}{2}\overline{k}\int\!\mathrm{d}\omega\mathrm{d}\omega'\sum_{kk'}P(k,\omega)P(k',\omega')Q(k,k')[\mathrm{e}^{-\mathrm{i}(\omega+\omega')}-1]$$

- soln of SP eqns:

$$P(k,\omega) = p(k)k!\,\frac{\mathrm{e}^{\mathrm{i}\omega k+\overline{k}\mathrm{e}^{-\mathrm{i}\omega}\gamma(k)}}{2\pi\overline{k}^k\gamma^k(k)}, \qquad \gamma(k) = \sum_{k'}Q(k,k')p(k')\frac{k'}{\overline{k}\gamma(k')}$$

$\gamma(k)$ unique, but how to find it?
special simple kernels only?
numerics?

Stat mech calculation – analysis of saddle point eqns

- $\delta\Psi/\delta P = \delta\Psi/\delta\hat{P} = 0$,

$$\Psi[P,\hat{P}] = \mathrm{i}\int \mathrm{d}\omega \sum_k \hat{P}(k,\omega)P(k,\omega) + \sum_k p(k)\log\int \frac{d\omega}{2\pi}\mathrm{e}^{\mathrm{i}\omega k - \mathrm{i}\hat{P}(k,\omega)}$$
$$+\frac{1}{2}\overline{k}\int \mathrm{d}\omega\mathrm{d}\omega' \sum_{kk'} P(k,\omega)P(k',\omega')Q(k,k')[\mathrm{e}^{-\mathrm{i}(\omega+\omega')}-1]$$

- soln of SP eqns:

$$P(k,\omega) = p(k)k!\,\frac{\mathrm{e}^{\mathrm{i}\omega k + \overline{k}\mathrm{e}^{-\mathrm{i}\omega}\gamma(k)}}{2\pi\overline{k}^k\gamma^k(k)}, \qquad \gamma(k) = \sum_{k'} Q(k,k')p(k')\frac{k'}{\overline{k}\gamma(k')}$$

$\gamma(k)$ unique, but how to find it?
special simple kernels only?
numerics?

Stat mech calculation – analysis of saddle point eqns

- $\delta\Psi/\delta P = \delta\Psi/\delta\hat{P} = 0$,

$$\Psi[P,\hat{P}] = \mathrm{i}\int\!\mathrm{d}\omega\sum_k \hat{P}(k,\omega)P(k,\omega) + \sum_k p(k)\log\int\frac{d\omega}{2\pi}\mathrm{e}^{\mathrm{i}\omega k-\mathrm{i}\hat{P}(k,\omega)}$$
$$+\frac{1}{2}\overline{k}\int\!\mathrm{d}\omega\mathrm{d}\omega'\sum_{kk'}P(k,\omega)P(k',\omega')Q(k,k')[\mathrm{e}^{-\mathrm{i}(\omega+\omega')}-1]$$

- soln of SP eqns:

$$P(k,\omega) = p(k)k!\,\frac{\mathrm{e}^{\mathrm{i}\omega k+\overline{k}\mathrm{e}^{-\mathrm{i}\omega}\gamma(k)}}{2\pi\overline{k}^k\gamma^k(k)}, \qquad \gamma(k) = \sum_{k'}Q(k,k')p(k')\frac{k'}{\overline{k}\gamma(k')}$$

$\gamma(k)$ unique, but how to find it?
special simple kernels only?
numerics?

Canonical kernel

can solve SP eqn **analytically**!

- Calculate $\Pi(k,k')$:

$$\Pi(k,k') = Q(k,k')\Big(\frac{p(k)k}{\overline{k}\gamma(k)}\Big)\Big(\frac{p(k')k'}{\overline{k}\gamma(k')}\Big), \quad \gamma(k) = \sum_{k'} Q(k,k')p(k')\frac{k'}{\overline{k}\gamma(k')}$$

- transform $Q(k,k') \to \mu(k)Q(k,k')\mu(k')$:

$$\Pi(k,k') \to Q(k,k')\Big(\frac{p(k)k}{\overline{k}\tilde{\gamma}(k)}\Big)\Big(\frac{p(k')k'}{\overline{k}\tilde{\gamma}(k')}\Big), \quad \tilde{\gamma}(k) = \sum_{k'} Q(k,k')p(k')\frac{k}{\overline{k}\tilde{\gamma}(k')}$$

$$\tilde{\gamma}(k) = \gamma(k)/\mu(k)$$

$\Pi(k,k')$ **invariant**!

- choose: $Q(k,k') = \Pi(k,k')kk'/\overline{k}^2$

$$\Big(\frac{\gamma(k)}{k/\overline{k}}\Big)W(k) = \sum_{k'} W(k,k')\Big(\frac{k'/\overline{k}}{\gamma(k')}\Big)$$

solution trivial!

Canonical kernel

can solve SP eqn **analytically**!

- Calculate $\Pi(k,k')$:

$$\Pi(k,k') = Q(k,k')\Big(\frac{p(k)k}{\overline{k}\gamma(k)}\Big)\Big(\frac{p(k')k'}{\overline{k}\gamma(k')}\Big), \qquad \gamma(k) = \sum_{k'} Q(k,k')p(k')\frac{k'}{\overline{k}\gamma(k')}$$

- transform $Q(k,k') \to \mu(k)Q(k,k')\mu(k')$:

$$\Pi(k,k') \to Q(k,k')\Big(\frac{p(k)k}{\overline{k}\tilde{\gamma}(k)}\Big)\Big(\frac{p(k')k'}{\overline{k}\tilde{\gamma}(k')}\Big), \qquad \tilde{\gamma}(k) = \sum_{k'} Q(k,k')p(k')\frac{k}{\overline{k}\tilde{\gamma}(k')}$$

$$\tilde{\gamma}(k) = \gamma(k)/\mu(k)$$

$\Pi(k,k')$ **invariant**!

- choose: $Q(k,k') = \Pi(k,k')kk'/\overline{k}^2$

$$\Big(\frac{\gamma(k)}{k/\overline{k}}\Big)W(k) = \sum_{k'} W(k,k')\Big(\frac{k'/\overline{k}}{\gamma(k')}\Big)$$

solution trivial!

Canonical kernel

can solve SP eqn **analytically**!

- Calculate $\Pi(k,k')$:

$$\Pi(k,k') = Q(k,k')\Big(\frac{p(k)k}{\overline{k}\gamma(k)}\Big)\Big(\frac{p(k')k'}{\overline{k}\gamma(k')}\Big), \qquad \gamma(k) = \sum_{k'} Q(k,k')p(k')\frac{k'}{\overline{k}\gamma(k')}$$

- transform $Q(k,k') \to \mu(k)Q(k,k')\mu(k')$:

$$\Pi(k,k') \to Q(k,k')\Big(\frac{p(k)k}{\overline{k}\tilde{\gamma}(k)}\Big)\Big(\frac{p(k')k'}{\overline{k}\tilde{\gamma}(k')}\Big), \qquad \tilde{\gamma}(k) = \sum_{k'} Q(k,k')p(k')\frac{k}{\overline{k}\tilde{\gamma}(k')}$$

$$\tilde{\gamma}(k) = \gamma(k)/\mu(k)$$

$\Pi(k,k')$ **invariant**!

- choose: $Q(k,k') = \Pi(k,k')kk'/\overline{k}^2$

$$\Big(\frac{\gamma(k)}{k/\overline{k}}\Big)W(k) = \sum_{k'} W(k,k')\Big(\frac{k'/\overline{k}}{\gamma(k')}\Big)$$

solution trivial!

Canonical kernel

can solve SP eqn **analytically**!

- Calculate $\Pi(k,k')$:

$$\Pi(k,k') = Q(k,k')\Big(\frac{p(k)k}{\overline{k}\gamma(k)}\Big)\Big(\frac{p(k')k'}{\overline{k}\gamma(k')}\Big), \qquad \gamma(k) = \sum_{k'} Q(k,k')p(k')\frac{k'}{\overline{k}\gamma(k')}$$

- transform $Q(k,k') \to \mu(k)Q(k,k')\mu(k')$:

$$\Pi(k,k') \to Q(k,k')\Big(\frac{p(k)k}{\overline{k}\tilde{\gamma}(k)}\Big)\Big(\frac{p(k')k'}{\overline{k}\tilde{\gamma}(k')}\Big), \qquad \tilde{\gamma}(k) = \sum_{k'} Q(k,k')p(k')\frac{k}{\overline{k}\tilde{\gamma}(k')}$$

$$\tilde{\gamma}(k) = \gamma(k)/\mu(k)$$

$\Pi(k,k')$ **invariant**!

- choose: $Q(k,k') = \Pi(k,k')kk'/\overline{k}^2$

$$\Big(\frac{\gamma(k)}{k/\overline{k}}\Big)W(k) = \sum_{k'} W(k,k')\Big(\frac{k'/\overline{k}}{\gamma(k')}\Big)$$

solution trivial!

