Community detection and Role extraction in Networks

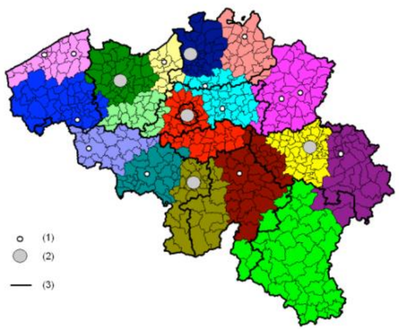
A. Browet

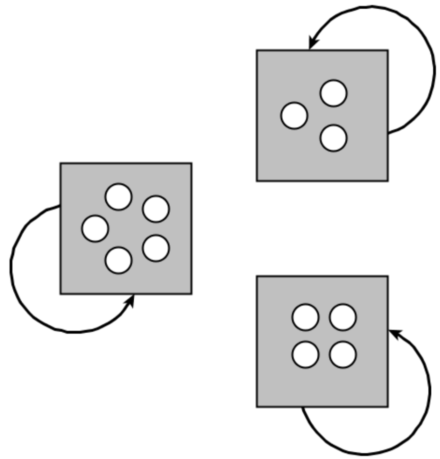
Université catholique de Louvain
EPL - ICTEAM

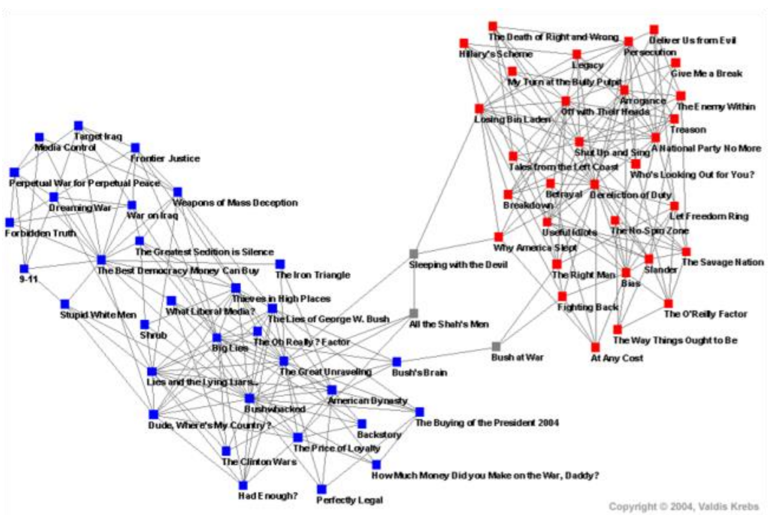
Large Graphs and Networks

Image and Signal Processing Seminars
October 2014

Networks Topology
Community Structures

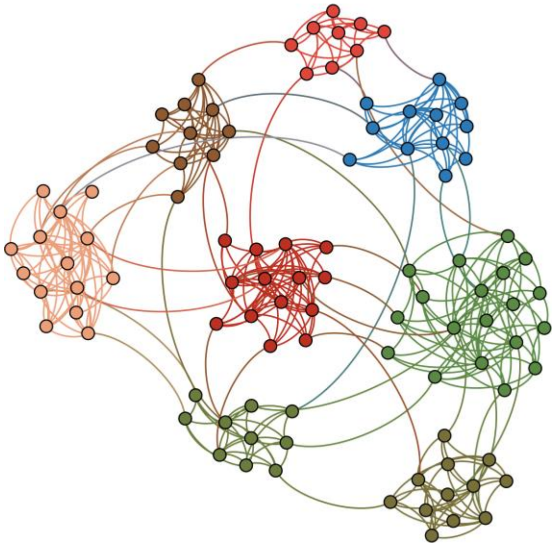






Networks Topology

Community Structures

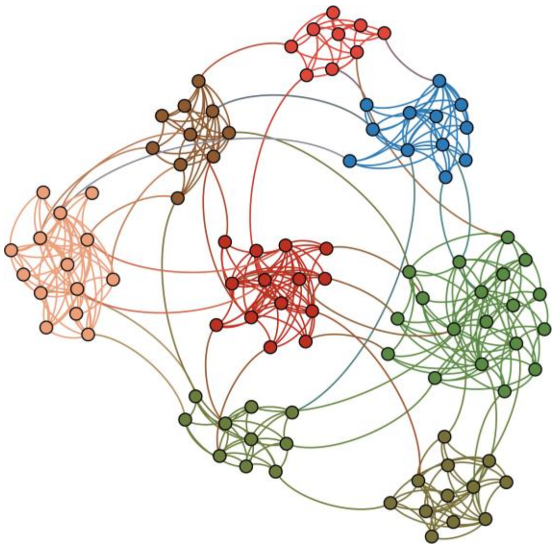


$$H(\sigma) = -H_0 - \sum_{i,j \in V} \left[\alpha_{ij} A(i,j) - \beta_{ij}\right] \delta\left(\sigma_i, \sigma_j\right)$$

Reichardt & Bornholdt (2006)

<table>
<tr><th></th><th>Unweighted network</th><th>Weighted network</th></tr>
<tr><td rowspan="2">Reichardt & Bornholdt</td><td>$\alpha_{ij} = 1$</td><td>$\alpha_{ij} = w(i,j)$</td></tr>
<tr><td colspan="2">$\beta_{ij} = \gamma_{RB} p_{ij} \qquad p_{ij} = \frac{mn_c^2}{n^2}$</td></tr>
<tr><td>Newman & Girvan (modularity)</td><td>$\alpha_{ij} = 1$
$\beta_{ij} = \frac{k_i^{out} k_j^{in}}{m}$</td><td>$\alpha_{ij} = w(i,j)$
$\beta_{ij} = \frac{s_i^{out} s_j^{in}}{m_w}$</td></tr>
<tr><td rowspan="2">Traag et al. (CPM)</td><td>$\alpha_{ij} = 1$</td><td>$\alpha_{ij} = w(i,j)$</td></tr>
<tr><td colspan="2">$\beta_{ij} = \gamma_{CPM}$</td></tr>
<tr><td rowspan="2">Ronhovde & Nussinov</td><td>$\alpha_{ij} = 1 + \gamma_{RN}$</td><td>$\alpha_{ij} = w(i,j) + \gamma_{RN}$</td></tr>
<tr><td colspan="2">$\beta_{ij} = \gamma_{RN}$</td></tr>
<tr><td>Raghavan et al. (label propagation)</td><td colspan="2">$\alpha_{ij} = w(i,j) \qquad \beta_{ij} = 0$</td></tr>
</table>

Networks Topology
Community Structures



$$H_M(\sigma) = q_{out}H(q_{out}) + \sum_{k=1}^{m} q_{k,in}H(q_{k,in}).$$

Rosvall & Bergstrom (2008)

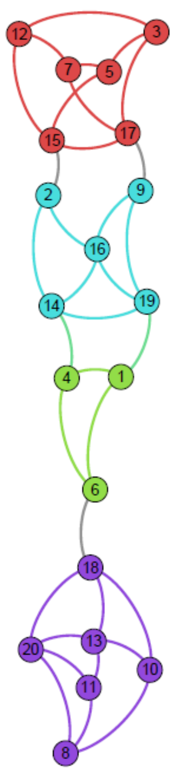
$$H_S(\sigma) = -\log \sum_{j=f}^{\min(F,m)} \frac{\begin{pmatrix} F \\ j \end{pmatrix}\begin{pmatrix} M-F \\ m-j \end{pmatrix}}{\begin{pmatrix} M \\ m \end{pmatrix}}$$

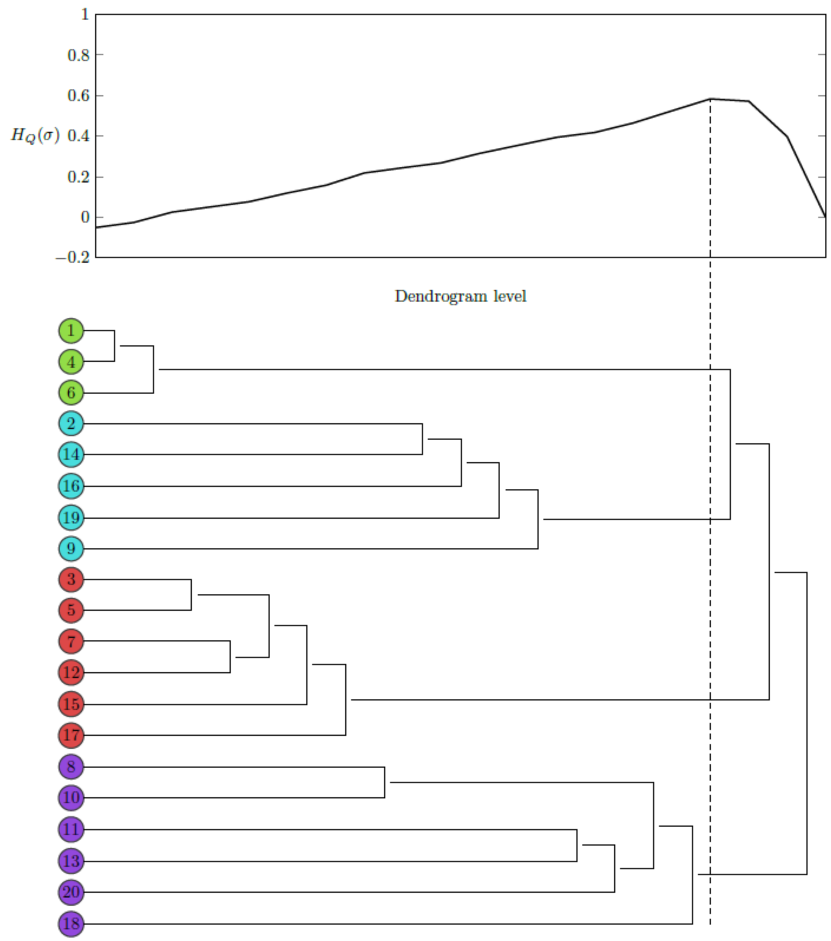
Aldecoa & Marin (2011)

Community Detection Algorithm

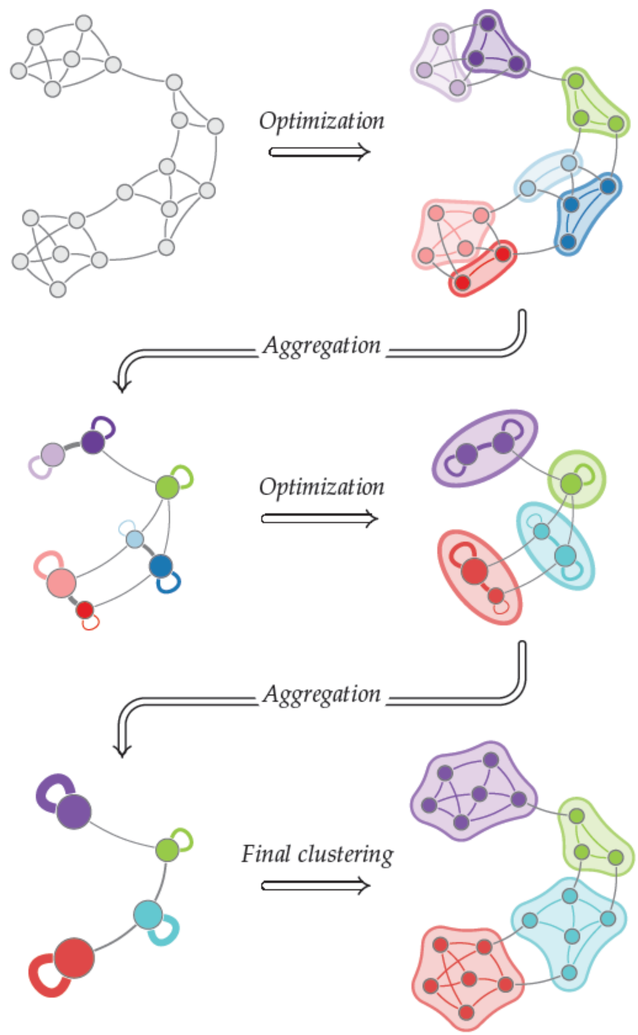
- Simulated annealing (SA)
  - *Kirkpatrick, Gelatt & Vecchi (1983)*
- Fast modularity
  - *Clauset, Newman & Moore (2004)*
- Fast modularity (+ TCER)
  - *Schuetz & Caflisch (2008)*
- Label propagation (LP)
  - *Raghavan, Albert & Kumara (2007)*
- Louvain Method (LM)
  - *Blondel, Guillaume, Lambiotte, et al. (2008)*
- Infomap
  - *Rosvall & Bergstrom (2010)*

Community Detection Algorithm





Community Detection Algorithm



Fast community extraction

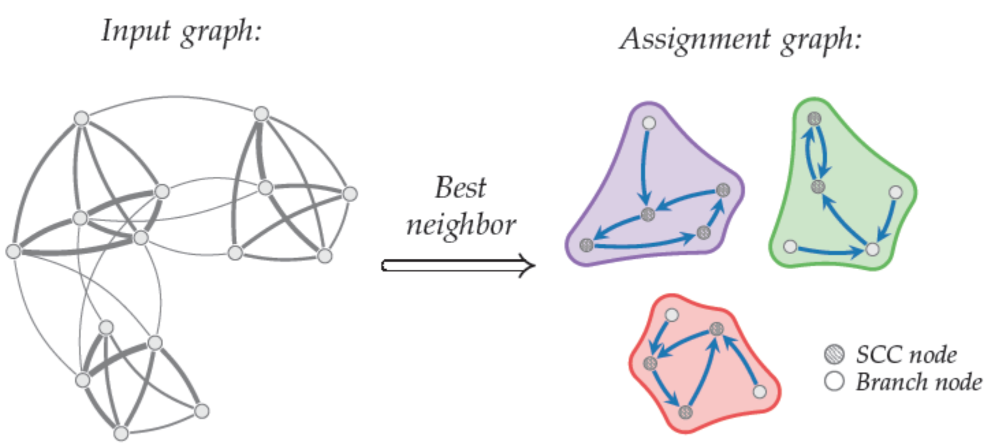
Input : a graph $G(V, E)$
Output : a community partition matrix $C \in \mathbb{R}^{k \times n}$

Initialize $C = I_n,\ C_t = 0,\ G_t = G$

while $C_t \neq I$ **do**
  $C_t \leftarrow \textsc{Assign}(G_t)$
  $C_t \leftarrow \textsc{Positive}(C_t, G_t)$
  while $\exists\, i \in V_t, c \in C_t$ with $\Delta H\,(c_i \to i \to c) > 0$ **do**
    $C_t \leftarrow \textsc{Maximal}(C_t, G_t)$
    $C_t \leftarrow \textsc{Positive}(C_t, G_t)$
  $G_t \leftarrow \textsc{Aggregate}(G_t, C_t)$
  $C = C_t C$

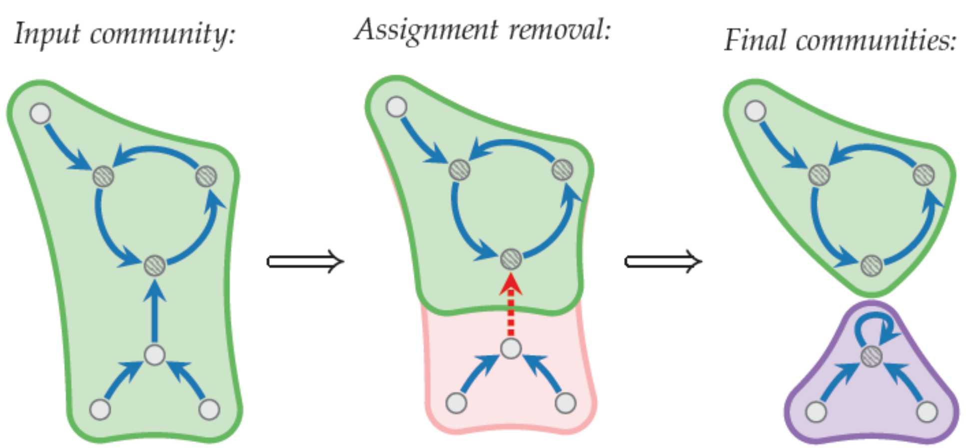
Fast community extraction

function $\textsc{Assign}(G(V,E))$
    for all $i \in V$ **do**
        $a(i) = \arg\max_j \Delta H(i \to j)$
    end for
    $T \leftarrow \text{graph}(V, \{(i, a(i))\ \forall i\})$
    $C_t \leftarrow WCC\ (T)$
    return C_t