Canonical kernel

can solve SP eqn **analytically**!

- Calculate $\Pi(k,k')$:

$$\Pi(k,k') = Q(k,k')\left(\frac{p(k)k}{\bar{k}\gamma(k)}\right)\left(\frac{p(k')k'}{\bar{k}\gamma(k')}\right), \qquad \gamma(k) = \sum_{k'} Q(k,k')p(k')\frac{k'}{\bar{k}\gamma(k')}$$

- transform $Q(k,k') \to \mu(k)Q(k,k')\mu(k')$:

$$\Pi(k,k') \to Q(k,k')\left(\frac{p(k)k}{\bar{k}\tilde{\gamma}(k)}\right)\left(\frac{p(k')k'}{\bar{k}\tilde{\gamma}(k')}\right), \qquad \tilde{\gamma}(k) = \sum_{k'} Q(k,k')p(k')\frac{k}{\bar{k}\tilde{\gamma}(k')}$$

$$\tilde{\gamma}(k) = \gamma(k)/\mu(k)$$

$\Pi(k,k')$ **invariant**!

- choose: $Q(k,k') = \Pi(k,k')kk'/\bar{k}^2$

$$\left(\frac{\gamma(k)}{k/\bar{k}}\right)W(k) = \sum_{k'} W(k,k')\left(\frac{k'/\bar{k}}{\gamma(k')}\right)$$

solution trivial!

Canonical kernel

can solve SP eqn **analytically**!

- Calculate $\Pi(k,k')$:

$$\Pi(k,k') = Q(k,k')\left(\frac{p(k)k}{\overline{k}\gamma(k)}\right)\left(\frac{p(k')k'}{\overline{k}\gamma(k')}\right), \qquad \gamma(k) = \sum_{k'} Q(k,k')p(k')\frac{k'}{\overline{k}\gamma(k')}$$

- transform $Q(k,k') \to \mu(k)Q(k,k')\mu(k')$:

$$\Pi(k,k') \to Q(k,k')\left(\frac{p(k)k}{\overline{k}\tilde{\gamma}(k)}\right)\left(\frac{p(k')k'}{\overline{k}\tilde{\gamma}(k')}\right), \qquad \tilde{\gamma}(k) = \sum_{k'} Q(k,k')p(k')\frac{k}{\overline{k}\tilde{\gamma}(k')}$$

$$\tilde{\gamma}(k) = \gamma(k)/\mu(k)$$

$\Pi(k,k')$ **invariant**!

- choose: $Q(k,k') = \Pi(k,k')kk'/\overline{k}^2$

$$\left(\frac{\gamma(k)}{k/\overline{k}}\right)W(k) = \sum_{k'} W(k,k')\left(\frac{k'/\overline{k}}{\gamma(k')}\right)$$

solution trivial!

Canonical kernel

can solve SP eqn **analytically**!

- Calculate $\Pi(k,k')$:

$$\Pi(k,k') = Q(k,k')\Big(\frac{p(k)k}{\overline{k}\gamma(k)}\Big)\Big(\frac{p(k')k'}{\overline{k}\gamma(k')}\Big), \qquad \gamma(k) = \sum_{k'} Q(k,k')p(k')\frac{k'}{\overline{k}\gamma(k')}$$

- transform $Q(k,k') \to \mu(k)Q(k,k')\mu(k')$:

$$\Pi(k,k') \to Q(k,k')\Big(\frac{p(k)k}{\overline{k}\tilde{\gamma}(k)}\Big)\Big(\frac{p(k')k'}{\overline{k}\tilde{\gamma}(k')}\Big), \qquad \tilde{\gamma}(k) = \sum_{k'} Q(k,k')p(k')\frac{k}{\overline{k}\tilde{\gamma}(k')}$$

$$\tilde{\gamma}(k) = \gamma(k)/\mu(k)$$

$\Pi(k,k')$ **invariant**!

- choose: $Q(k,k') = \Pi(k,k')kk'/\overline{k}^2$

$$\Big(\frac{\gamma(k)}{k/\overline{k}}\Big)W(k) = \sum_{k'} W(k,k')\Big(\frac{k'/\overline{k}}{\gamma(k')}\Big)$$

solution trivial!

Entropy and complexity - final result

$$S[p,\Pi] = -\frac{1}{N}\sum_{\mathbf{c}} p(\mathbf{c}|p,\Pi)\ \log p(\mathbf{c}|p,\Pi), \qquad \mathcal{N}[p,\Pi] = e^{NS[p,\Pi]}$$

Stat mech calculation:

$$S[p,\Pi] = S_0 - \mathcal{C}[p,\Pi] + o(1) \qquad (N\to\infty)$$

$$S_0 = \frac{1}{2}\langle k\rangle\left[\log[N/\overline{k}+1\right] \qquad (\mathrm{Erdos-Renyi\ entropy})$$

graph complexity:

$$\mathcal{C}[p,\Pi] = \underbrace{\sum_k p(k)\log\left[\frac{p(k)}{\pi(k)}\right]}_{\text{degree complexity}} + \underbrace{\frac{1}{2\overline{k}}\sum_{kk'} p(k)p(k')kk'\Pi(k,k')\log\Pi(k,k')}_{\text{degree correlation complexity}}$$

$$\pi(k) = e^{-\overline{k}}\overline{k}^k/k!$$

Entropy and complexity - final result

$$S[p,\Pi] = -\frac{1}{N}\sum_{\mathbf{c}} p(\mathbf{c}|p,\Pi)\ \log p(\mathbf{c}|p,\Pi), \qquad \mathcal{N}[p,\Pi] = e^{NS[p,\Pi]}$$

Stat mech calculation:

$$S[p,\Pi] = S_0 - \mathcal{C}[p,\Pi] + o(1) \qquad (N\to\infty)$$

$$S_0 = \frac{1}{2}\langle k\rangle\left[\log[N/\overline{k}+1]\right] \qquad (\text{Erdos}-\text{Renyi entropy})$$

graph complexity:

$$\mathcal{C}[p,\Pi] = \underbrace{\sum_k p(k)\log\left[\frac{p(k)}{\pi(k)}\right]}_{\text{degree complexity}} + \underbrace{\frac{1}{2\overline{k}}\sum_{kk'} p(k)p(k')kk'\Pi(k,k')\log\Pi(k,k')}_{\text{degree correlation complexity}}$$

$$\pi(k) = e^{-\overline{k}}\overline{k}^k/k!$$

Inform-theoretic distances

Quantifying structural distance between networks:
regard network as random realization of a graph with characteristics $\{p, \Pi\}$

$$
\begin{aligned}
D_{AB} &= \frac{1}{2N}\sum_{\mathbf{c}} p(\mathbf{c}|p_A,\Pi_A)\log\left[\frac{p(\mathbf{c}|p_A,\Pi_A)}{p(\mathbf{c}|p_B,\Pi_B)}\right] \\
&\quad + \frac{1}{2N}\sum_{\mathbf{c}} p(\mathbf{c}|p_B,\Pi_B)\log\left[\frac{p(\mathbf{c}|p_B,\Pi_B)}{p(\mathbf{c}|p_A,\Pi_A)}\right]
\end{aligned}
$$

stat mech result:

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

Inform-theoretic distances

Quantifying structural distance between networks:
regard network as random realization of a graph with characteristics $\{p,\Pi\}$

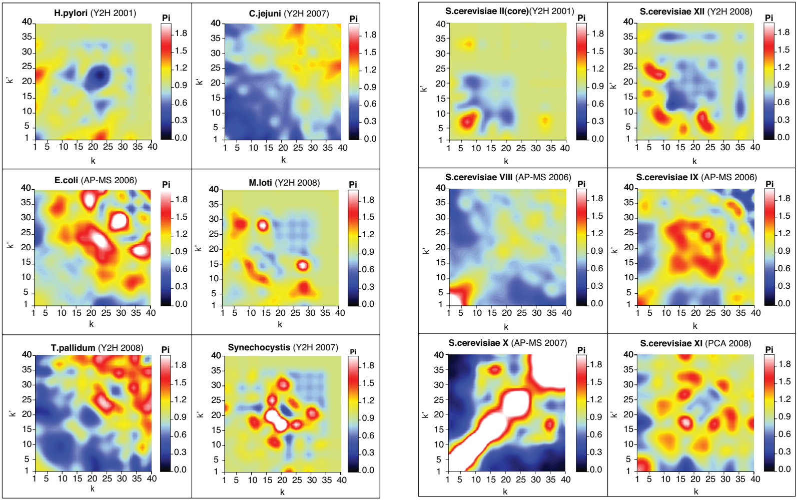
$$
\begin{aligned}
D_{AB} &= \frac{1}{2N}\sum_{\mathbf{c}} p(\mathbf{c}|p_A,\Pi_A)\log\left[\frac{p(\mathbf{c}|p_A,\Pi_A)}{p(\mathbf{c}|p_B,\Pi_B)}\right] \\
&\quad+\frac{1}{2N}\sum_{\mathbf{c}} p(\mathbf{c}|p_B,\Pi_B)\log\left[\frac{p(\mathbf{c}|p_B,\Pi_B)}{p(\mathbf{c}|p_A,\Pi_A)}\right]
\end{aligned}
$$

stat mech result:

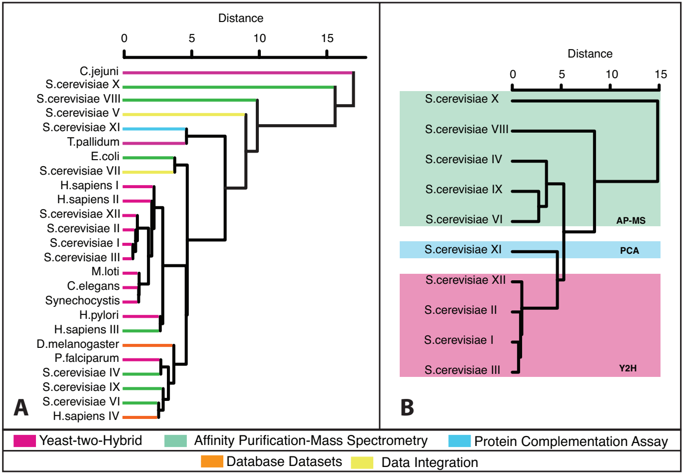
$$
\begin{aligned}
D_{AB} = {}& \frac{1}{2}\sum_k p_A(k)\log\left[\frac{p_A(k)}{p_B(k)}\right] + \sum_{kk'}\frac{p_A(k)p_A(k')kk'}{4\langle k\rangle_A}\Pi_A(k,k')\log\left[\frac{\Pi_A(k,k')}{\Pi_B(k,k')}\right] \\
& +\frac{1}{2}\sum_k p_B(k)\log\left[\frac{p_B(k)}{p_A(k)}\right] + \sum_{kk'}\frac{p_B(k)p_B(k')kk'}{4\langle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

Outline

PIN data reproducibility low (overlap),
problems with degree correlation patterns ...

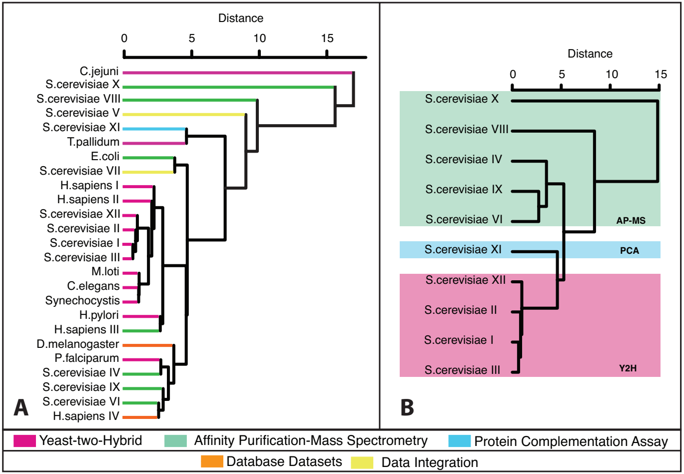


clustering of PIN datasets with inform-theoretic distance measure



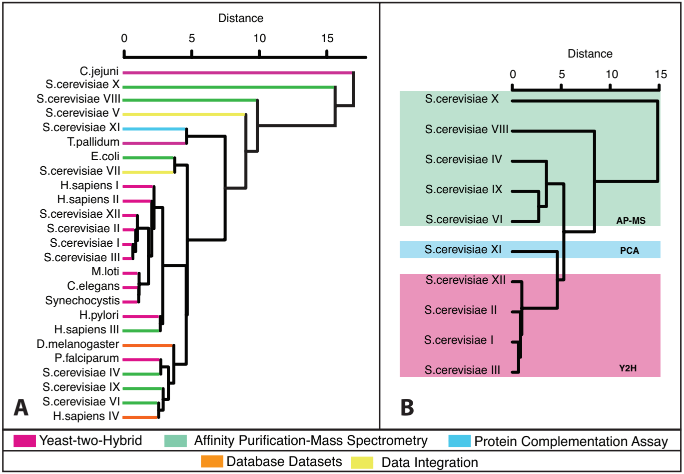
- PINs of same species and measured via same experimental method are statistically similar (in spite of limited overlap)
- PINs measured via same method cluster together, revealing **bias introduced by experimental method** that overrules species information

clustering of PIN datasets with inform-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

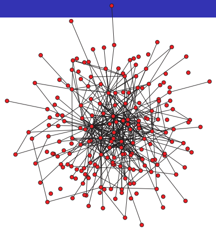
clustering of PIN datasets with inform-theoretic distance measure



- 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

Outline

Generating tailored random graphs



$G[\mathbf{k}]$: all size-N graphs with degrees $\mathbf{k}$
How to generate $\mathbf{c} \in G[\mathbf{k}]$ from ensembles
$p(\mathbf{c}|\mathbf{k})$ or $p(\mathbf{c}|\mathbf{k}, \Pi)$ numerically?

– sampling all graphs in $G[\mathbf{k}]$: in principle easy
– main difficulty: sampling with **correct probabilities**

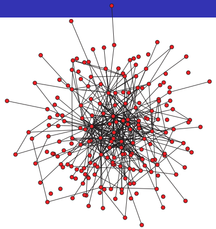
common method to generate graphs with degrees $\mathbf{k}$:

- construct ad hoc graph with $\mathbf{k}$
- shuffle links (randomize)
  via 'edge swaps'

randomly drawn edge swaps: ergodic on $G[\mathbf{k}]$
often *assumed* to give uniform probabilities, i.e. $p(\mathbf{c}|\mathbf{k})$ ...

– is this true?
– how to generate graphs from e.g. $p(\mathbf{c}|\mathbf{k}, \Pi)$?

Generating tailored random graphs



$G[\mathbf{k}]$: all size-N graphs with degrees $\mathbf{k}$
How to generate $\mathbf{c} \in G[\mathbf{k}]$ from ensembles
$p(\mathbf{c}|\mathbf{k})$ or $p(\mathbf{c}|\mathbf{k}, \Pi)$ numerically?

– sampling all graphs in $G[\mathbf{k}]$: in principle easy
– main difficulty: sampling with **correct probabilities**

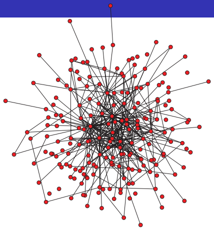
common method to generate graphs with degrees $\mathbf{k}$:

- construct ad hoc graph with $\mathbf{k}$
- shuffle links (randomize) via 'edge swaps'

randomly drawn edge swaps: ergodic on $G[\mathbf{k}]$
often *assumed* to give uniform probabilities, i.e. $p(\mathbf{c}|\mathbf{k})$ ...

– is this true?
– how to generate graphs from e.g. $p(\mathbf{c}|\mathbf{k}, \Pi)$?

Generating tailored random graphs



$G[\mathbf{k}]$: all size-N graphs with degrees $\mathbf{k}$
How to generate $\mathbf{c} \in G[\mathbf{k}]$ from ensembles
$p(\mathbf{c}|\mathbf{k})$ or $p(\mathbf{c}|\mathbf{k}, \Pi)$ numerically?

– sampling all graphs in $G[\mathbf{k}]$: in principle easy
– main difficulty: sampling with **correct probabilities**

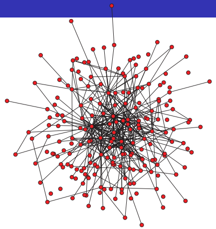
common method to generate graphs with degrees $\mathbf{k}$:

- construct ad hoc graph with $\mathbf{k}$
- shuffle links (randomize)
  via 'edge swaps'

randomly drawn edge swaps: ergodic on $G[\mathbf{k}]$
often *assumed* to give uniform probabilities, i.e. $p(\mathbf{c}|\mathbf{k})$ ...

– is this true?
– how to generate graphs from e.g. $p(\mathbf{c}|\mathbf{k}, \Pi)$?

Generating tailored random graphs

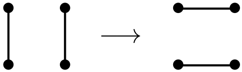


$G[\mathbf{k}]$: all size-N graphs with degrees $\mathbf{k}$
How to generate $\mathbf{c} \in G[\mathbf{k}]$ from ensembles
$p(\mathbf{c}|\mathbf{k})$ or $p(\mathbf{c}|\mathbf{k}, \Pi)$ numerically?

– sampling all graphs in $G[\mathbf{k}]$: in principle easy
– main difficulty: sampling with **correct probabilities**

common method to generate graphs with degrees $\mathbf{k}$:

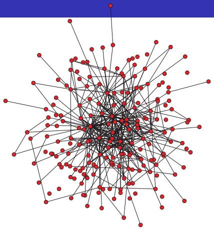
- construct ad hoc graph with $\mathbf{k}$
- shuffle links (randomize) via 'edge swaps'



randomly drawn edge swaps: ergodic on $G[\mathbf{k}]$
often *assumed* to give uniform probabilities, i.e. $p(\mathbf{c}|\mathbf{k})$ ...

– is this true?
– how to generate graphs from e.g. $p(\mathbf{c}|\mathbf{k}, \Pi)$?

Generating tailored random graphs

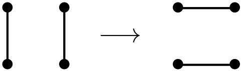


$G[\mathbf{k}]$: all size-N graphs with degrees $\mathbf{k}$
How to generate $\mathbf{c} \in G[\mathbf{k}]$ from ensembles
$p(\mathbf{c}|\mathbf{k})$ or $p(\mathbf{c}|\mathbf{k}, \Pi)$ numerically?

– sampling all graphs in $G[\mathbf{k}]$: in principle easy
– main difficulty: sampling with **correct probabilities**

common method to generate graphs with degrees $\mathbf{k}$:

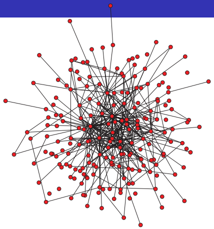
- construct ad hoc graph with $\mathbf{k}$
- shuffle links (randomize)
  via 'edge swaps'



randomly drawn edge swaps: ergodic on $G[\mathbf{k}]$
often *assumed* to give uniform probabilities, i.e. $p(\mathbf{c}|\mathbf{k})$ ...

– is this true?
– how to generate graphs from e.g. $p(\mathbf{c}|\mathbf{k}, \Pi)$?

Generating tailored random graphs

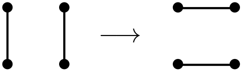


$G[\mathbf{k}]$: all size-N graphs with degrees $\mathbf{k}$
How to generate $\mathbf{c} \in G[\mathbf{k}]$ from ensembles
$p(\mathbf{c}|\mathbf{k})$ or $p(\mathbf{c}|\mathbf{k}, \Pi)$ numerically?

– sampling all graphs in $G[\mathbf{k}]$: in principle easy
– main difficulty: sampling with **correct probabilities**

common method to generate graphs with degrees $\mathbf{k}$:

- construct ad hoc graph with $\mathbf{k}$
- shuffle links (randomize) via 'edge swaps'



randomly drawn edge swaps: ergodic on $G[\mathbf{k}]$
often *assumed* to give uniform probabilities, i.e. $p(\mathbf{c}|\mathbf{k})$ ...

– is this true?
– how to generate graphs from e.g. $p(\mathbf{c}|\mathbf{k}, \Pi)$?

Outline

1. Background and origin of the problem
   - Protein interaction and gene regulation networks
   - Nonequilibrium stat mech of cellular signalling
   - Quantify graph topology beyond degrees
2. Analysis of tailored random graph ensembles
   - Canonical graph ensemble with controlled $\{p, \Pi\}$
   - Statistical mechanical analysis - entropy and beyond
   - Application to real PIN data
3. Generating tailored random graphs numerically
   - Degree-constrained 'shuffling'
   - **Formulation as a stochastic process**
   - Graph mobility
4. Ongoing work
5. Summary

Generating graphs - Monte-Carlo

How to generate graphs from $G[\mathbf{k}]$
with probabilities $p(\mathbf{c})$ numerically?

- stochastic dynamics a la Monte Carlo dynamics in spin systems
elementary moves: edge swaps (conserve all degrees)

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

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

Φ: set of all edge swaps

given state $\mathbf{c}'$:

– propose an allowed 'edge swap' F with prob $q(F|\mathbf{c}')$
– carry out $\mathbf{c} \to \mathbf{c}' = F\mathbf{c}$ with acceptance probability $A(\mathbf{c}'|\mathbf{c})$
– repeat until equilibrium

Generating graphs - Monte-Carlo

How to generate graphs from $G[\mathbf{k}]$
with probabilities $p(\mathbf{c})$ numerically?

- stochastic dynamics a la Monte Carlo dynamics in spin systems
  elementary moves: edge swaps (conserve all degrees)

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

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

Φ: set of all edge swaps

given state $\mathbf{c}'$:

– propose an allowed 'edge swap' F with prob $q(F|\mathbf{c}')$
– carry out $\mathbf{c} \to \mathbf{c}' = F\mathbf{c}$ with acceptance probability $A(\mathbf{c}'|\mathbf{c})$
– repeat until equilibrium

Generating graphs - Monte-Carlo

How to generate graphs from $G[\mathbf{k}]$
with probabilities $p(\mathbf{c})$ numerically?

- stochastic dynamics a la Monte Carlo dynamics in spin systems
  elementary moves: edge swaps (conserve all degrees)

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

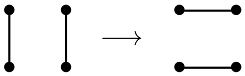
$$W(\mathbf{c}|\mathbf{c}') = \sum_{F\in\Phi} q(F|\mathbf{c}')\Big[\delta_{\mathbf{c},F\mathbf{c}'} A(F\mathbf{c}'|\mathbf{c}') + \delta_{\mathbf{c},\mathbf{c}'}[1 - A(F\mathbf{c}'|\mathbf{c}')]\Big]$$

Φ: set of all edge swaps

given state $\mathbf{c}'$:

– propose an allowed 'edge swap' F with prob $q(F|\mathbf{c}')$
– carry out $\mathbf{c} \to \mathbf{c}' = F\mathbf{c}$ with acceptance probability $A(\mathbf{c}'|\mathbf{c})$
– repeat until equilibrium

difference with e.g. MC dynamics in spin systems:



not all proposed 'edge swaps' are allowed ...
nr of possible moves depends on current state $\mathbf{c}$
(similar to e.g. constrained dynamics in spin chains)

$I_F(\mathbf{c}) = 1$ if F can act on $\mathbf{c}$,
zero otherwise

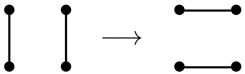
$$n(\mathbf{c}) = \sum_{F\in\Phi} I_F(\mathbf{c}) \qquad \text{mobility of } \mathbf{c}$$

process to converge to $p_\infty(\mathbf{c}) = \frac{1}{Z} e^{-H(\mathbf{c})}$:

$$\begin{aligned} q(F|\mathbf{c}) &= I_F(\mathbf{c})/n(\mathbf{c}) \\ A(\mathbf{c}|\mathbf{c}') &= \frac{n(\mathbf{c}')\mathrm{e}^{-\frac{1}{2}[H(\mathbf{c})-H(\mathbf{c}')]}}{n(\mathbf{c}')e^{-\frac{1}{2}[H(\mathbf{c})-H(\mathbf{c}')]} + n(\mathbf{c})\mathrm{e}^{\frac{1}{2}[H(\mathbf{c})-H(\mathbf{c}')]}} \end{aligned}$$

entropic effect reflecting the mobility of graph $\mathbf{c}$
naive accept-all edge-swapping: $p_\infty(\mathbf{c}) = n(\mathbf{c})/\sum_{\mathbf{c}'\in G[\mathbf{k}]} n(\mathbf{c}')$ ...

difference with e.g. MC dynamics in spin systems:



not all proposed 'edge swaps' are allowed ...
nr of possible moves depends on current state $\mathbf{c}$
(similar to e.g. constrained dynamics in spin chains)

$I_F(\mathbf{c}) = 1$ if F can act on $\mathbf{c}$,
zero otherwise

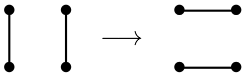
$$n(\mathbf{c}) = \sum_{F\in\Phi} I_F(\mathbf{c}) \qquad \text{mobility of } \mathbf{c}$$

process to converge to $p_\infty(\mathbf{c}) = \frac{1}{Z}e^{-H(\mathbf{c})}$:

$$\begin{aligned} q(F|\mathbf{c}) &= I_F(\mathbf{c})/n(\mathbf{c}) \\ A(\mathbf{c}|\mathbf{c}') &= \frac{n(\mathbf{c}')\mathrm{e}^{-\frac{1}{2}[H(\mathbf{c})-H(\mathbf{c}')]}}{n(\mathbf{c}')e^{-\frac{1}{2}[H(\mathbf{c})-H(\mathbf{c}')]} + n(\mathbf{c})\mathrm{e}^{\frac{1}{2}[H(\mathbf{c})-H(\mathbf{c}')]}} \end{aligned}$$

entropic effect reflecting the mobility of graph $\mathbf{c}$
naive accept-all edge-swapping: $p_\infty(\mathbf{c}) = n(\mathbf{c})/\sum_{\mathbf{c}'\in G[\mathbf{k}]} n(\mathbf{c}')$ ...