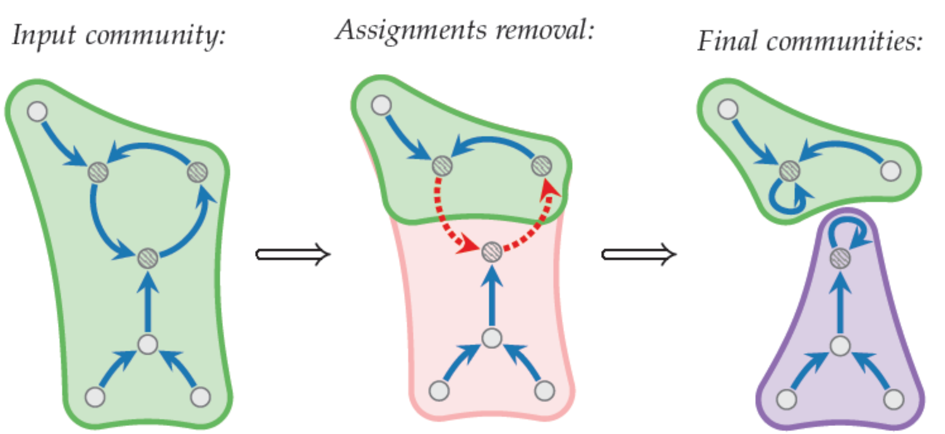
Fast community extraction

function $\text{POSITIVE}(C_t, G(V, E))$
  for all $i \in V$ **do**
    $g(i) = -\Delta H\left(c_i \to i \to \{\}\right)$
  while $\exists i \in c_i$ **with** $g(i) < 0$ **do**
    $c_1, c_2 \leftarrow \text{SPLIT}(c_i)$
    for all $j \in c_1 \cup c_2$ **do**
      $g(j) = -\Delta H\left(c_j \to j \to \{\}\right).$
    $C_t = C_t \setminus \{c_i\} \cup \{c_1, c_2\}$
  return C_t

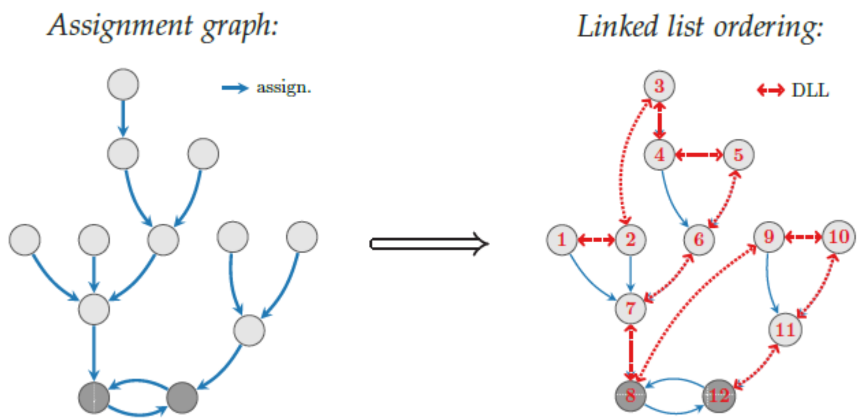


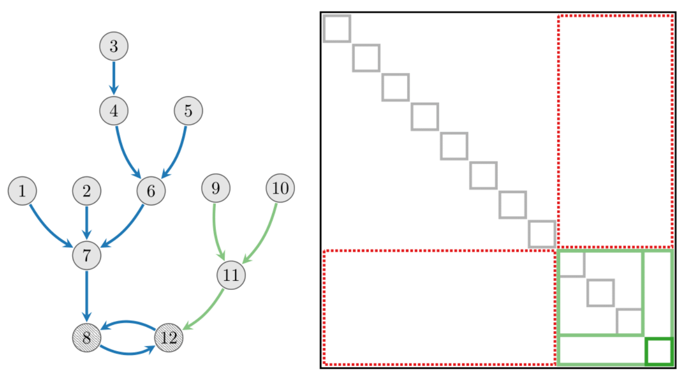
Fast community extraction

function $\text{POSITIVE}(C_t, G(V,E))$
  for all $i \in V$ **do**
    $g(i) = -\Delta H\,(c_i \to i \to \{\})$
  while $\exists i \in c_i$ with $g(i) < 0$ **do**
    $c_1, c_2 \leftarrow \text{SPLIT}(c_i)$
    for all $j \in c_1 \cup c_2$ **do**
      $g(j) = -\Delta H\,(c_j \to j \to \{\})\,.$
    $C_t = C_t \setminus \{c_i\} \cup \{c_1, c_2\}$
  return C_t



Fast community extraction





Fast community extraction

function MAXIMAL($C_t, G(V,E)$)

  $C = C_t$

  for all $i \in V$ **do**

    $c_i^* = \arg\max_c \Delta H(c_i \to i \to c)$

  for all $i \in V$, if $c_i^* \neq c_i$ **do**

    draw $p(i)$ uniform $\in [0,1]$

    if $p(i) < p$ **then**

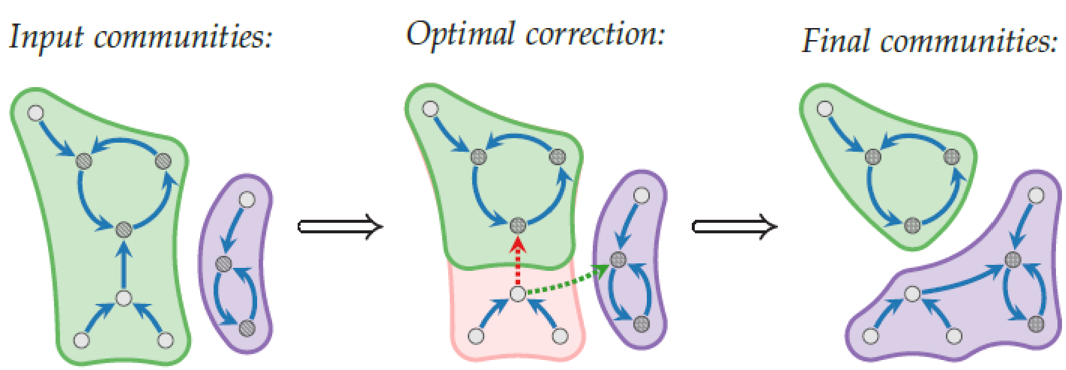
      $b(i) = branch(i)$

      if $\Delta H(c_i \to b(i) \to c_i^*) > 0$ **then**

        $a(i) = \arg\max_{j \in c_i^*} \Delta H(i \to j)$

        $C \leftarrow insert\,(b(i), c_i^*)\,.$

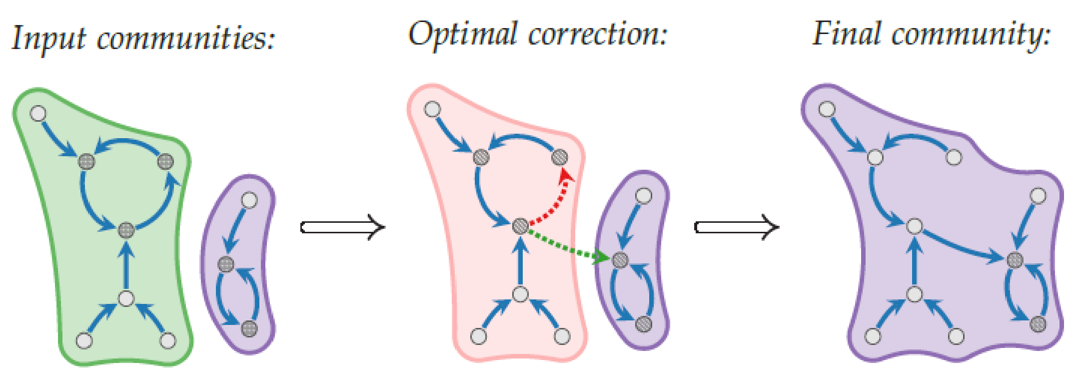
  return C



Fast community extraction

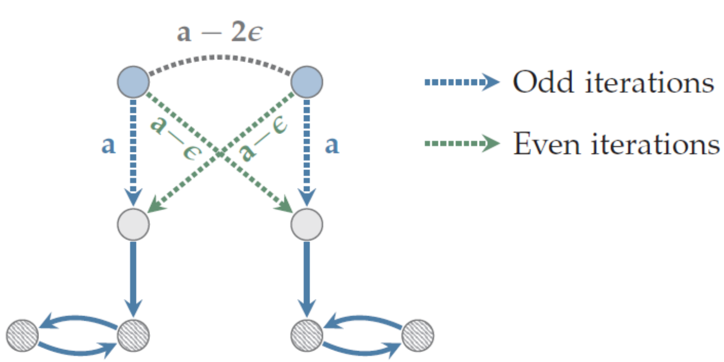
---

function MAXIMAL($C_t, G(V, E)$)
  $C = C_t$
  for all $i \in V$ **do**
    $c_i^* = \arg\max_c \Delta H\left(c_i \to i \to c\right)$
  for all $i \in V$, if $c_i^* \neq c_i$ **do**
    draw $p(i)$ uniform $\in [0, 1]$
    if $p(i) < p$ **then**
      $b(i) = branch(i)$
      if $\Delta H\left(c_i \to b(i) \to c_i^*\right) > 0$ **then**
        $a(i) = \arg\max_{j \in c_i^*} \Delta H(i \to j)$
        $C \leftarrow insert\left(b(i), c_i^*\right)$.
  return C



Fast community extraction

function $\textsc{Maximal}(C_t, G(V,E))$
  $C = C_t$
  for all $i \in V$ **do**
    $c_i^* = \arg\max_c \Delta H(c_i \to i \to c)$
  for all $i \in V$, if $c_i^* \neq c_i$ **do**
    draw $p(i)$ uniform $\in [0,1]$
    if $p(i) < p$ **then**
      $b(i) = branch(i)$
      if $\Delta H(c_i \to b(i) \to c_i^*) > 0$ **then**
        $a(i) = \arg\max_{j \in c_i^*} \Delta H(i \to j)$
        $C \leftarrow insert\,(b(i), c_i^*)$.
  return C



Performance & Accuracy

LFR benchmark model

Lancichinetti, Fortunato & Radicchi (2008)

$$k_i \backsim k^{-\tau_1} \qquad n_c \backsim n^{-\tau_2}$$

$$\langle k_{int} \rangle = (1 - \mu_T) \langle k \rangle ,$$
$$\langle k_{ext} \rangle = \mu_T \langle k \rangle .$$

$$\left\langle w^{int} \right\rangle = \frac{(1 - \mu_W) \langle s \rangle}{(1 - \mu_T) \langle k \rangle} = \frac{(1 - \mu_W)}{(1 - \mu_T)} \langle k \rangle^{\beta - 1} ,$$

$$s_i^{int} = (1 - \mu_W) k_i^{\beta},$$
$$s_i^{ext} = \mu_W k_i^{\beta}.$$

$$\left\langle w^{ext} \right\rangle = \frac{\mu_W \langle s \rangle}{\mu_T \langle k \rangle} = \frac{\mu_W}{\mu_T} \langle k \rangle^{\beta - 1} ,$$

Normalized mutual information

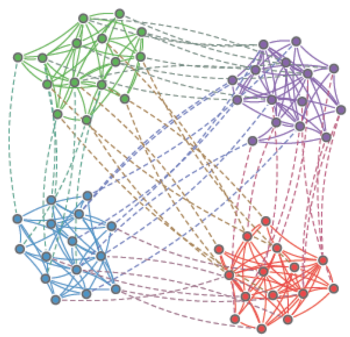
Danon, Diaz-Guilera, Duch, et al. (2005)

$$NMI(X,Y) = \frac{2\, I(X,Y)}{H(X) + H(Y)}.$$

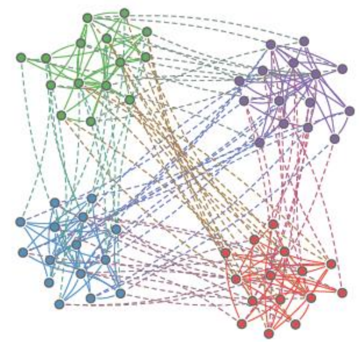
Performance & Accuracy

LFR benchmark model

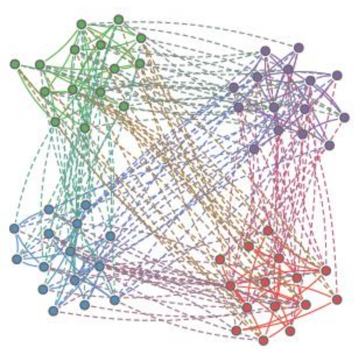
Lancichinetti, Fortunato & Radicchi (2008)