difference with e.g. MC dynamics in spin systems:



not all proposed 'edge swaps' are allowed ...
nr of possible moves depends on current state **c**
(similar to e.g. constrained dynamics in spin chains)

$I_F(\mathbf{c}) = 1$ if F can act on $\mathbf{c}$,
zero otherwise

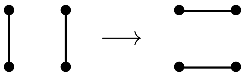
$$n(\mathbf{c}) = \sum_{F\in\Phi} I_F(\mathbf{c}) \qquad \text{mobility of } \mathbf{c}$$

process to converge to $p_\infty(\mathbf{c}) = \frac{1}{Z} e^{-H(\mathbf{c})}$:

$$\begin{aligned} q(F|\mathbf{c}) &= I_F(\mathbf{c})/n(\mathbf{c}) \\ A(\mathbf{c}|\mathbf{c}') &= \frac{n(\mathbf{c}')\mathrm{e}^{-\frac{1}{2}[H(\mathbf{c})-H(\mathbf{c}')]}}{n(\mathbf{c}')e^{-\frac{1}{2}[H(\mathbf{c})-H(\mathbf{c}')]} + n(\mathbf{c})\mathrm{e}^{\frac{1}{2}[H(\mathbf{c})-H(\mathbf{c}')]}} \end{aligned}$$

entropic effect reflecting the mobility of graph **c**
naive accept-all edge-swapping: $p_\infty(\mathbf{c}) = n(\mathbf{c})/\sum_{\mathbf{c}'\in G[\mathbf{k}]} n(\mathbf{c}')$ ...

difference with e.g. MC dynamics in spin systems:



not all proposed ‘edge swaps’ are allowed ...
nr of possible moves depends on current state $\mathbf{c}$
(similar to e.g. constrained dynamics in spin chains)

$I_F(\mathbf{c}) = 1$ if F can act on $\mathbf{c}$,
zero otherwise

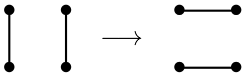
$$n(\mathbf{c}) = \sum_{F\in\Phi} I_F(\mathbf{c}) \qquad \text{mobility of } \mathbf{c}$$

process to converge to $p_\infty(\mathbf{c}) = \frac{1}{Z} e^{-H(\mathbf{c})}$:

$$\begin{aligned} q(F|\mathbf{c}) &= I_F(\mathbf{c})/n(\mathbf{c}) \\ A(\mathbf{c}|\mathbf{c}') &= \frac{n(\mathbf{c}')\mathrm{e}^{-\frac{1}{2}[H(\mathbf{c})-H(\mathbf{c}')]}}{n(\mathbf{c}')e^{-\frac{1}{2}[H(\mathbf{c})-H(\mathbf{c}')]} + n(\mathbf{c})\mathrm{e}^{\frac{1}{2}[H(\mathbf{c})-H(\mathbf{c}')]}} \end{aligned}$$

entropic effect reflecting the mobility of graph $\mathbf{c}$
naive accept-all edge-swapping: $p_\infty(\mathbf{c}) = n(\mathbf{c})/\sum_{\mathbf{c}'\in G[\mathbf{k}]} n(\mathbf{c}')$ ...

difference with e.g. MC dynamics in spin systems:



not all proposed ‘edge swaps’ are allowed ...
nr of possible moves depends on current state $\mathbf{c}$
(similar to e.g. constrained dynamics in spin chains)

$I_F(\mathbf{c}) = 1$ if F can act on $\mathbf{c}$,
zero otherwise

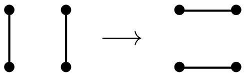
$$n(\mathbf{c}) = \sum_{F\in\Phi} I_F(\mathbf{c}) \qquad \text{mobility of } \mathbf{c}$$

process to converge to $p_\infty(\mathbf{c}) = \frac{1}{Z}e^{-H(\mathbf{c})}$:

$$\begin{aligned} q(F|\mathbf{c}) &= I_F(\mathbf{c})/n(\mathbf{c}) \\ A(\mathbf{c}|\mathbf{c}') &= \frac{n(\mathbf{c}')\mathrm{e}^{-\frac{1}{2}[H(\mathbf{c})-H(\mathbf{c}')]}}{n(\mathbf{c}')e^{-\frac{1}{2}[H(\mathbf{c})-H(\mathbf{c}')]} + n(\mathbf{c})\mathrm{e}^{\frac{1}{2}[H(\mathbf{c})-H(\mathbf{c}')]}} \end{aligned}$$

entropic effect reflecting the mobility of graph $\mathbf{c}$
naive accept-all edge-swapping: $p_\infty(\mathbf{c}) = n(\mathbf{c})/\sum_{\mathbf{c}'\in G[\mathbf{k}]} n(\mathbf{c}')$ ...

difference with e.g. MC dynamics in spin systems:



not all proposed 'edge swaps' are allowed ...
nr of possible moves depends on current state $\mathbf{c}$
(similar to e.g. constrained dynamics in spin chains)

$I_F(\mathbf{c}) = 1$ if F can act on $\mathbf{c}$,
zero otherwise

$$n(\mathbf{c}) = \sum_{F\in\Phi} I_F(\mathbf{c}) \qquad \text{mobility of } \mathbf{c}$$

process to converge to $p_\infty(\mathbf{c}) = \frac{1}{Z}e^{-H(\mathbf{c})}$:

$$\begin{aligned} q(F|\mathbf{c}) &= I_F(\mathbf{c})/n(\mathbf{c}) \\ A(\mathbf{c}|\mathbf{c}') &= \frac{n(\mathbf{c}')\mathrm{e}^{-\frac{1}{2}[H(\mathbf{c})-H(\mathbf{c}')]}}{n(\mathbf{c}')e^{-\frac{1}{2}[H(\mathbf{c})-H(\mathbf{c}')]} + n(\mathbf{c})\mathrm{e}^{\frac{1}{2}[H(\mathbf{c})-H(\mathbf{c}')]}} \end{aligned}$$

entropic effect reflecting the mobility of graph $\mathbf{c}$
naive accept-all edge-swapping: $p_\infty(\mathbf{c}) = n(\mathbf{c})/\sum_{\mathbf{c}'\in G[\mathbf{k}]} n(\mathbf{c}')$ ...

Outline

1. Background and origin of the problem
   - Protein interaction and gene regulation networks
   - Nonequilibrium stat mech of cellular signalling
   - Quantify graph topology beyond degrees
2. Analysis of tailored random graph ensembles
   - Canonical graph ensemble with controlled $\{p, \Pi\}$
   - Statistical mechanical analysis - entropy and beyond
   - Application to real PIN data
3. Generating tailored random graphs numerically
   - Degree-constrained ‘shuffling’
   - Formulation as a stochastic process
   - Graph mobility
4. Ongoing work
5. Summary

combinatorics of admissible edge swaps

$Q=\{(i,j,k,\ell)\in\{1,\ldots,N\}^4|\ i<j<k<\ell\}$, 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 $\alpha\in\{1,2,3\}$

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

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

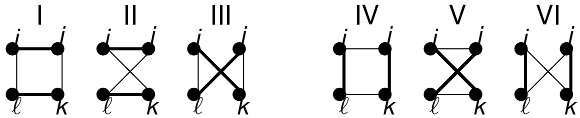
$$
\begin{aligned}
F_{ijk\ell;\alpha}(\mathbf{c})_{qr} &= 1-c_{qr} && \text{for } (q,r)\in\mathcal{S}_{ijk\ell;\alpha}\\
F_{ijk\ell;\alpha}(\mathbf{c})_{qr} &= c_{qr} && \text{for } (q,r)\notin\mathcal{S}_{ijk\ell;\alpha}
\end{aligned}
$$

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

Calculation of graph mobility

combinatorics of admissible edge swaps
$Q = \{(i,j,k,\ell) \in \{1,\ldots,N\}^4 |\ i<j<k<\ell\}$, 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 $\alpha \in \{1,2,3\}$

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

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

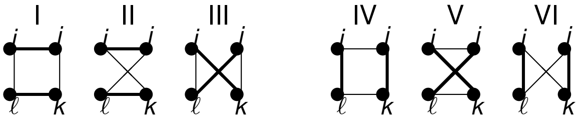
$$
\begin{aligned}
F_{ijk\ell;\alpha}(\mathbf{c})_{qr} &= 1 - c_{qr} \quad \text{for } (q,r) \in \mathcal{S}_{ijk\ell;\alpha} \\
F_{ijk\ell;\alpha}(\mathbf{c})_{qr} &= c_{qr} \quad \text{for } (q,r) \notin \mathcal{S}_{ijk\ell;\alpha}
\end{aligned}
$$

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

Calculation of graph mobility

combinatorics of admissible edge swaps
$Q=\{(i,j,k,\ell)\in\{1,\ldots,N\}^4|\ i<j<k<\ell\}$, 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 $\alpha\in\{1,2,3\}$