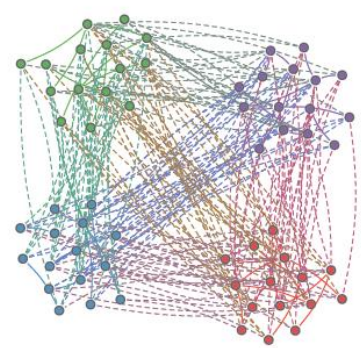
(a) $\mu_T = 0.2$



(b) $\mu_T = 0.4$

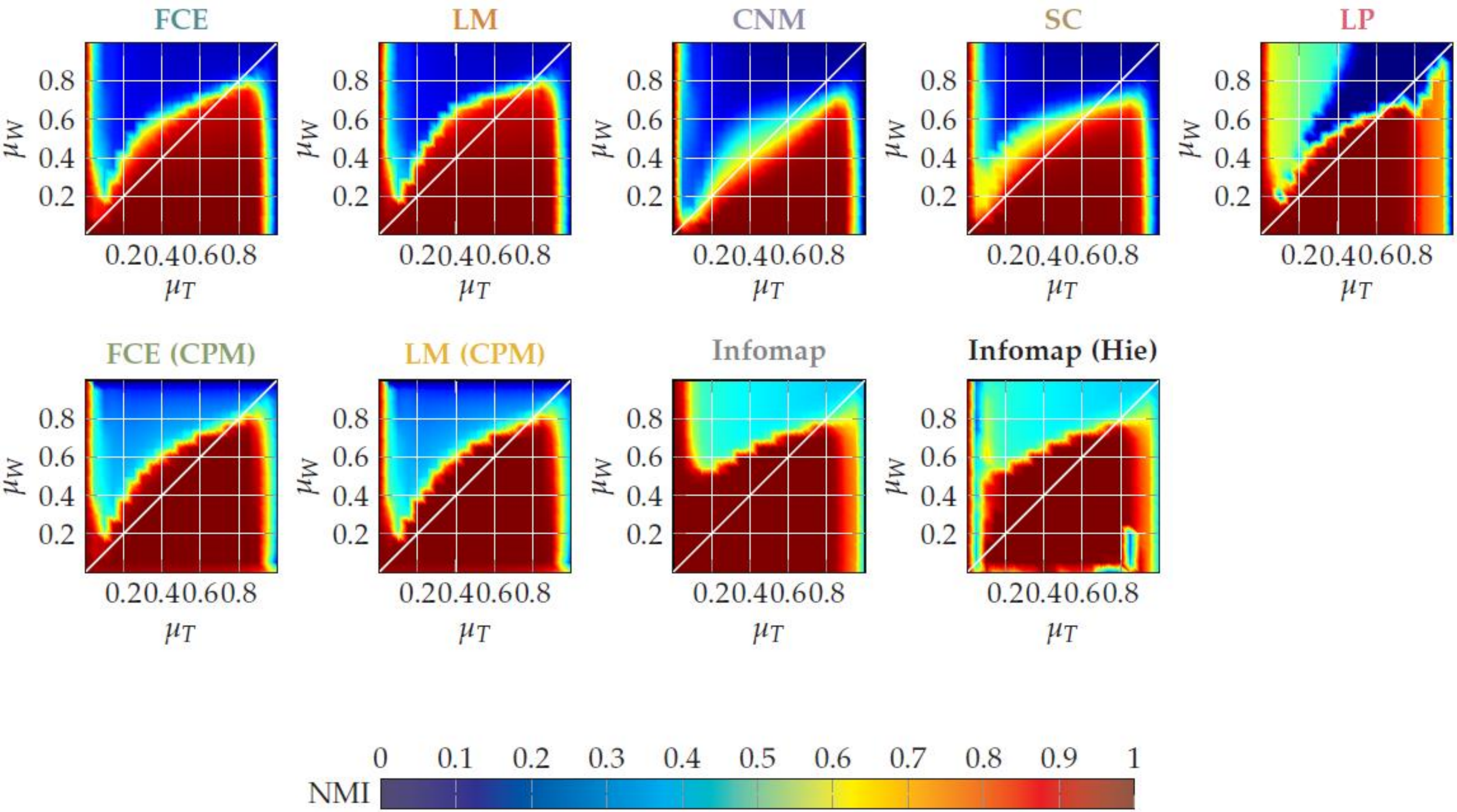


(c) $\mu_T = 0.6$

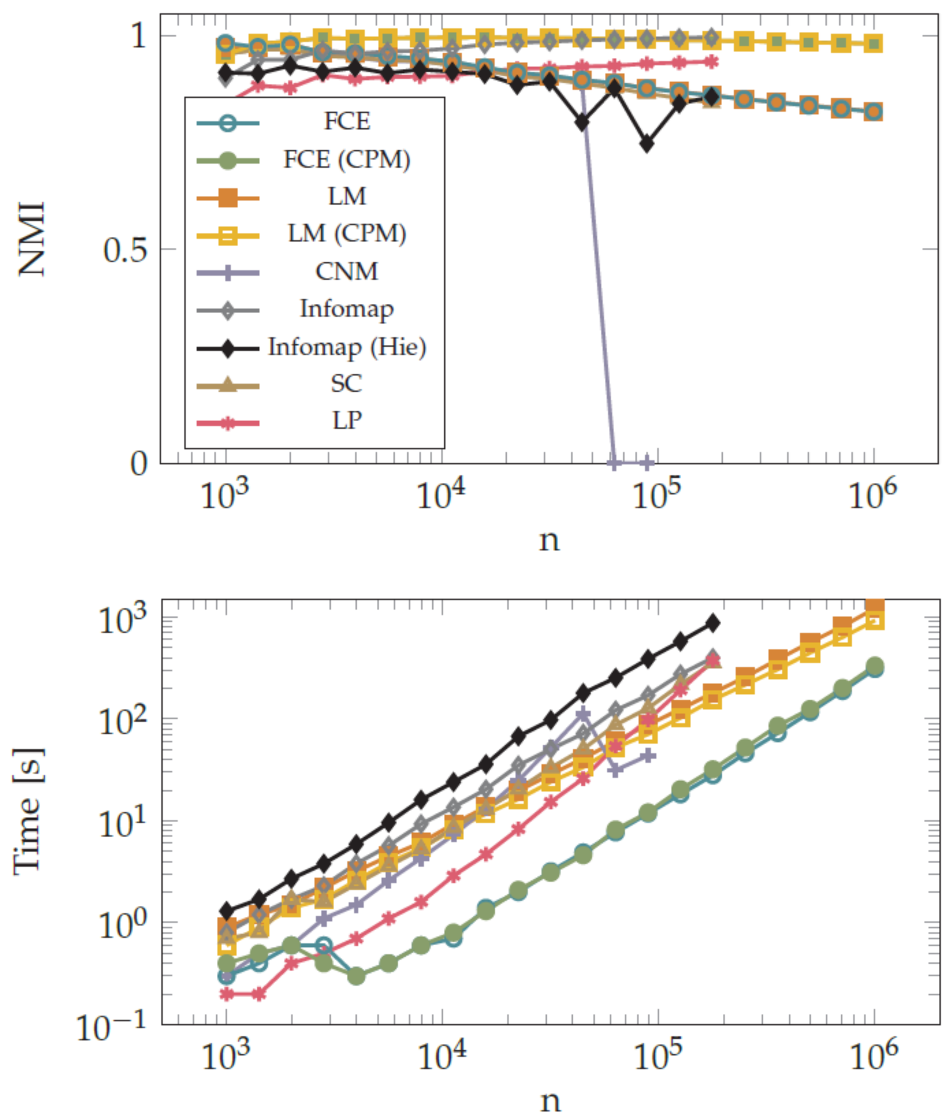


(d) $\mu_T = 0.8$

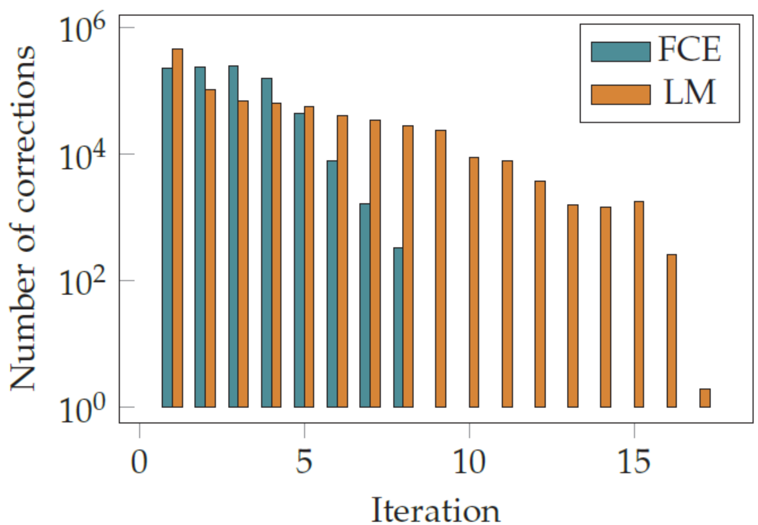
Performance & Accuracy

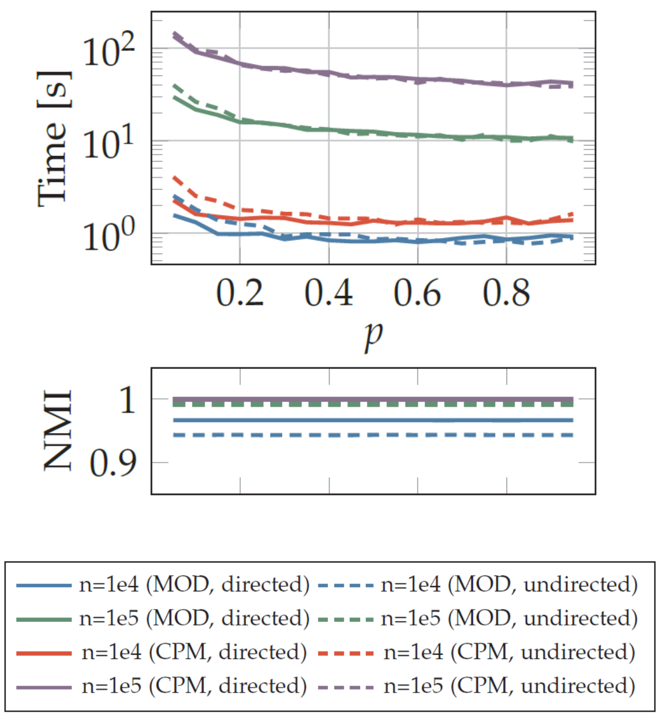


Performance & Accuracy



Performance & Accuracy



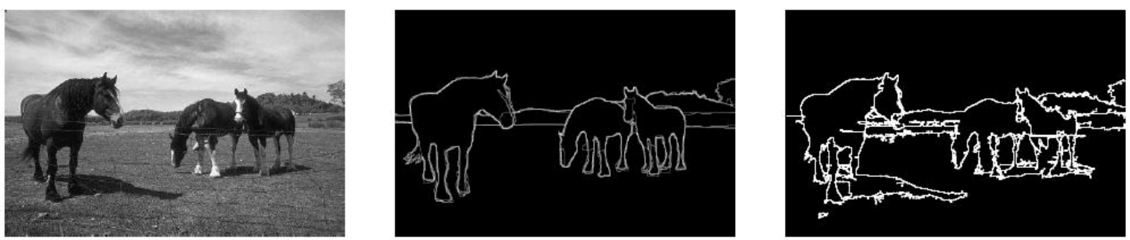


Application to image processing

$$w(i,j) = \begin{cases} e^{\frac{d(i,j)^2}{\sigma_x^2}} e^{\frac{|F(i)-F(j)|^2}{\sigma_i^2}} & \text{if } d(i,j) < d_{max}, \\ 0 & \text{otherwise} \end{cases}$$

$$Q_\Lambda(\sigma) = \frac{1}{m} \sum_{i,j \in V} \left[W - \frac{S \Lambda S}{m} \right]_{(i,j)} \delta(\sigma_i, \sigma_j)$$

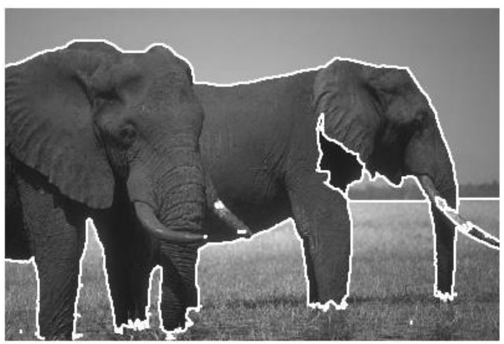
Input picture Human benchmark FCE segmentation



Application to image processing

$$w(i,j) = \begin{cases} e^{\frac{d(i,j)^2}{\sigma_x^2}} e^{\frac{|F(i)-F(j)|^2}{\sigma_i^2}} & \text{if } d(i,j) < d_{max}, \\ 0 & \text{otherwise} \end{cases}$$

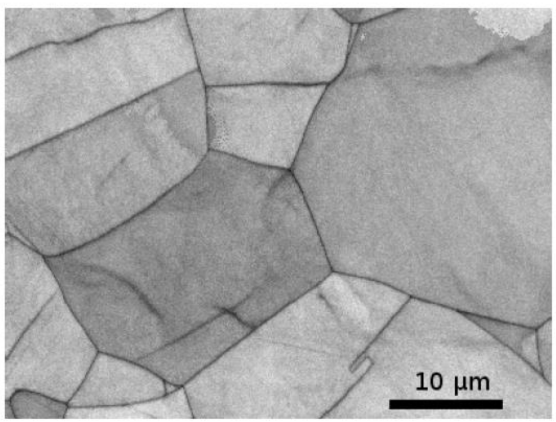
$$Q_\Lambda(\sigma) = \frac{1}{m} \sum_{i,j \in V} \left[W - \frac{S \Lambda S}{m} \right]_{(i,j)} \delta(\sigma_i, \sigma_j)$$

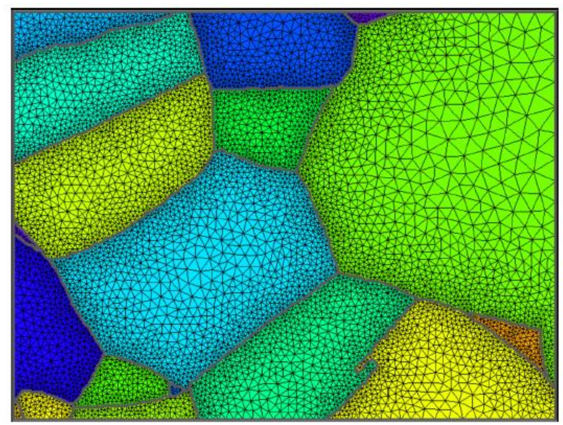


Application to image processing

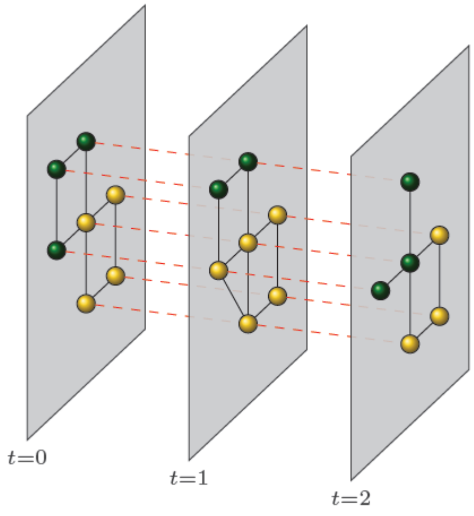
$$w(i,j) = \begin{cases} e^{\frac{d(i,j)^2}{\sigma_x^2}} e^{\frac{|F(i)-F(j)|^2}{\sigma_i^2}} & \text{if } d(i,j) < d_{max}, \\ 0 & \text{otherwise} \end{cases}$$

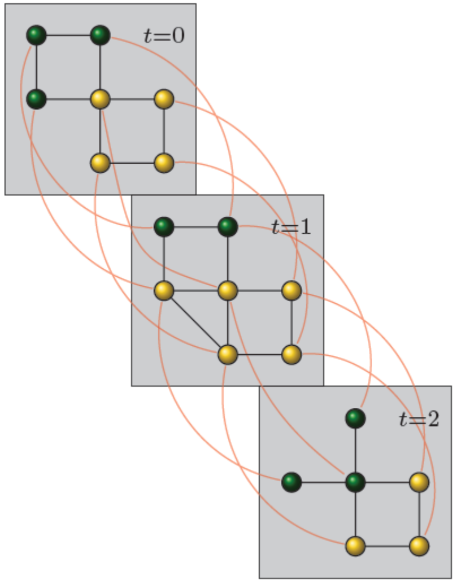
$$Q_\Lambda(\sigma) = \frac{1}{m} \sum_{i,j \in V} \left[W - \frac{S \Lambda S}{m} \right]_{(i,j)} \delta(\sigma_i, \sigma_j)$$