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

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

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

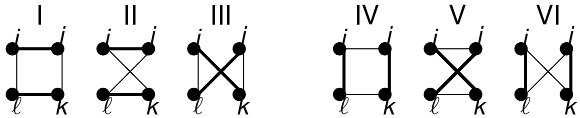
$$
S_{ijk\ell;1}=\{(i,j),(k,\ell),(i,\ell),(j,k)\},\qquad S_{ijk\ell;2}=\{(i,j),(k,\ell),(i,k),(j,\ell)\}
$$
$$
S_{ijk\ell;3}=\{(i,k),(j,\ell),(i,\ell),(j,k)\}
$$

Calculation of graph mobility

combinatorics of admissible edge swaps

$Q=\{(i,j,k,\ell)\in\{1,\ldots,N\}^4|\ i<j<k<\ell\}$, 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 $\alpha\in\{1,2,3\}$

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

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

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

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

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

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

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

consequence:

general and exact Monte-Carlo recipe
for generating random graphs with
prescribed degrees $(k_1, \ldots, k_N)$
and arbitrary measures

$$p(\mathbf{c}) = \frac{e^{-H(\mathbf{c})} \prod_i \delta_{k_i, \sum_j c_{ij}}}{\sum_{\mathbf{c}'} e^{-H(\mathbf{c}')} \prod_i \delta_{k_i, \sum_j c'_{ij}}}$$

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

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

consequence:

general and exact Monte-Carlo recipe
for generating random graphs with
prescribed degrees $(k_1, \ldots, k_N)$
and arbitrary measures

$$p(\mathbf{c}) = \frac{e^{-H(\mathbf{c})} \prod_i \delta_{k_i, \sum_j c_{ij}}}{\sum_{\mathbf{c}'} e^{-H(\mathbf{c}')} \prod_i \delta_{k_i, \sum_j c'_{ij}}}$$

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

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

consequence:

general and exact Monte-Carlo recipe
for generating random graphs with
prescribed degrees $(k_1, \ldots, k_N)$
and arbitrary measures

$$p(\mathbf{c}) = \frac{e^{-H(\mathbf{c})} \prod_i \delta_{k_i, \sum_j c_{ij}}}{\sum_{\mathbf{c}'} e^{-H(\mathbf{c}')} \prod_i \delta_{k_i, \sum_j c'_{ij}}}$$

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

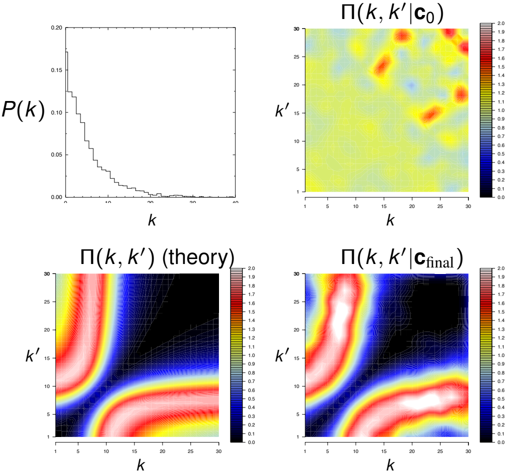
$$n(\mathbf{c}) = \underbrace{\frac{1}{4}\Big(\sum_i k_i\Big)^2 + \frac{1}{4}\sum_i k_i - \frac{1}{2}\sum_i k_i^2}_{invariant} + \underbrace{\frac{1}{4}\mathrm{Tr}(\mathbf{c}^4) + \frac{1}{2}\mathrm{Tr}(\mathbf{c}^3) - \frac{1}{2}\sum_{ij} k_i c_{ij} k_j}_{state\ dependent}$$

consequence:

general and exact Monte-Carlo recipe
for generating random graphs with
prescribed degrees $(k_1, \ldots, k_N)$
and arbitrary measures

$$p(\mathbf{c}) = \frac{e^{-H(\mathbf{c})} \prod_i \delta_{k_i, \sum_j c_{ij}}}{\sum_{\mathbf{c}'} e^{-H(\mathbf{c}')} \prod_i \delta_{k_i, \sum_j c'_{ij}}}$$

An example



$N = 4000$,
$\langle k \rangle = 5$

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

Ongoing work

- very easy:
  directed networks, $c_{ij} \neq c_{ji}$

$$S = \langle k\rangle[1+\log(N/\langle k\rangle)] - \sum_{k_{\rm in} k_{\rm out}} p(k_{\rm in}, k_{\rm out}) \log\left[\frac{p(k_{\rm in}, k_{\rm out})}{\pi(k_{\rm in})\pi(k_{\rm out})}\right] + \ldots$$

- easy:
  effects of sampling a fraction $x \in [0,1]$ of the nodes

$$p(k|x) = \sum_{k'\geq k} p(k') \binom{k'}{k'-k} (1-x)^{k'-k} x^k$$

$$W(k,k'|x) = \sum_{q\geq k} \sum_{q'\geq k'} W(q,q') \frac{(q-1)!}{(k-1)!(q-k)!} \frac{(q'-1)!}{(k'-1)!(q'-k')!} \times (1-x)^{q-k} x^{k-1} (1-x)^{q'-k'} x^{k'-1}$$

Ongoing work

- very easy:
  directed networks, $c_{ij} \neq c_{ji}$

$$S = \langle k\rangle[1+\log(N/\langle k\rangle)] - \sum_{k_{\rm in}k_{\rm out}} p(k_{\rm in},k_{\rm out})\log\left[\frac{p(k_{\rm in},k_{\rm out})}{\pi(k_{\rm in})\pi(k_{\rm out})}\right] + \ldots$$

- easy:
  effects of sampling a fraction $x \in [0,1]$ of the nodes

$$p(k|x) = \sum_{k'\geq k} p(k')\binom{k'}{k'-k}(1-x)^{k'-k}x^k$$

$$W(k,k'|x) = \sum_{q\geq k}\sum_{q'\geq k'} W(q,q')\frac{(q-1)!}{(k-1)!(q-k)!}\frac{(q'-1)!}{(k'-1)!(q'-k')!} \times(1-x)^{q-k}x^{k-1}(1-x)^{q'-k'}x^{k'-1}$$

Ongoing work

- very easy:
  directed networks, $c_{ij} \neq c_{ji}$

$$S = \langle k \rangle [1 + \log(N/\langle k \rangle)] - \sum_{k_{\rm in} k_{\rm out}} p(k_{\rm in}, k_{\rm out}) \log \left[\frac{p(k_{\rm in}, k_{\rm out})}{\pi(k_{\rm in})\pi(k_{\rm out})} \right] + \ldots$$

- easy:
  effects of sampling a fraction $x \in [0, 1]$ of the nodes

$$p(k|x) = \sum_{k' \geq k} p(k') \binom{k'}{k' - k} (1-x)^{k'-k} x^k$$

$$W(k, k'|x) = \sum_{q \geq k} \sum_{q' \geq k'} W(q, q') \frac{(q-1)!}{(k-1)!(q-k)!} \frac{(q'-1)!}{(k'-1)!(q'-k')!} \times (1-x)^{q-k} x^{k-1} (1-x)^{q'-k'} x^{k'-1}$$

- intermediate:
  ensembles with hidden variables

$$p(\mathbf{c}|\mathbf{k},\mathbf{x}) \;=\; Z^{-1}\prod_i \delta_{k_i,k_i(\mathbf{c})}\prod_{i<j}\Big[\frac{\overline{k}}{N}Q(x_i,x_j)\delta_{c_{ij},1}+(1-\frac{\overline{k}}{N}Q(x_i,x_j))\delta_{c_{ij},0}\Big]$$

$$S=\frac{1}{2}\overline{k}[1+\log(N/\overline{k})]+\sum_k p(k)\log p(k)-\sum_k p(k)\log[p(k)/\pi(k)]$$
$$-\frac{1}{2}\overline{k}\sum_{xx'}W(x,x')\log[W(x,x')/W(x)W(x')]+\epsilon_N$$

$$Q(x,x')=W(x,x')/p(x)p(x'),\qquad W(x,x')=\frac{1}{N\langle k\rangle}\sum_{ij}c_{ij}\delta_{x,x_i}\delta_{x',x_j}$$

most probable $\mathbf{x}$ to explain $\mathbf{c}$?
maximize for $x\in D$:

$$\Omega[\mathbf{x}]=\overbrace{\sum_{xx'}W(x,x'|\mathbf{c})\log\Big[\frac{W(x,x'|\mathbf{c})}{W(x|\mathbf{c})W(x'|\mathbf{c})}\Big]}^{evidence\ for\ structure}-\overbrace{\frac{2}{\overline{k}}\log|D|}^{no\ overfitting}$$

- intermediate:
  ensembles with hidden variables

$$p(\mathbf{c}|\mathbf{k},\mathbf{x}) \;=\; Z^{-1}\prod_i \delta_{k_i,k_i(\mathbf{c})}\prod_{i<j}\Big[\frac{\overline{k}}{N}Q(x_i,x_j)\delta_{c_{ij},1}+(1-\frac{\overline{k}}{N}Q(x_i,x_j))\delta_{c_{ij},0}\Big]$$

$$\begin{aligned} S = \frac{1}{2}\overline{k}[1+\log(N/\overline{k})] + \sum_k p(k)\log p(k) - \sum_k p(k)\log[p(k)/\pi(k)] \\ -\frac{1}{2}\overline{k}\sum_{xx'} W(x,x')\log[W(x,x')/W(x)W(x')] + \epsilon_N \end{aligned}$$

$$Q(x,x') = W(x,x')/p(x)p(x'), \qquad W(x,x') = \frac{1}{N\langle k\rangle}\sum_{ij} c_{ij}\delta_{x,x_i}\delta_{x',x_j}$$

most probable $\mathbf{x}$ to explain $\mathbf{c}$?
maximize for $x \in D$:

$$\Omega[\mathbf{x}] = \overbrace{\sum_{xx'} W(x,x'|\mathbf{c})\log\Big[\frac{W(x,x'|\mathbf{c})}{W(x|\mathbf{c})W(x'|\mathbf{c})}\Big]}^{evidence\ for\ structure} - \overbrace{\frac{2}{\overline{k}}\log|D|}^{no\ overfitting}$$

- intermediate:
  ensembles with hidden variables

$$p(\mathbf{c}|\mathbf{k},\mathbf{x}) \quad = \quad Z^{-1}\prod_i \delta_{k_i,k_i(\mathbf{c})}\prod_{i<j}\Big[\frac{\overline{k}}{N}Q(x_i,x_j)\delta_{c_{ij},1}+(1-\frac{\overline{k}}{N}Q(x_i,x_j))\delta_{c_{ij},0}\Big]$$

$$\begin{aligned} S = \frac{1}{2}\overline{k}[1+\log(N/\overline{k})] + \sum_k p(k)\log p(k) - \sum_k p(k)\log[p(k)/\pi(k)] \\ -\frac{1}{2}\overline{k}\sum_{xx'} W(x,x')\log[W(x,x')/W(x)W(x')] + \epsilon_N \end{aligned}$$

$$Q(x,x') = W(x,x')/p(x)p(x'), \qquad W(x,x') = \frac{1}{N\langle k\rangle}\sum_{ij} c_{ij}\delta_{x,x_i}\delta_{x',x_j}$$

most probable **x** to explain **c**?
maximize for $x \in D$:

$$\Omega[\mathbf{x}] = \overbrace{\sum_{xx'} W(x,x'|\mathbf{c})\log\Big[\frac{W(x,x'|\mathbf{c})}{W(x|\mathbf{c})W(x'|\mathbf{c})}\Big]}^{evidence\ for\ structure} - \overbrace{\frac{2}{\overline{k}}\log|D|}^{no\ overfitting}$$

- hard:
  ensembles that involve **loops** ...

$$p(\mathbf{c})=\frac{e^{u\sum_{ij}c_{ij}+v\sum_{ijk}c_{ij}c_{jk}c_{ki}}}{Z(u,v)},\quad \overline{k}=\langle\frac{1}{N}\sum_{ij}c_{ij}\rangle,\quad \overline{m}=\langle\frac{1}{N}\sum_{ijk}c_{ij}c_{jk}c_{ki}\rangle$$

here

$$S=\phi(u,v)-\lim_{x\to 1}\frac{\partial}{\partial x}\phi(xu,xv)\qquad \overline{k}=\frac{\partial}{\partial u}\phi(u,v),\qquad \overline{m}=\frac{\partial}{\partial v}\phi(u,v)$$

$$\phi(u,v)=\frac{1}{N}\log\sum_{\mathbf{c}}e^{N\int d\mu\;(u\mu^2+v\mu^3)\varrho(\mu|\mathbf{c})}$$

spectrum formula

$$\varrho(\mu|\mathbf{c})\quad=\quad\frac{2}{N\pi}\lim_{\varepsilon\downarrow 0}\mathrm{Im}\frac{\partial}{\partial\mu}\log Z(\mu+i\varepsilon|\mathbf{c})\qquad Z(\mu|\mathbf{c})=\int d\phi\; e^{-\frac{1}{2}i\phi\cdot[\mathbf{c}-\mu I]\phi}$$

result

$$\phi(u,v)=\lim_{\varepsilon,\Delta\downarrow 0}\frac{1}{N}\log\sum_{\mathbf{c}}\prod_{\mu}\left[Z(\mu+i\varepsilon|\mathbf{c})^{-i}\overline{Z(\mu+i\varepsilon|\mathbf{c})}^{i}\right]^{\frac{\Delta}{\pi}(2u\mu+3v\mu^2)}$$

imaginary replicas?

- hard:
  ensembles that involve **loops** ...

$$p(\mathbf{c}) = \frac{\mathrm{e}^{u\sum_{ij} c_{ij} + v\sum_{ijk} c_{ij}c_{jk}c_{ki}}}{Z(u,v)}, \quad \overline{k} = \langle \frac{1}{N}\sum_{ij} c_{ij}\rangle, \quad \overline{m} = \langle \frac{1}{N}\sum_{ijk} c_{ij}c_{jk}c_{ki}\rangle$$

here

$$S = \phi(u,v) - \lim_{x\to 1}\frac{\partial}{\partial x}\phi(xu,xv) \qquad \overline{k} = \frac{\partial}{\partial u}\phi(u,v), \qquad \overline{m} = \frac{\partial}{\partial v}\phi(u,v)$$

$$\phi(u,v) = \frac{1}{N}\log\sum_{\mathbf{c}} \mathrm{e}^{N\int \mathrm{d}\mu\ (u\mu^2+v\mu^3)\varrho(\mu|\mathbf{c})}$$

spectrum formula

$$\varrho(\mu|\mathbf{c}) \quad = \quad \frac{2}{N\pi}\lim_{\varepsilon\downarrow 0}\mathrm{Im}\frac{\partial}{\partial\mu}\log Z(\mu+\mathrm{i}\varepsilon|\mathbf{c}) \qquad Z(\mu|\mathbf{c}) = \int d\phi\ e^{-\frac{1}{2}\mathrm{i}\phi\cdot[\mathbf{c}-\mu\mathrm{I}]\phi}$$

result

$$\phi(u,v) = \lim_{\varepsilon,\Delta\downarrow 0}\frac{1}{N}\log\sum_{\mathbf{c}}\prod_{\mu}\left[Z(\mu+\mathrm{i}\varepsilon|\mathbf{c})^{-\mathrm{i}}\overline{Z(\mu+\mathrm{i}\varepsilon|\mathbf{c})}^{\mathrm{i}}\right]^{\frac{\Delta}{\pi}(2u\mu+3v\mu^2)}$$

imaginary replicas?

- hard:
  ensembles that involve **loops** ...

$$p(\mathbf{c})=\frac{e^{u\sum_{ij}c_{ij}+v\sum_{ijk}c_{ij}c_{jk}c_{ki}}}{Z(u,v)},\qquad \overline{k}=\langle\frac{1}{N}\sum_{ij}c_{ij}\rangle,\quad \overline{m}=\langle\frac{1}{N}\sum_{ijk}c_{ij}c_{jk}c_{ki}\rangle$$

  here

$$S=\phi(u,v)-\lim_{x\to 1}\frac{\partial}{\partial x}\phi(xu,xv)\qquad \overline{k}=\frac{\partial}{\partial u}\phi(u,v),\qquad \overline{m}=\frac{\partial}{\partial v}\phi(u,v)$$

$$\phi(u,v)=\frac{1}{N}\log\sum_{\mathbf{c}}e^{N\int d\mu\ (u\mu^2+v\mu^3)\varrho(\mu|\mathbf{c})}$$

  spectrum formula

$$\varrho(\mu|\mathbf{c}) = \frac{2}{N\pi}\lim_{\varepsilon\downarrow 0}\mathrm{Im}\frac{\partial}{\partial\mu}\log Z(\mu+i\varepsilon|\mathbf{c})\qquad Z(\mu|\mathbf{c})=\int d\phi\ e^{-\frac{1}{2}i\phi\cdot[\mathbf{c}-\mu I]\phi}$$

  result

$$\phi(u,v)=\lim_{\varepsilon,\Delta\downarrow 0}\frac{1}{N}\log\sum_{\mathbf{c}}\prod_{\mu}\left[Z(\mu+i\varepsilon|\mathbf{c})^{-i}\overline{Z(\mu+i\varepsilon|\mathbf{c})}^{i}\right]^{\frac{\Delta}{\pi}(2u\mu+3v\mu^2)}$$

imaginary replicas?