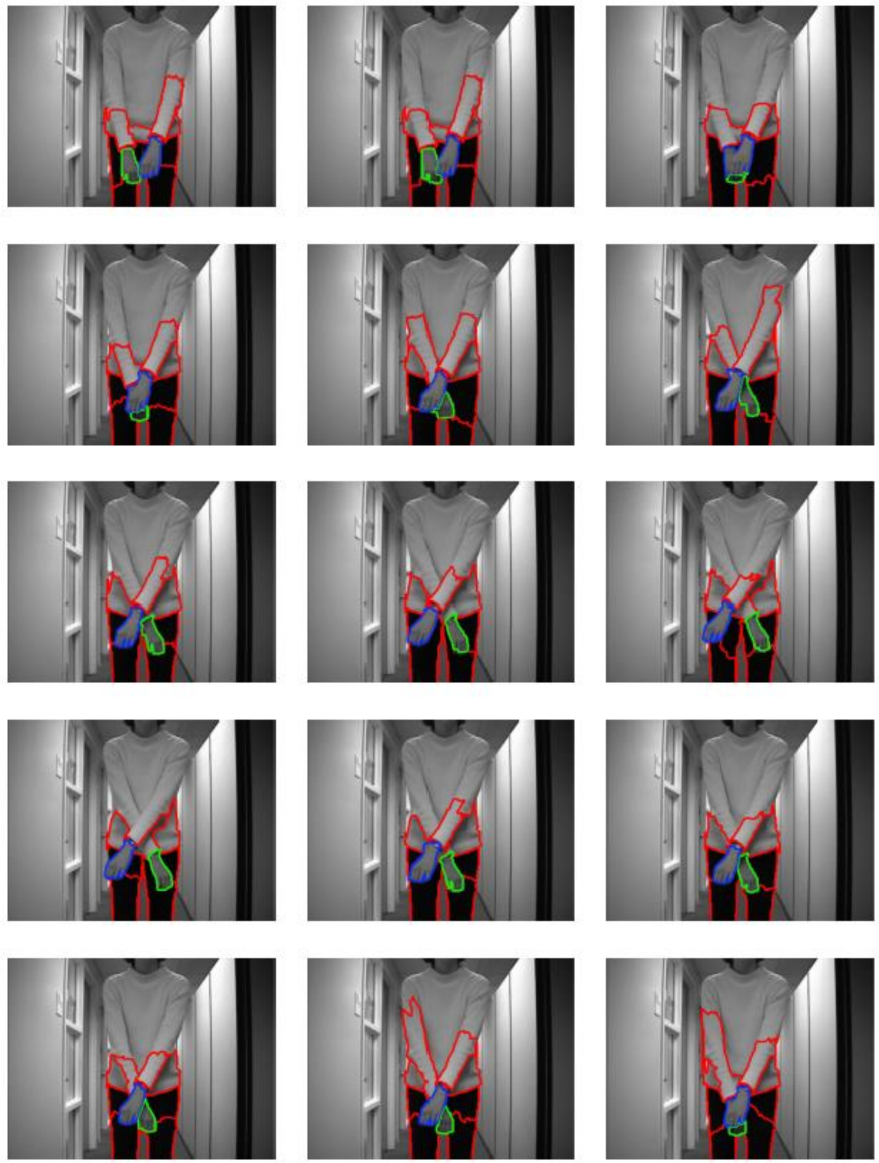


Application to video tracking



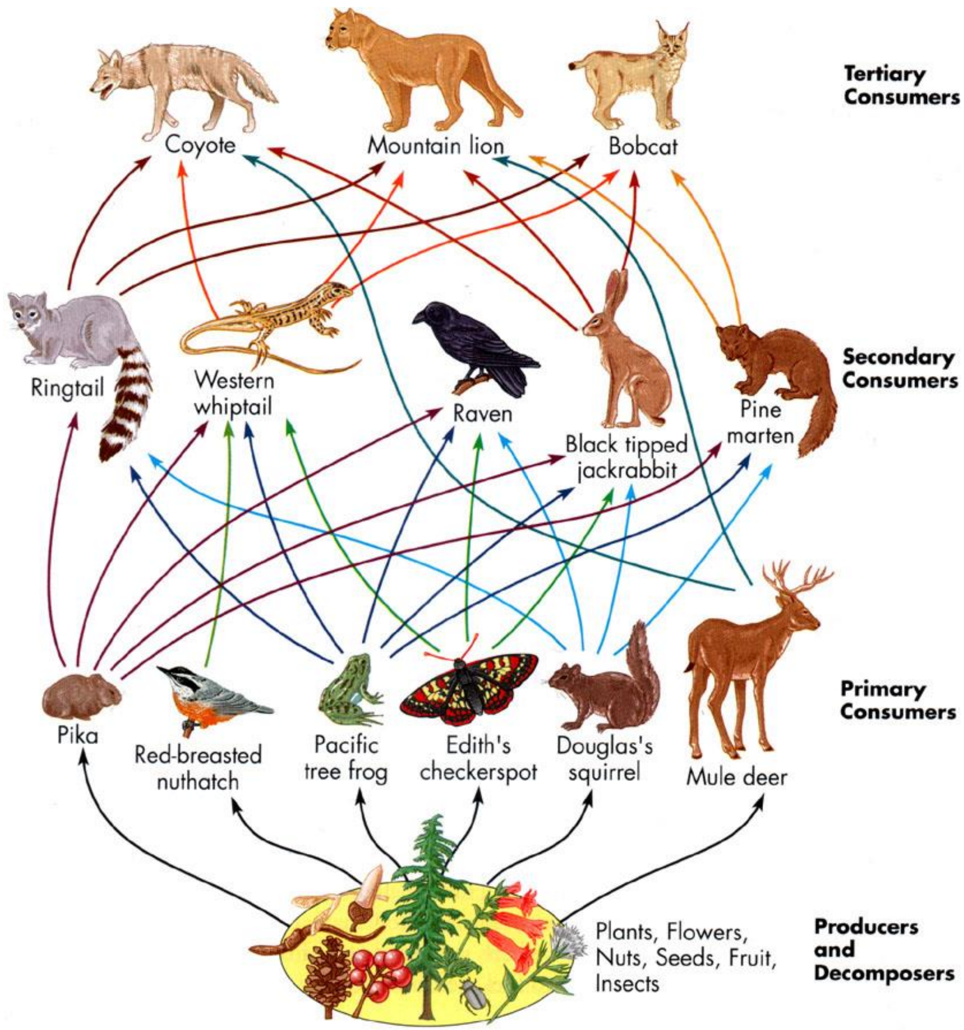


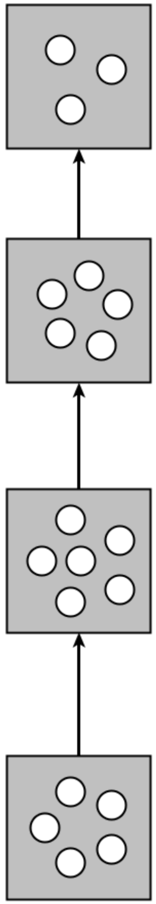
Application to video tracking



Networks Topology

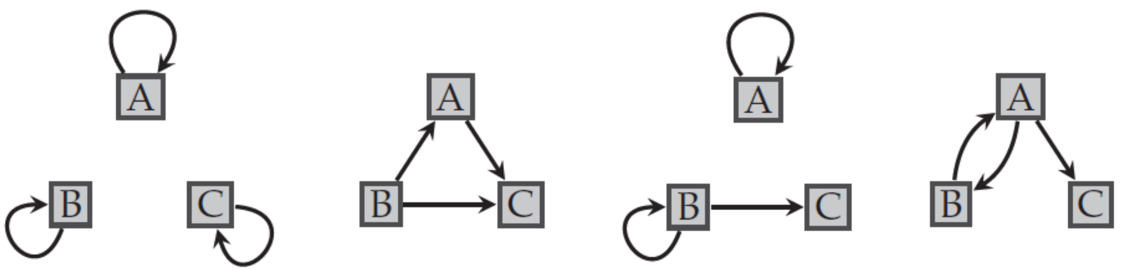
Role Structure

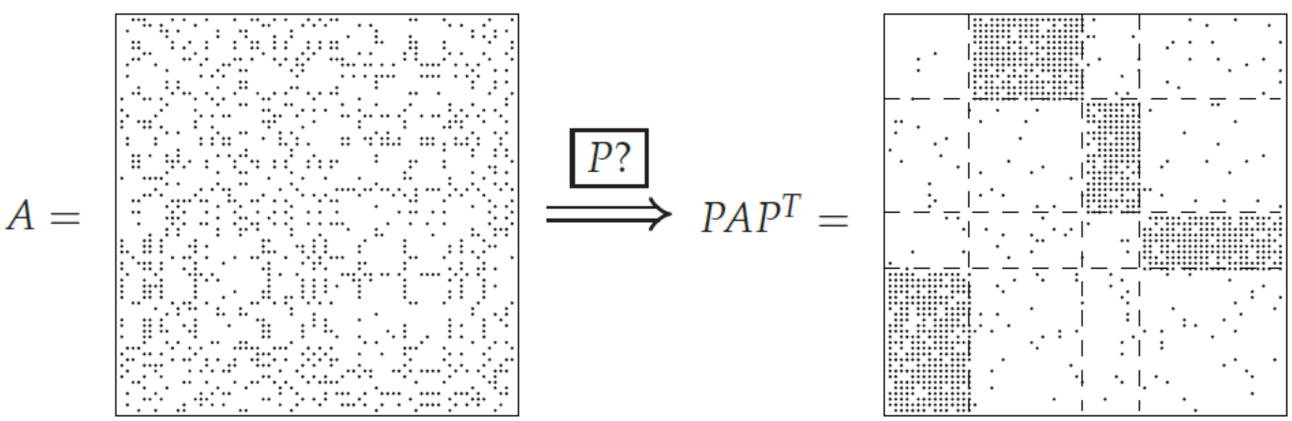




Networks Topology

Role Structure





Role modeling
Pairwise node similarity

Bondel, Gajardo, Heymans, et al. (2004)

$$S_{k+1} = \frac{A\, S_k\, A^T + A^T\, S_k\, A}{\|A\, S_k\, A^T + A^T\, S_k\, A\|_F}.$$

Cooper & Barahona (2011)

$$X = \left[\beta A \mathbf{1} \mid \ldots \mid (\beta A)^{l_{max}} \mathbf{1} \mid \beta A^T \mathbf{1} \mid \ldots \mid \left(\beta A^T\right)^{l_{max}} \mathbf{1}\right]$$

$$S_A^{CB}(i,j) = \frac{x_i x_j^T}{\|x_i\|\,\|x_j\|}.$$

Leicht, Holme & Newman (2006)

$$S_A^L(i,j) = \delta(i,j) + \frac{m\lambda}{k_i^{out} k_j^{in}} \sum_{l=1}^{\infty} \left(\frac{\alpha}{\lambda}\right)^l \left[A^l\right](i,j)$$

Role modeling

Pairwise node similarity

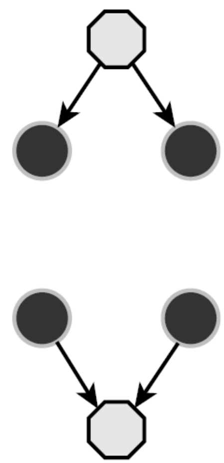
✓ Groups of nodes sharing similar neighborhood patterns in the graph

coming/outgoing edges

$\boxed{l = 1}$

pattern: I

$A^T A$



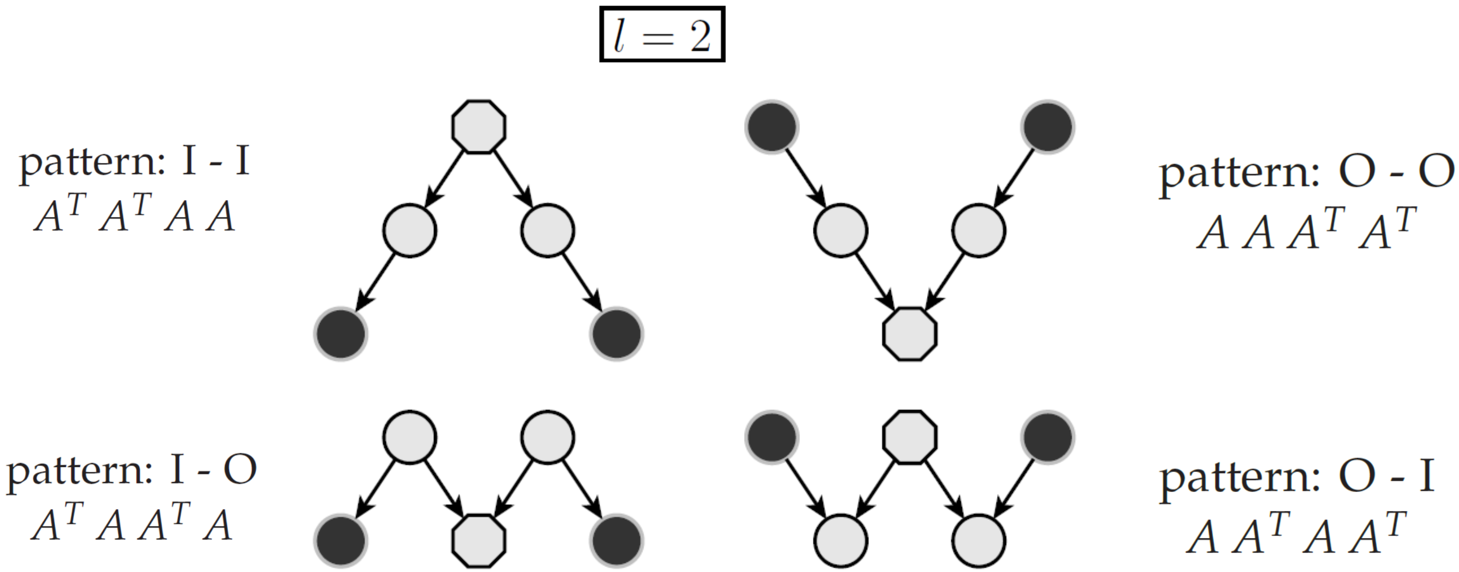
pattern: O

$A A^T$

$$T_1 = AA^T + A^T A$$

Role or Block modeling

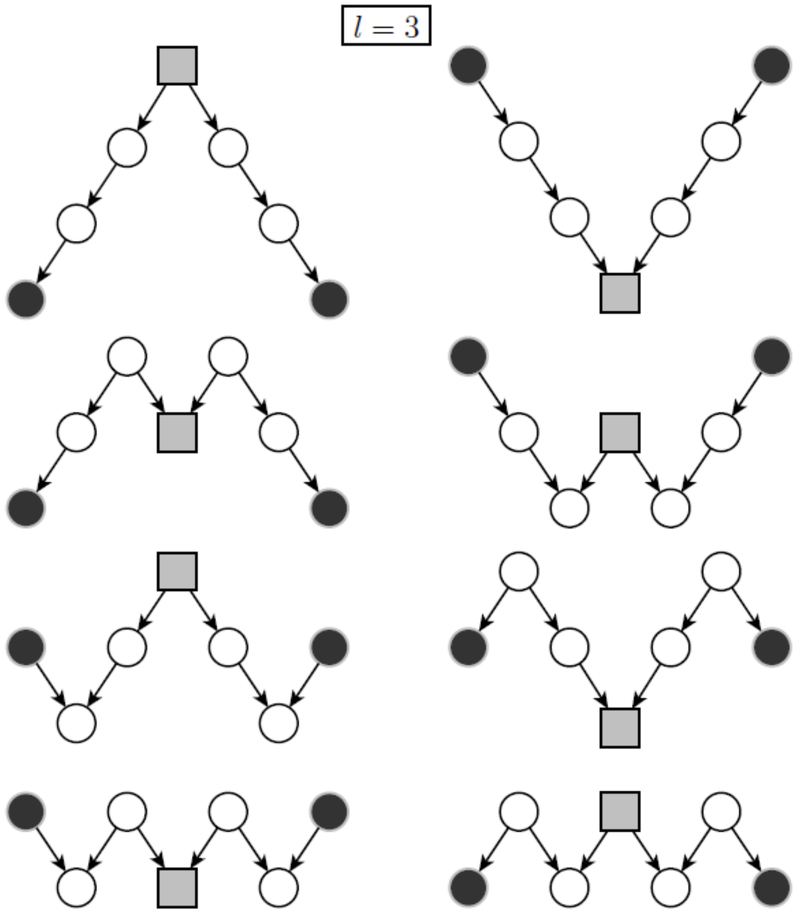
- ✓ Groups of nodes sharing similar neighborhood patterns in the graph



$$T_2 = AAA^TA^T + AA^TAA^T + A^TAA^TA + A^TA^TAA.$$

Role or Block modeling

✓ Groups of nodes sharing similar neighborhood patterns in the graph



Role or Block modeling
Pairwise Similarity Measure

- ✓ Groups of nodes sharing similar neighborhood patterns in the graph

$$S = \sum_{\ell=1}^{\infty} \beta^{2(\ell-1)} T_{\ell}.$$

$$\Gamma_A : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n} : \Gamma_A[X] = AXA^T + A^T XA$$

$$T_1 = \Gamma_A[I],$$
$$T_2 = \Gamma_A[T_1] = \Gamma_A^2[I],$$
$$T_3 = \Gamma_A[T_2] = \Gamma_A^3[I],$$

$$S = \sum_{\ell=1}^{\infty} \beta^{2(\ell-1)} \Gamma_A^{\ell}[I],$$

Role or Block modeling
Pairwise Similarity Measure

- ✓ Groups of nodes sharing similar neighborhood patterns in the graph

$$S = \sum_{\ell=1}^{\infty} \beta^{2(\ell-1)} \Gamma_A^{\ell}[I],$$

$$\Gamma_A : \mathbb{R}^{n \times n} \to \mathbb{R}^{n \times n} : \Gamma_A[X] = AXA^T + A^T XA$$

$$S_{k+1} = \Gamma_A[I] + \cdots + \left(\beta^2\right)^k \Gamma_A^{k+1}[I] + \left(\beta^2\right)^{k+1} \Gamma_A^{k+1}[S_0]$$

$$\Downarrow$$

$$\boxed{S_{k+1} = \Gamma_A\left[I + \beta^2 S_k\right]}$$

Pairwise Similarity Measure

$$S_{k+1} = \Gamma_A[I] + \cdots + \left(\beta^2\right)^k \Gamma_A^{k+1}[I] + \left(\beta^2\right)^{k+1} \Gamma_A^{k+1}[S_0]$$

Converges if $\rho\left(\beta^2 \Gamma_A[.]\right) < 1$

$$\Gamma_A[X] = AXA^T + A^T XA \quad \longrightarrow \quad vec\left(\Gamma_A[X]\right) = \left(A \otimes A + A^T \otimes A^T\right) vec(X)$$

$$\rho\left(\beta^2\left(A \otimes A + A^T \otimes A^T\right)\right) < 1$$

$$\boxed{\beta^2 < \frac{1}{\rho\left(A \otimes A + A^T \otimes A^T\right)}}$$

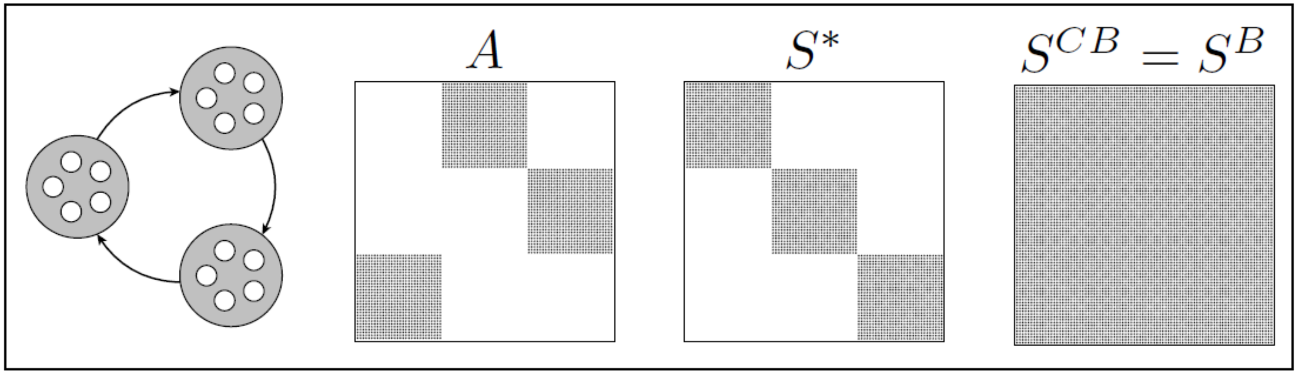
Sufficient condition : $\beta^2 \leq \dfrac{1}{\rho\left(A + A^T\right)^2}$

Pairwise Similarity Measure
Fixed point solution

$$S_0 = 0 \qquad\qquad S_1 = AA^T + A^T A$$

$$S_{k+1} = S_1 + \beta^2 \Gamma_A [S_k]$$

$$vec(S^*) = \left[I - \beta^2 \left(A \otimes A + (A \otimes A)^T\right)\right]^{-1} vec(S_1)$$



- × Exact solution: intractable
- × Power method: expensive $O(n^3)$

Low rank Approximation

$$S_k^{(r)} = X_k X_k^T \;, \qquad X_k \in \mathbb{R}^{n\times r}$$

$$S_{k+1}^{(r)} = \Pi^{(r)} \underbrace{\left[S_1^{(r)} + \beta^2 \Gamma_A \left[S_k^{(r)} \right] \right]}_{\text{Rank } 3r} = X_{k+1}\, X_{k+1}^T$$

$$S_1 = A\,A^T + A^T\,A = \left[A \mid A^T \right] \left[A \mid A^T \right]^T$$

Truncated SVD $\left[A \mid A^T \right] \approx U_1 \Sigma_1 V_1^T$

$$\begin{aligned} S_1^{(r)} &= \Pi^{(r)} \left[\left[A \mid A^T \right] \left[A \mid A^T \right]^T \right] \\ &= U_1 \Sigma_1^2 U_1^T = X_1 X_1^T \qquad\qquad X_1 = U_1 \Sigma_1 \end{aligned}$$

Low rank Approximation
Iterative solutions

$$S_{k+1}^{(r)} = \Pi^{(r)} \left[S_1^{(r)} + \beta^2 \Gamma_A \left[S_k^{(r)} \right] \right] = X_{k+1} \, X_{k+1}^T$$

$$S_1^{(r)} + \beta^2 \Gamma_A \left[S_k^{(r)} \right] = X_1 X_1^T + \beta^2 A X_k X_k^T A^T \; + \beta^2 A^T X_k X_k^T A$$

$$= Y_k \, Y_k^T$$

$$Y_k = \left[X_1 \;\mid \beta A X_k \;\mid \beta A^T X_k \right]$$

Low rank Approximation

Iterative solutions

$$X_{k+1} X_{k+1}^T = \Pi^{(r)} \left[Y_k Y_k^T \right]$$

$$Y_k = \left[X_1 \mid \beta A X_k \mid \beta A^T X_k \right]$$

QR factorization $\quad Y_k = Q_k R_k$

(keep the first r columns of Q_k) $\qquad X_{k+1} = Q_k \mathcal{U}_k \Omega_k$

Truncated SVD $\quad R_k \approx \mathcal{U}_k \Omega_k \mathcal{V}_k$

Existence of Fixed Point solution and Guaranteed
local convergence of the sequence for sufficiently small β !