- hard:
  ensembles that involve **loops** ...

$$p(\mathbf{c})=\frac{\mathrm{e}^{u\sum_{ij}c_{ij}+v\sum_{ijk}c_{ij}c_{jk}c_{ki}}}{Z(u,v)},\quad \overline{k}=\langle\frac{1}{N}\sum_{ij}c_{ij}\rangle,\quad \overline{m}=\langle\frac{1}{N}\sum_{ijk}c_{ij}c_{jk}c_{ki}\rangle$$

here

$$S=\phi(u,v)-\lim_{x\to 1}\frac{\partial}{\partial x}\phi(xu,xv)\qquad \overline{k}=\frac{\partial}{\partial u}\phi(u,v),\qquad \overline{m}=\frac{\partial}{\partial v}\phi(u,v)$$

$$\phi(u,v)=\frac{1}{N}\log\sum_{\mathbf{c}}\mathrm{e}^{N\int \mathrm{d}\mu\ (u\mu^2+v\mu^3)\varrho(\mu|\mathbf{c})}$$

spectrum formula

$$\varrho(\mu|\mathbf{c})\quad=\quad\frac{2}{N\pi}\lim_{\varepsilon\downarrow 0}\mathrm{Im}\frac{\partial}{\partial\mu}\log Z(\mu+\mathrm{i}\varepsilon|\mathbf{c})\qquad Z(\mu|\mathbf{c})=\int d\phi\ e^{-\frac{1}{2}\mathrm{i}\boldsymbol{\phi}\cdot[\mathbf{c}-\mu\mathrm{I}]\boldsymbol{\phi}}$$

result

$$\phi(u,v)=\lim_{\varepsilon,\Delta\downarrow 0}\frac{1}{N}\log\sum_{\mathbf{c}}\prod_{\mu}\left[Z(\mu+\mathrm{i}\varepsilon|\mathbf{c})^{-\mathrm{i}}\overline{Z(\mu+\mathrm{i}\varepsilon|\mathbf{c})}^{\mathrm{i}}\right]^{\frac{\Delta}{\pi}(2u\mu+3v\mu^2)}$$

imaginary replicas?

- hard:
ensembles that involve **loops** ...

$$p(\mathbf{c})=\frac{\mathrm{e}^{u\sum_{ij}c_{ij}+v\sum_{ijk}c_{ij}c_{jk}c_{ki}}}{Z(u,v)},\quad \overline{k}=\langle\frac{1}{N}\sum_{ij}c_{ij}\rangle,\quad \overline{m}=\langle\frac{1}{N}\sum_{ijk}c_{ij}c_{jk}c_{ki}\rangle$$

here

$$S=\phi(u,v)-\lim_{x\to 1}\frac{\partial}{\partial x}\phi(xu,xv)\qquad \overline{k}=\frac{\partial}{\partial u}\phi(u,v),\qquad \overline{m}=\frac{\partial}{\partial v}\phi(u,v)$$

$$\phi(u,v)=\frac{1}{N}\log\sum_{\mathbf{c}}\mathrm{e}^{N\int\mathrm{d}\mu\ (u\mu^2+v\mu^3)\varrho(\mu|\mathbf{c})}$$

spectrum formula

$$\varrho(\mu|\mathbf{c})\quad=\quad\frac{2}{N\pi}\lim_{\varepsilon\downarrow 0}\mathrm{Im}\frac{\partial}{\partial\mu}\log Z(\mu+\mathrm{i}\varepsilon|\mathbf{c})\qquad Z(\mu|\mathbf{c})=\int d\phi\ e^{-\frac{1}{2}\mathrm{i}\phi\cdot[\mathbf{c}-\mu\mathrm{I}]\phi}$$

result

$$\phi(u,v)=\lim_{\varepsilon,\Delta\downarrow 0}\frac{1}{N}\log\sum_{\mathbf{c}}\prod_{\mu}\left[Z(\mu+\mathrm{i}\varepsilon|\mathbf{c})^{-\mathrm{i}}\overline{Z(\mu+\mathrm{i}\varepsilon|\mathbf{c})}^{\mathrm{i}}\right]^{\frac{\Delta}{\pi}(2u\mu+3v\mu^2)}$$

imaginary replicas?

- hard:
  ensembles that involve **loops** ...

$$p(\mathbf{c}) = \frac{e^{u\sum_{ij} c_{ij} + v\sum_{ijk} c_{ij}c_{jk}c_{ki}}}{Z(u,v)}, \quad \overline{k} = \langle \frac{1}{N}\sum_{ij} c_{ij}\rangle, \quad \overline{m} = \langle \frac{1}{N}\sum_{ijk} c_{ij}c_{jk}c_{ki}\rangle$$

here

$$S = \phi(u,v) - \lim_{x\to 1}\frac{\partial}{\partial x}\phi(xu,xv) \qquad \overline{k} = \frac{\partial}{\partial u}\phi(u,v), \qquad \overline{m} = \frac{\partial}{\partial v}\phi(u,v)$$

$$\phi(u,v) = \frac{1}{N}\log\sum_{\mathbf{c}} e^{N\int d\mu\ (u\mu^2+v\mu^3)\varrho(\mu|\mathbf{c})}$$

spectrum formula

$$\varrho(\mu|\mathbf{c}) \quad = \quad \frac{2}{N\pi}\lim_{\varepsilon\downarrow 0}\mathrm{Im}\frac{\partial}{\partial\mu}\log Z(\mu+\mathrm{i}\varepsilon|\mathbf{c}) \qquad Z(\mu|\mathbf{c}) = \int d\phi\ e^{-\frac{1}{2}\mathrm{i}\phi\cdot[\mathbf{c}-\mu\mathrm{I}]\phi}$$

result

$$\phi(u,v) = \lim_{\varepsilon,\Delta\downarrow 0}\frac{1}{N}\log\sum_{\mathbf{c}}\prod_{\mu}\left[Z(\mu+\mathrm{i}\varepsilon|\mathbf{c})^{-\mathrm{i}}\overline{Z(\mu+\mathrm{i}\varepsilon|\mathbf{c})}^{\mathrm{i}}\right]^{\frac{\Delta}{\pi}(2u\mu+3v\mu^2)}$$

imaginary replicas?

Summary

Much interesting
equilibrium and nonequilibrium stat mech
at the interface with cellular biology!

- nonequilibrium stat mech of large cellular reaction equation systems
- nonequilibrium stat mech of large gene regulation networks
- stat mech of tailored random graph ensembles
- stochastic evolution of graphs
- ....

Summary

Much interesting
equilibrium and nonequilibrium stat mech
at the interface with cellular biology!

- nonequilibrium stat mech of large cellular reaction equation systems
- nonequilibrium stat mech of large gene regulation networks
- stat mech of tailored random graph ensembles
- stochastic evolution of graphs
- ....

Summary

Much interesting
equilibrium and nonequilibrium stat mech
at the interface with cellular biology!

- nonequilibrium stat mech of large cellular reaction equation systems
- nonequilibrium stat mech of large gene regulation networks
- stat mech of tailored random graph ensembles
- stochastic evolution of graphs
- ....

Thanks to

theory:

Alessia Annibale
Ginestra Bianconi
Andrea De Martino
Kazushi Mimura
Conrad Pérez-Vicente
Sabrina Rabello

bio-informatics:

Luis Fernandes
Franca Fraternali
Jens Kleinjung

papers:

CJ Pérez-Vicente and ACCC, J.Phys.A41 2008
G Bianconi, ACCC and CJ Pérez-Vicente, Phys.Rev.E78 2008
K Mimura and ACCC, J.Phys.A42 2009
ACCC and S Rabello, J.Phys.Conf.Series 197 2009
ACCC, A De Martino and A Annibale, J.Stat.Phys.136 2009
A Annibale, ACCC, LP Fernandes, F Fraternali and J Kleinjung, J.Phys.A42 2009
ACCC, F Fraternali, A Annibale, LP Fernandes and J Kleinjung, preprint 2010

Thanks to

theory:

- Alessia Annibale
- Ginestra Bianconi
- Andrea De Martino
- Kazushi Mimura
- Conrad Pérez-Vicente
- Sabrina Rabello

bio-informatics:

- Luis Fernandes
- Franca Fraternali
- Jens Kleinjung

papers:

CJ Pérez-Vicente and ACCC, J.Phys.A41 2008
G Bianconi, ACCC and CJ Pérez-Vicente, Phys.Rev.E78 2008
K Mimura and ACCC, J.Phys.A42 2009
ACCC and S Rabello, J.Phys.Conf.Series 197 2009
ACCC, A De Martino and A Annibale, J.Stat.Phys.136 2009
A Annibale, ACCC, LP Fernandes, F Fraternali and J Kleinjung, J.Phys.A42 2009
ACCC, F Fraternali, A Annibale, LP Fernandes and J Kleinjung, preprint 2010