Stewart, Error and perturbation bounds for subspaces associated with certain eigenvalue problems [1973]

Low rank Projection

Convergence

Δ small symmetric perturbation and $S^{(r)}$ low rank fixed point solution

$$f(S) = S_1^{(r)} + \beta^2 \Gamma_A [S] \qquad S^{(r)} = \Pi^{(r)}\left(f\left(S^{(r)}\right)\right) \qquad S^{(r)} = U\Sigma^2 U^T$$

$$[U\ V]^T\ f(S^{(r)})\ [U\ V] = \begin{bmatrix} \Sigma^2 & \\ & \sigma^2 \end{bmatrix}$$

$$f(S^{(r)} + \Delta) = f(S^{(r)}) + \beta^2 \Gamma_A [\Delta]$$

$$[U\ V]^T\ \left(f(S^{(r)}) + \beta^2 \Gamma[\Delta]\right)\ [U\ V] = \begin{bmatrix} E_{11} & E_{21}^T \\ E_{21} & E_{22} \end{bmatrix}$$

Low rank Projection
Convergence

$$[U\ V]^T\ \left(f(S^{(r)}) + \beta^2\Gamma[\Delta]\right)\ [U\ V] = \left[\begin{array}{cc} E_{11} & E_{21}^T \\ E_{21} & E_{22} \end{array}\right]$$

There exists $Q \in \mathbb{R}^{n\times r}$ such that UQ is an invariant subspace if

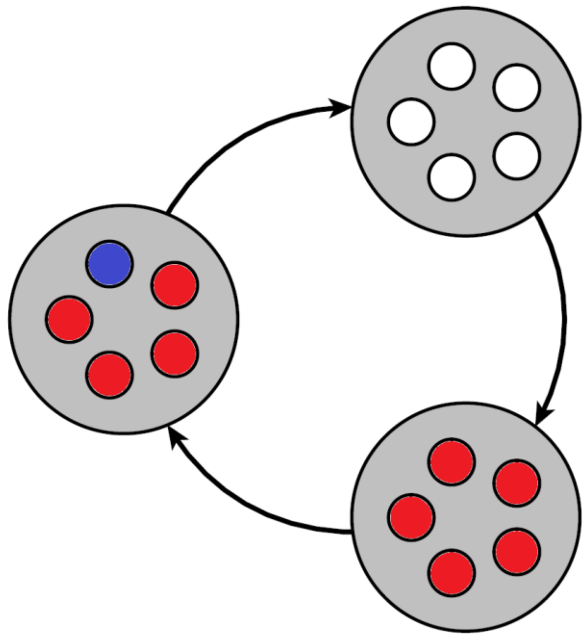
$$0 \leq 4\beta^2 \left\|\Gamma\left[\Delta\right]\right\|_F \leq \Sigma_{r,r}^2 - \sigma_{1,1}^2$$

It implies that $\left\|S^{(r)} - \Pi^{(r)}\left[f(S^{(r)} + \Delta)\right]\right\|_F \leq \gamma \left\|\Delta\right\|_F$

$\gamma < 1$ if $\beta^2 < \dfrac{1}{\left\|A \otimes A + A^T \otimes A^T\right\|_2 \left(\frac{4\|\Sigma^2\|}{\Sigma_{r,r}^2 - \sigma_{1,1}^2} + 1\right)}$

Erdos-Reyni random graphs
with a block stucture

$$G_B(V_B, E_B)$$



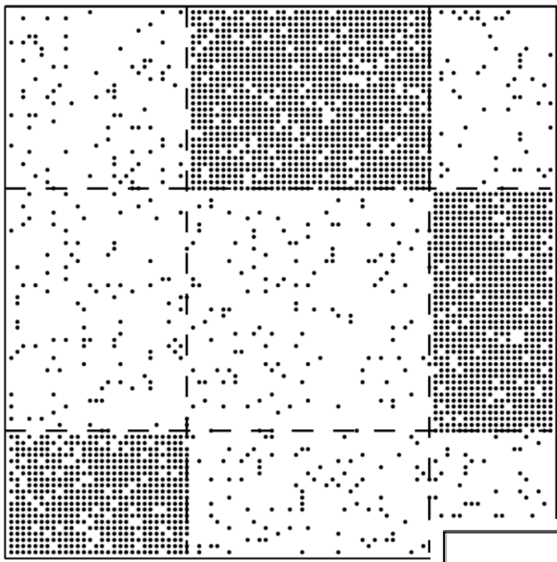
$$i, j \in V_A$$

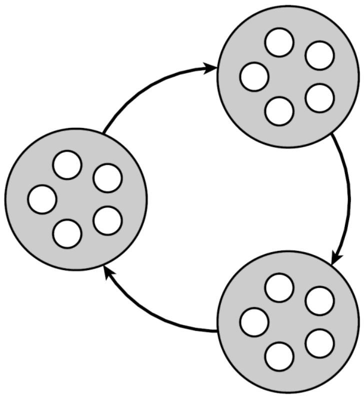
$$(i, j) \in E_A \text{ w.p. } p_{out}$$

$$\text{if } (R(i), R(j)) \notin E_B$$

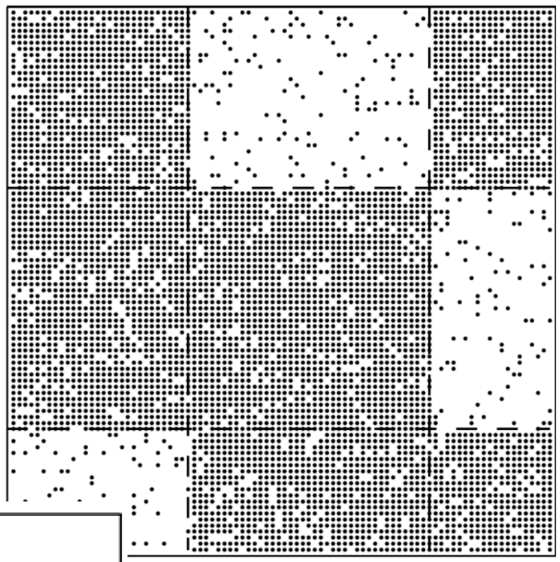
Erdos-Reyni random graphs
with a block stucture

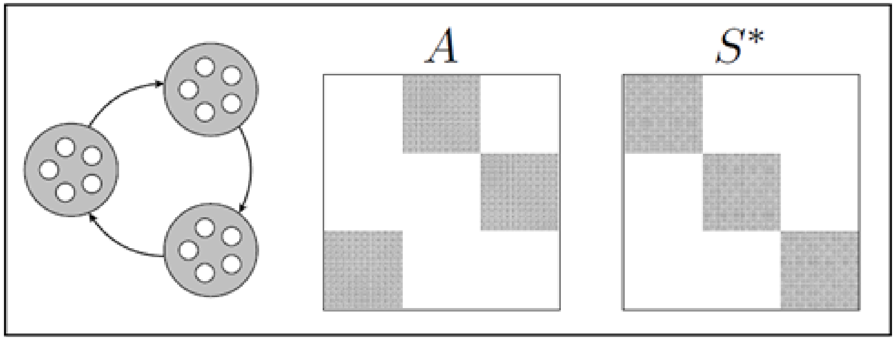
$p_{in} >> p_{out}$





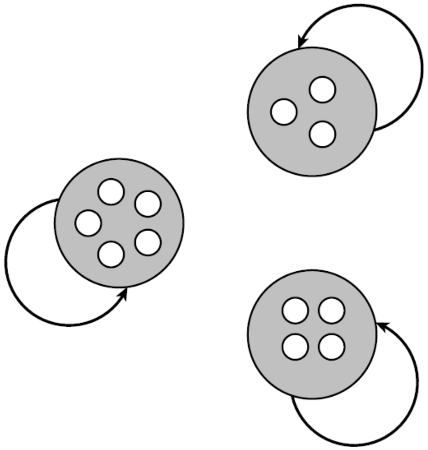
$p_{in} << p_{out}$

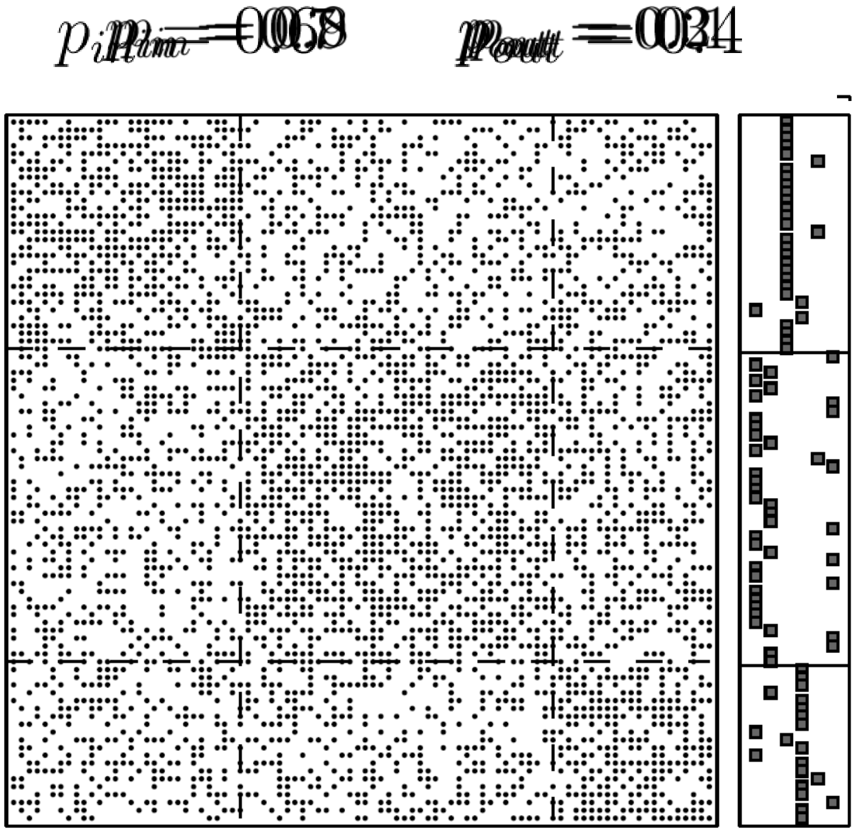




Erdos-Reyni random graphs

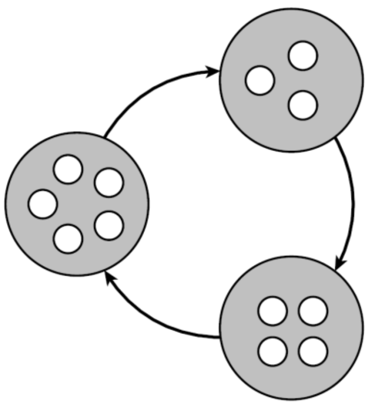
Results

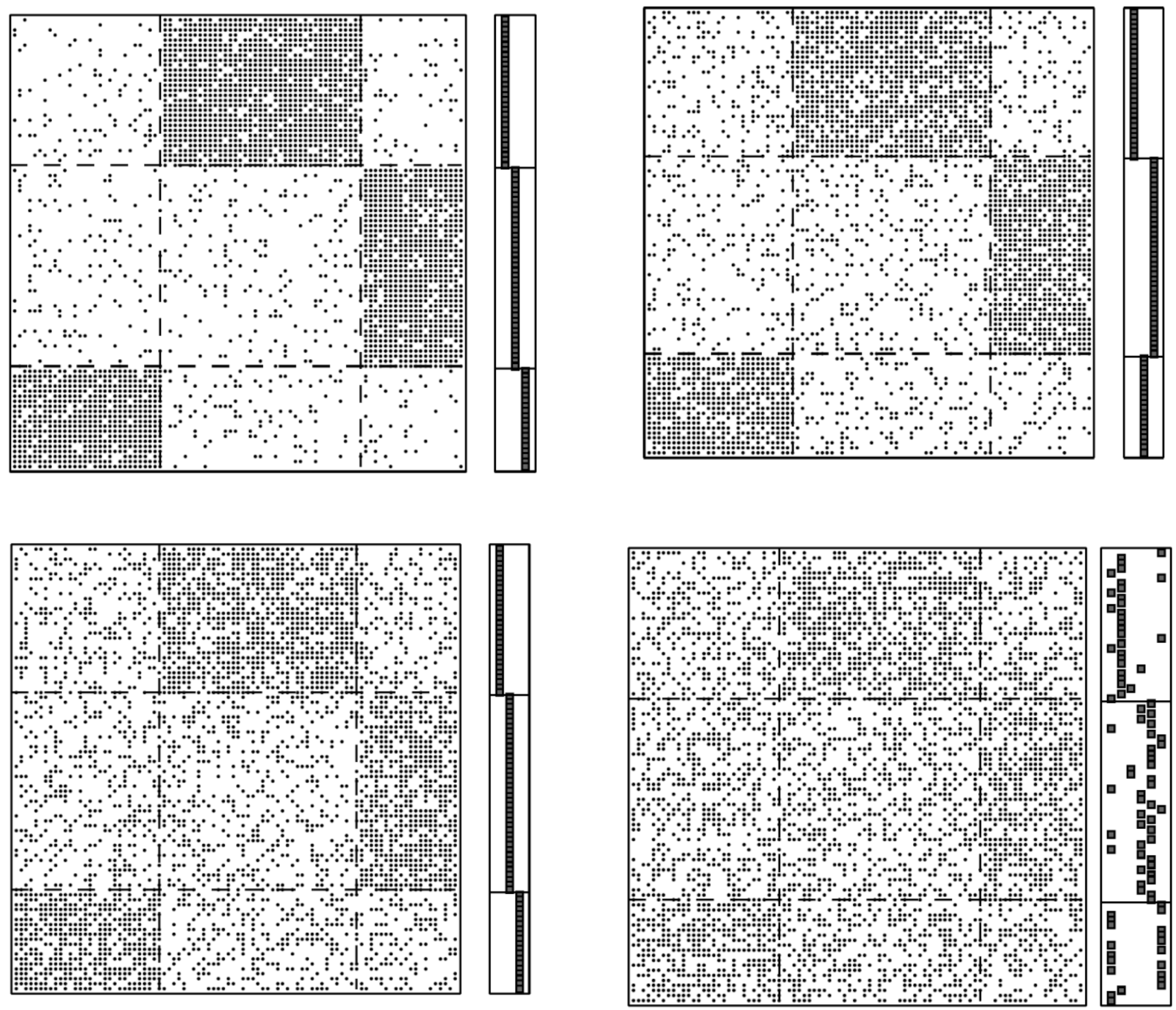




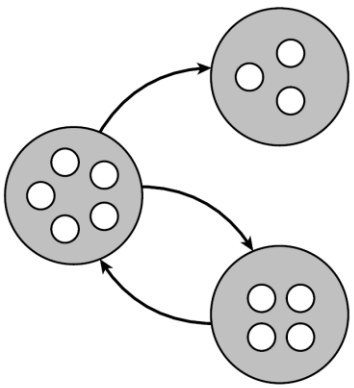
Erdos-Reyni random graphs

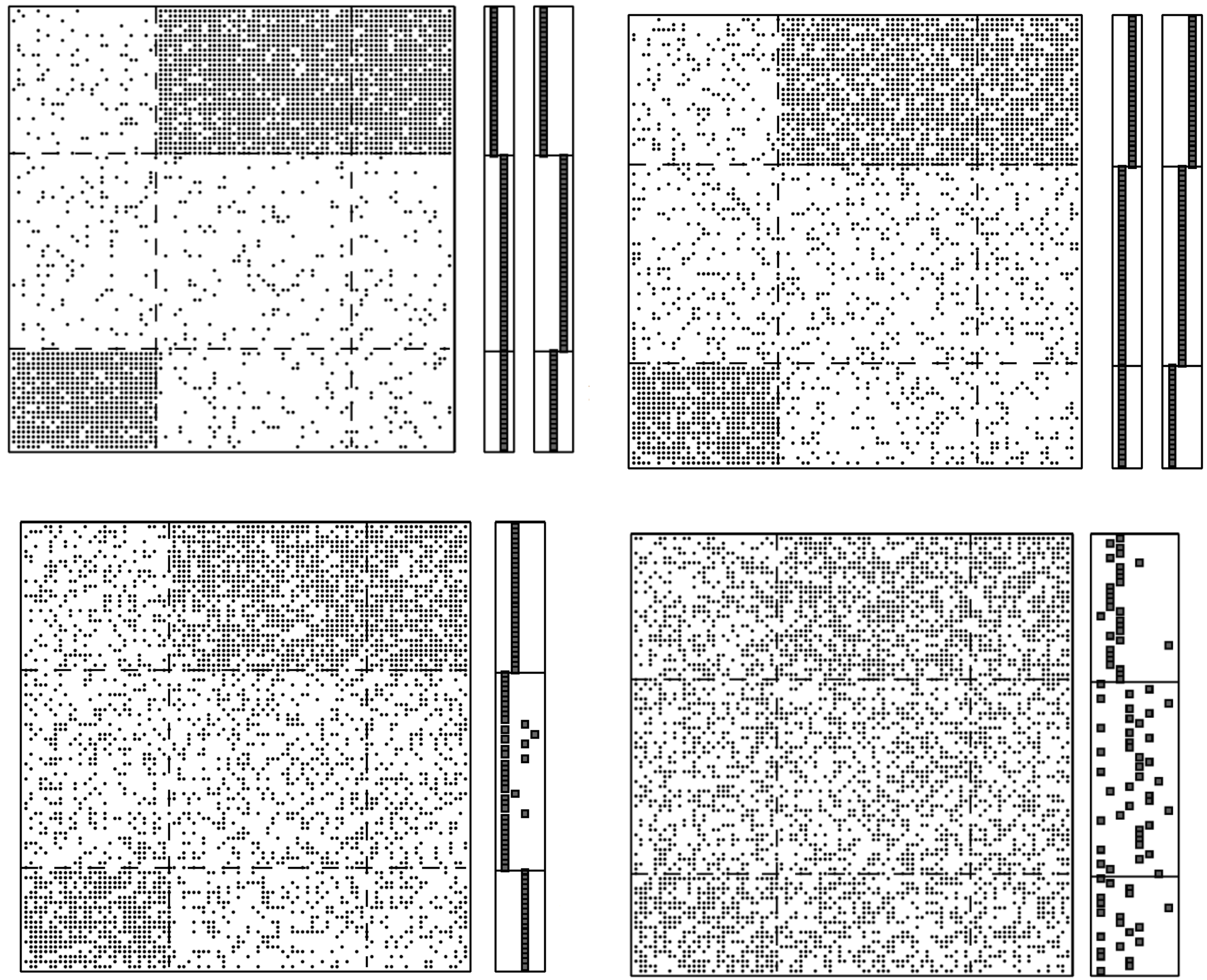
Results





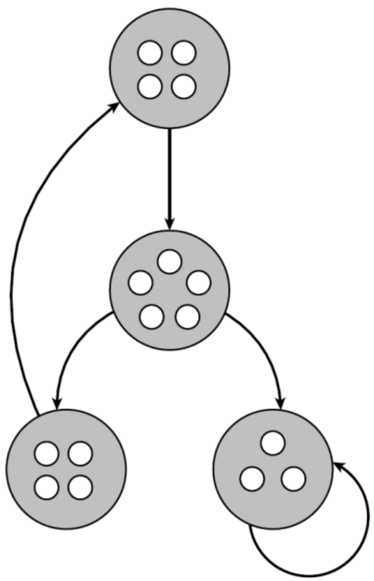
Erdos-Reyni random graphs
Results

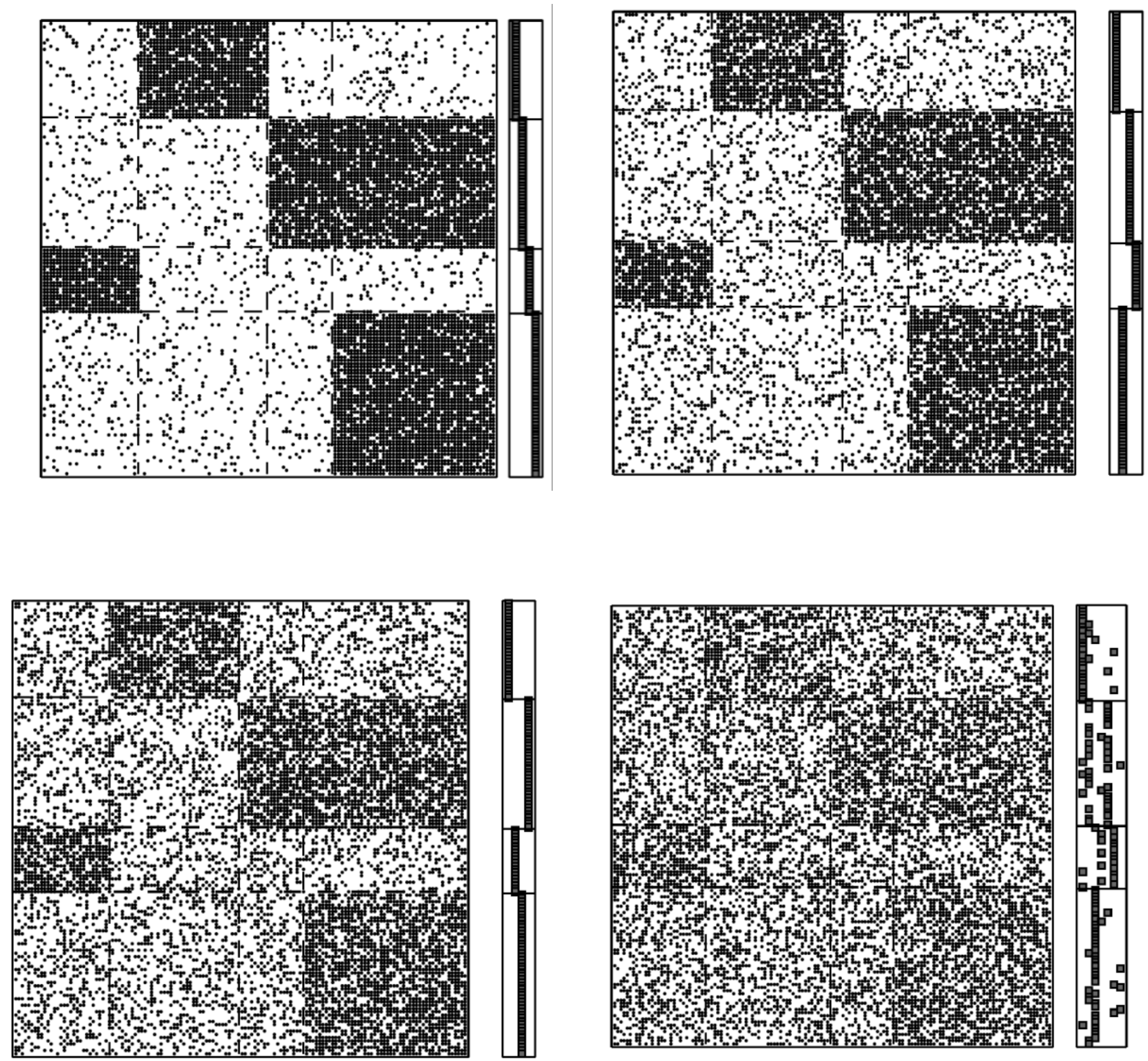




Erdos-Reyni random graphs

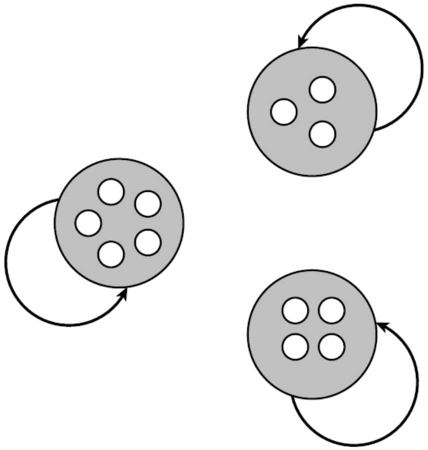
Results

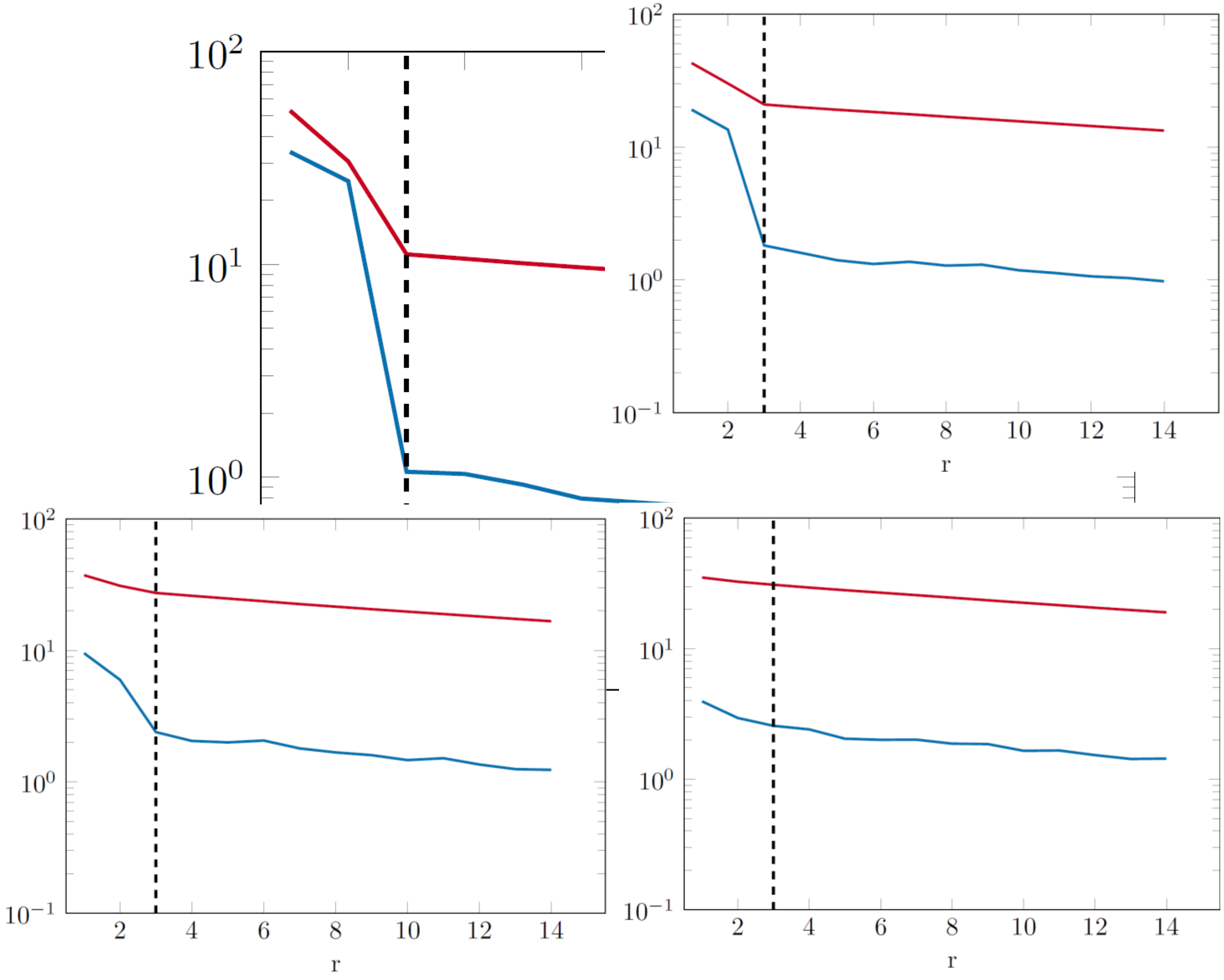




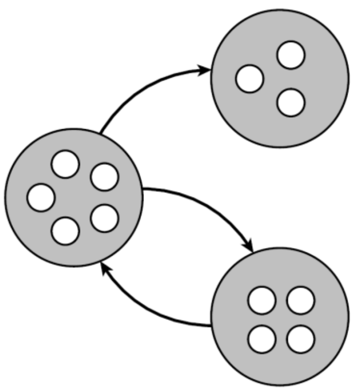
Erdos-Reyni random graphs

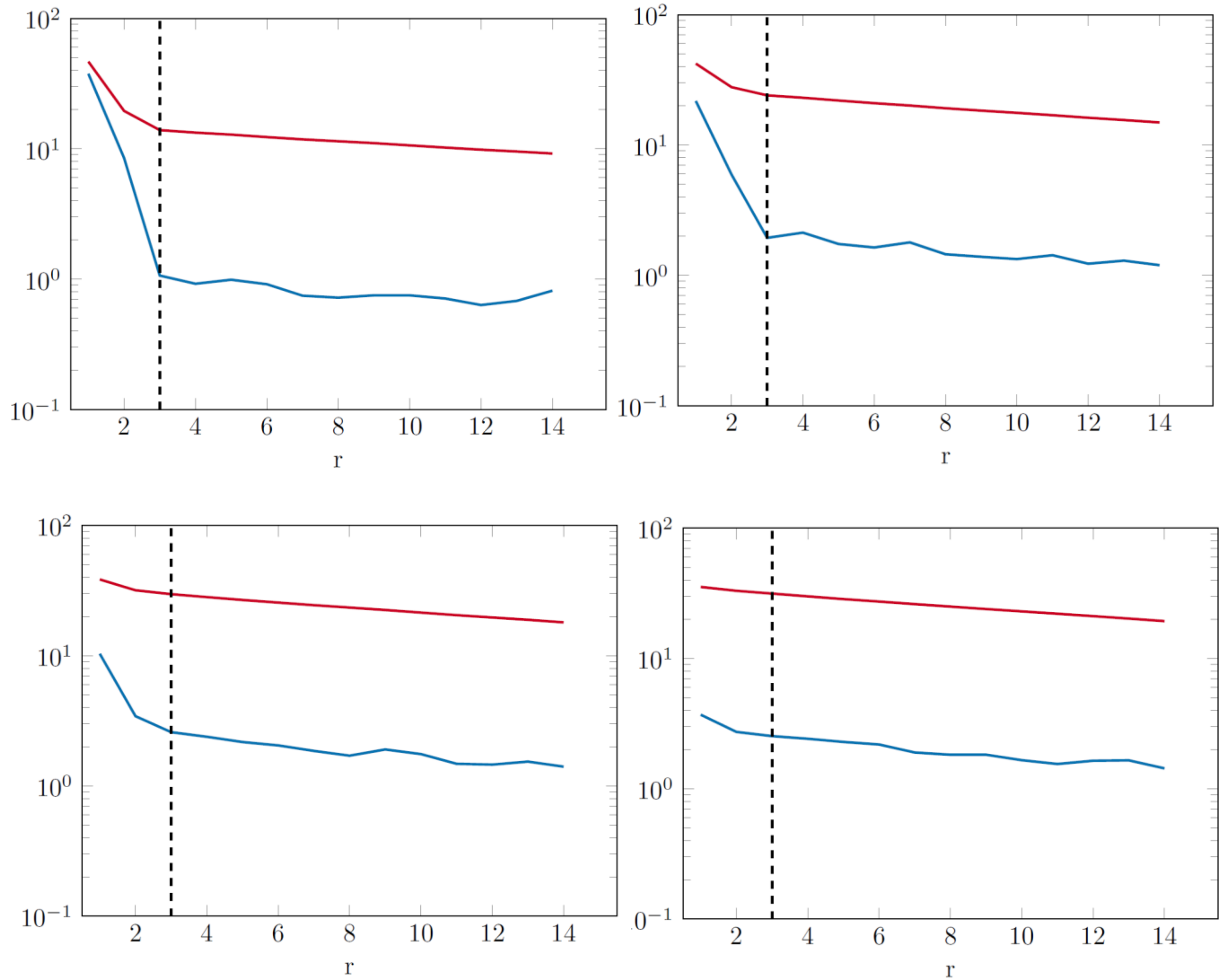
Results



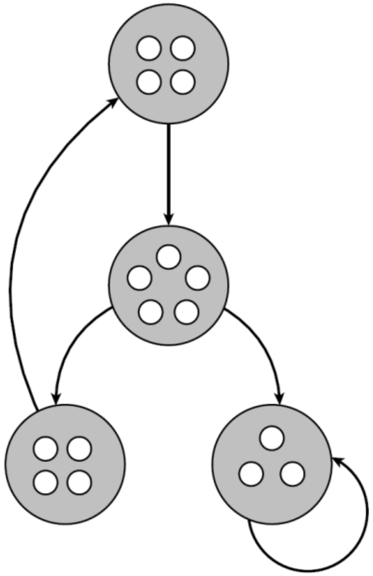


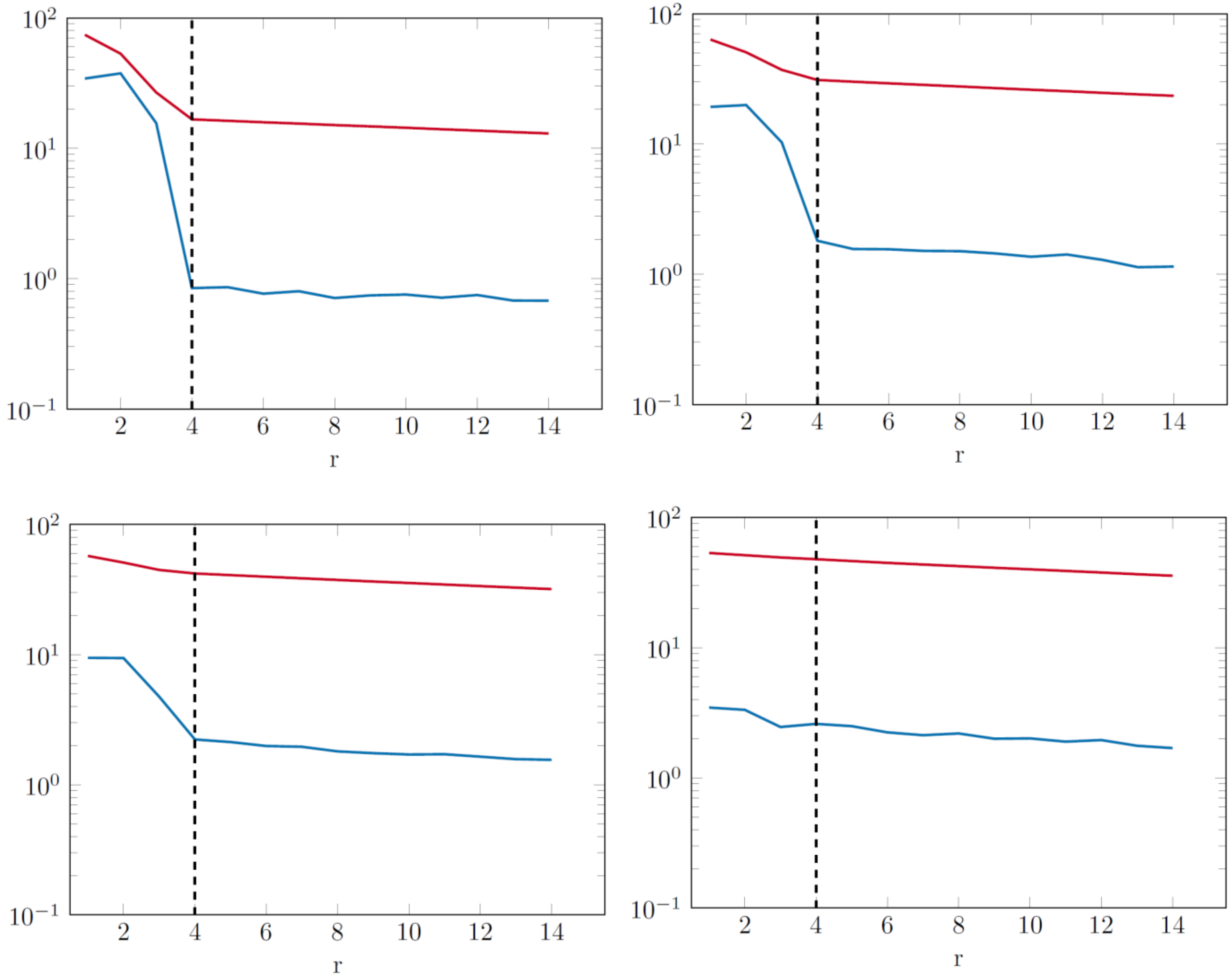
Erdos-Reyni random graphs
Results





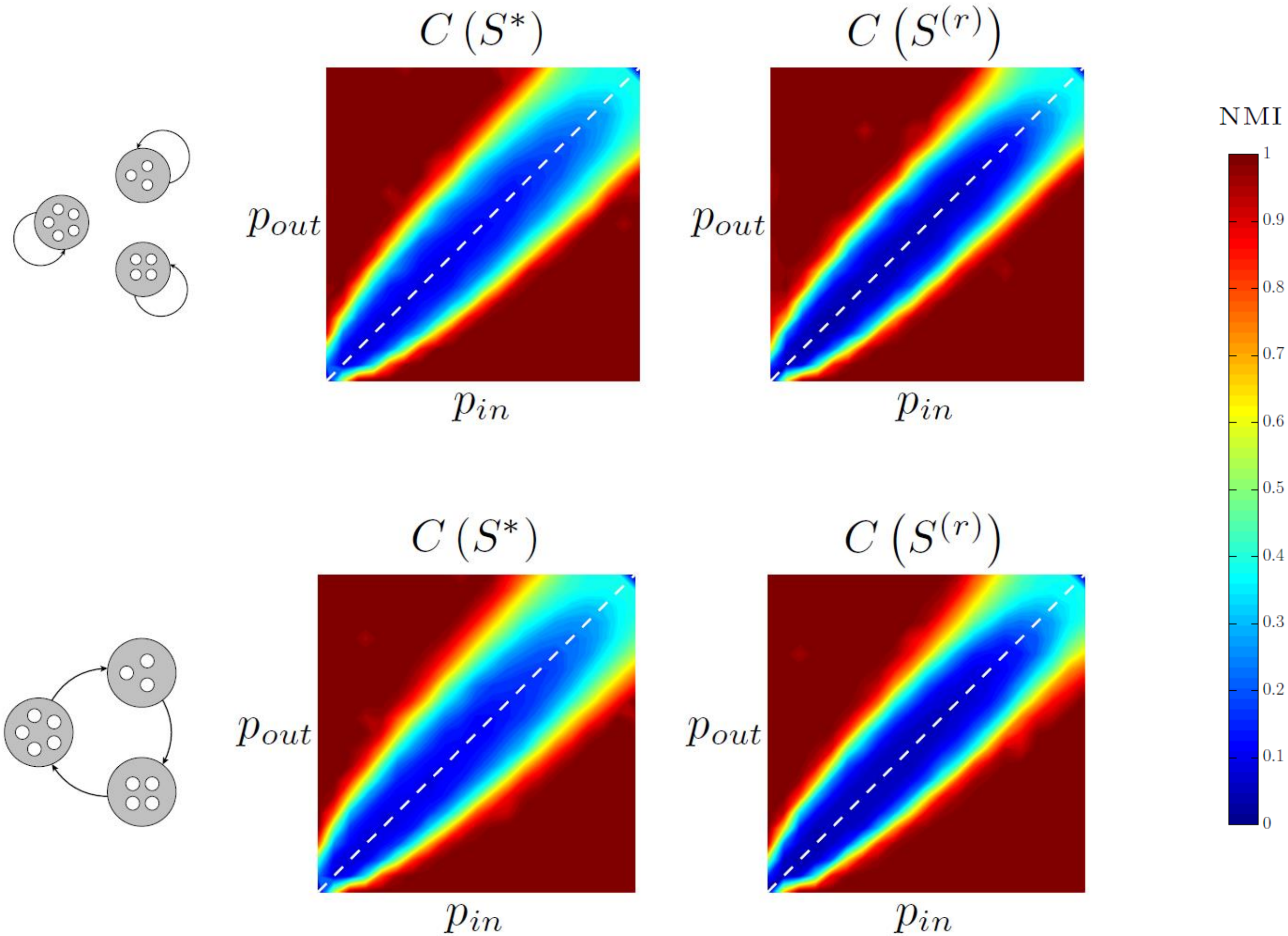
Erdos-Reyni random graphs
Results



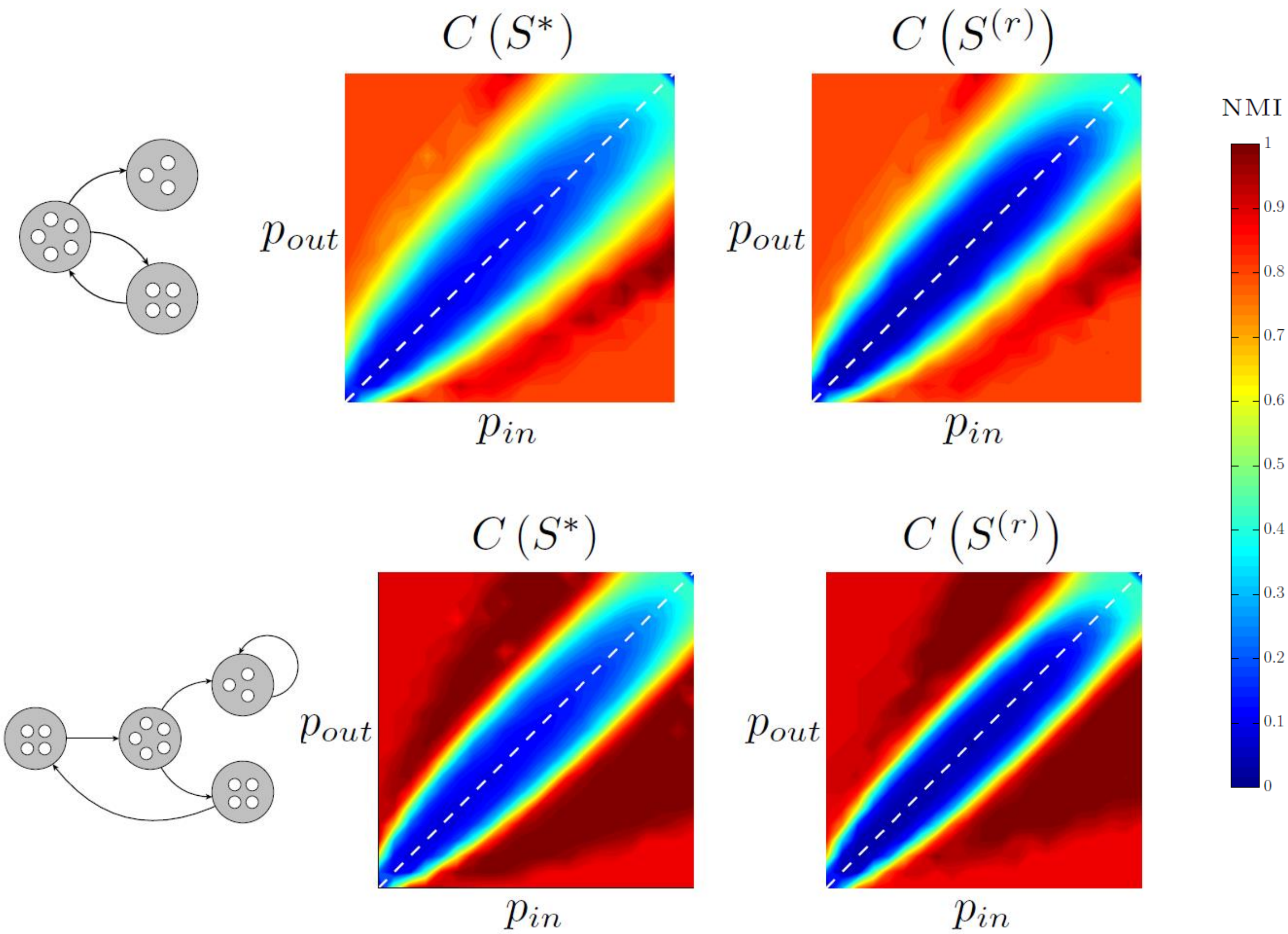


Erdos-Reyni random graphs

Results

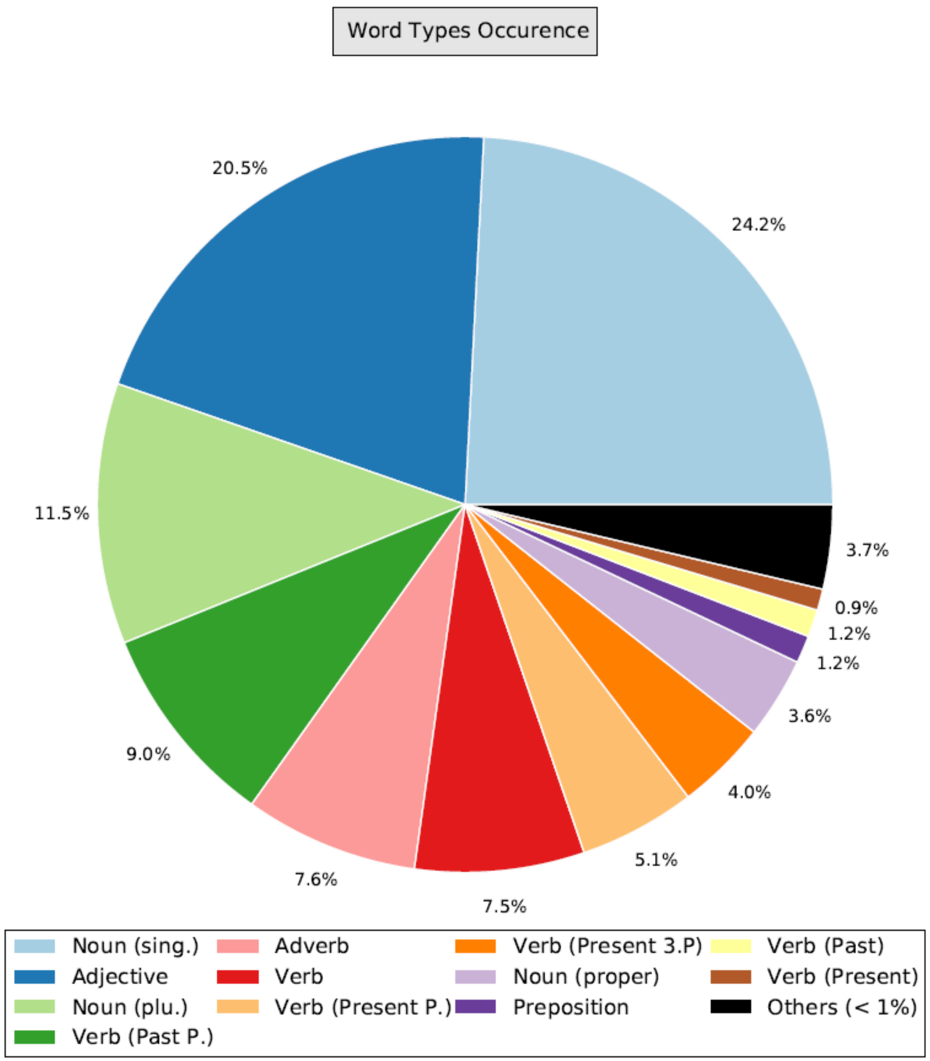


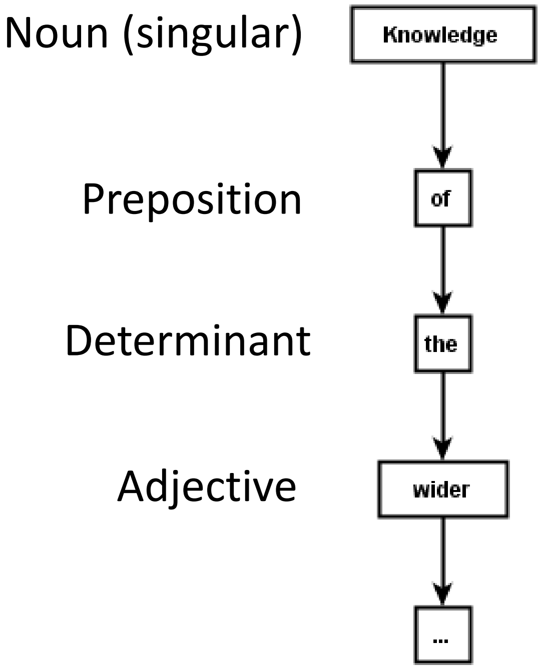
Erdos-Reyni random graphs
Results



Words graph

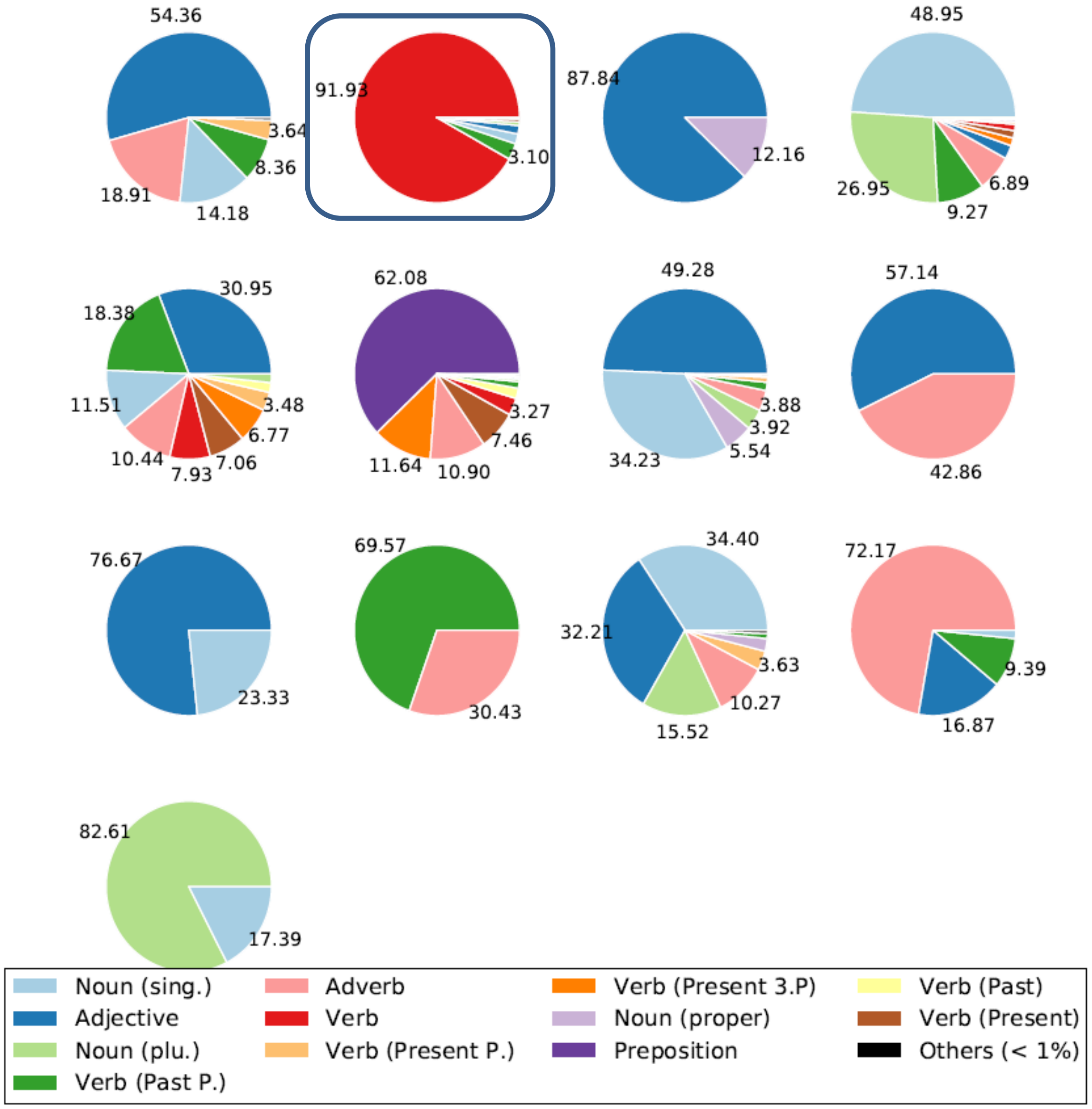
« Language, an introduction to the study of speech » *Sapir. (1972)*





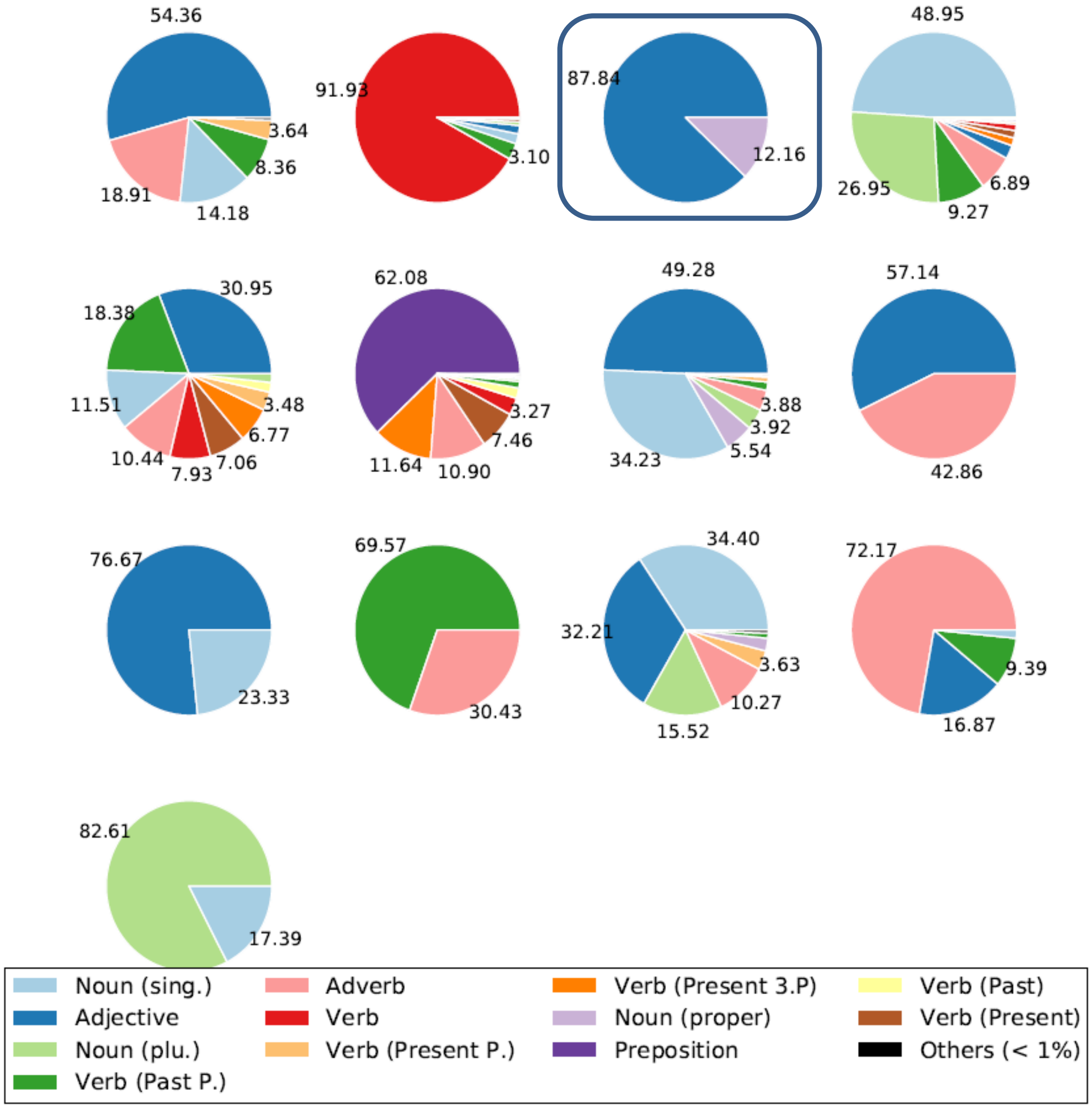
Python lib: NLTK
+ Stanford PoS Tagger

Words graph



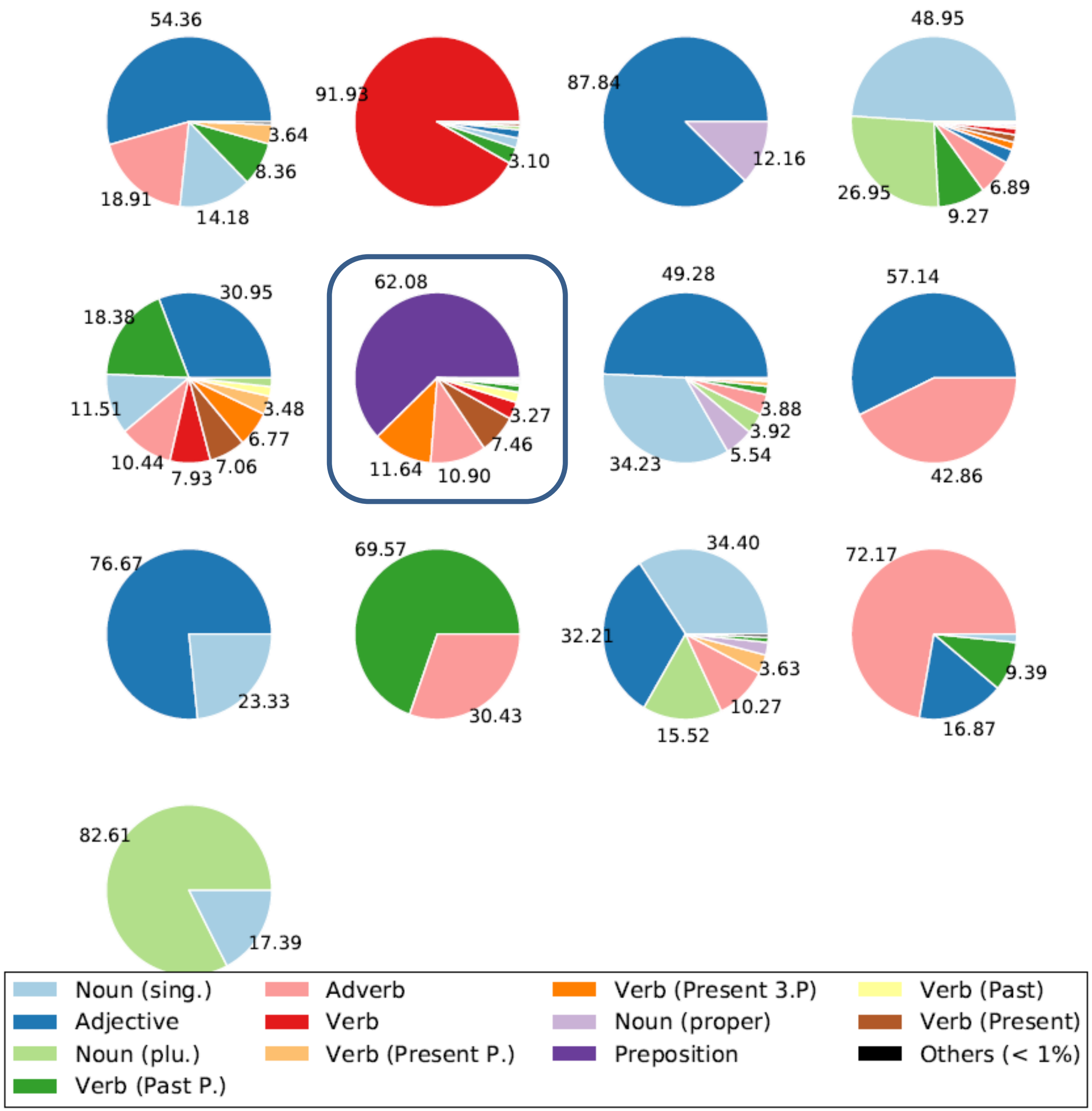
Cluster 2
be (582)
say (128)
go (37)
become (33)
him (28)
show (24)
believe (23)
me (17)
animate (17)
indicate (16)
look (15)
fall (14)
observe (14)
serve (12)
run (12)

Words graph



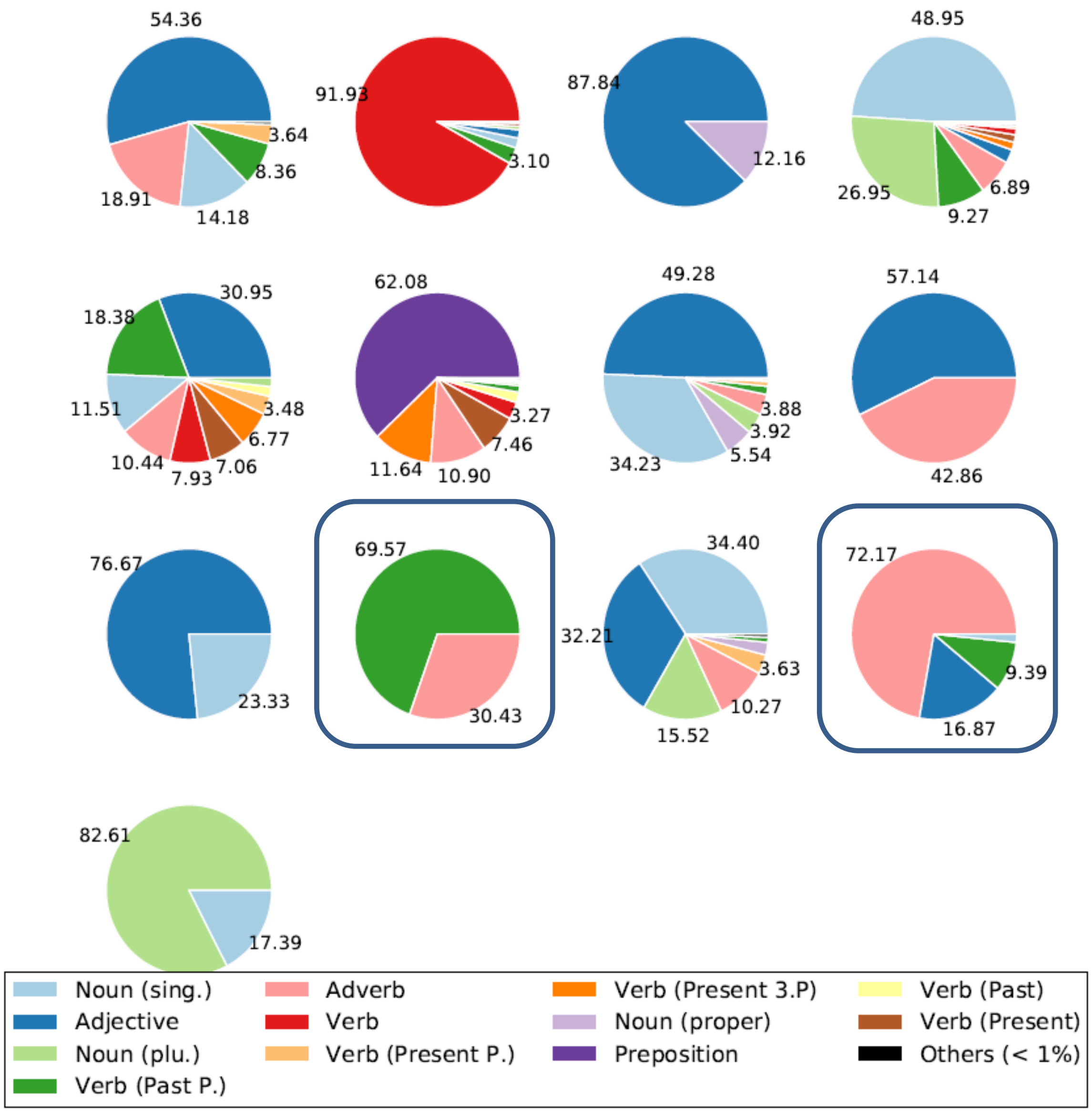
Cluster 3
greek (48)
elusive (10)
cambodgian (9)
archaic (7)
satisfying (5)
grouped (4)
siamese (4)
nowhere (4)
inclusive (4)
explicit (4)
religious (4)
infixes (4)
treated (4)
formless (4)
syllabic (3)

Words graph



Cluster 6
of (3636)
to (1820)
in (1647)
is (1557)
and (1514)
that (1220)
as (1100)
or (795)
are (766)
not (607)
by (411)
with (391)
but (387)
for (366)
have (364)

Words graph

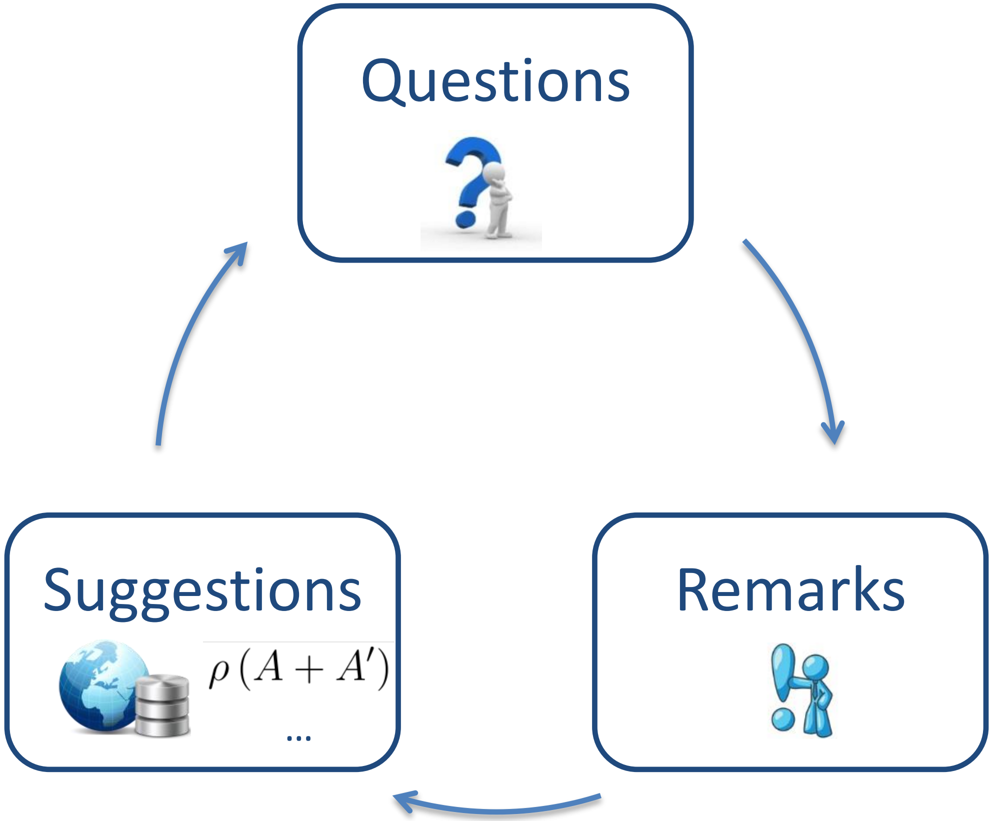


Cluster 10
looked (16)
conveniently (7)
enormously (4)
rapidly (4)
eliminated (4)
termed (4)
barely (4)
obsolete (3)
sleeping (3)
swept (2)
subdivided (2)
multiplied (2)
diffused (2)
tired (2)
abundantly (1)

Cluster 12
only (137)
still (64)
too (60)
entirely (34)
never (33)
hardly (31)
identical (25)
really (22)
clear (16)
concerned (15)
red (13)
obvious (12)
indicated (11)
constantly (10)
done (10)

Take Home

- ✓ Efficient and highly parallelizable algorithm for community detection
- ✓ Role Extraction or Block Modeling generalized community detection
- ✓ The pairwise node similarity measure allows to extract such roles
- ✓ Accurate low rank approximation for large graphs


Questions
Remarks
Suggestions
$\rho(A + A')$
...