Two Vignettes from the Interface of Learning and Optimization

Jiaming Xu

Krannert School of Management
Purdue University

Workshop in Operations/Management Science
February 27, 2018

Talk outline

1. Operate online service platforms with uncertain payoff and dynamic
    - Learning: unknown true payoffs
    - Optimization: service allocation

2. Recover a hidden Hamiltonian cycle in a network
    - Learning: unknown Hamiltonian cycle
    - Optimization: Travelling salesman problem

Operate online service platforms with uncertain payoff and dynamic

joint work with Wei-Kang Hsu, Xiaojun Lin, and Mark R. Bell (Purdue ECE)

A proliferation of online service platforms

Goal: assign client to server to maximize total payoffs subj. to capacity

- Clients: Advertiser
- Servers: Keyword searches
- Payoffs: Click-through-rate

A proliferation of online service platforms

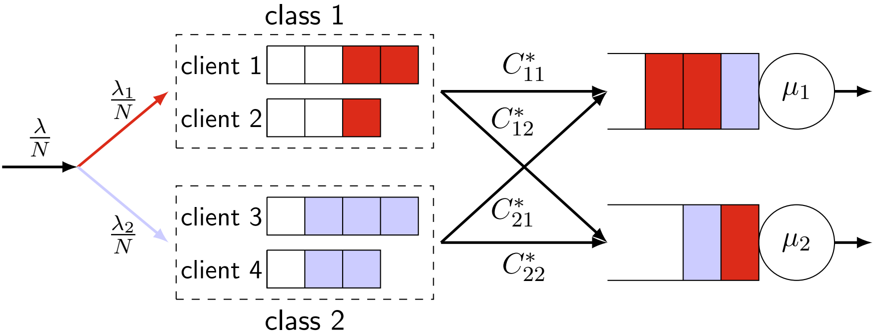
Goal: assign client to server to maximize total payoffs subj. to capacity

- Clients: Advertiser
- Servers: Keyword searches
- Payoffs: Click-through-rate

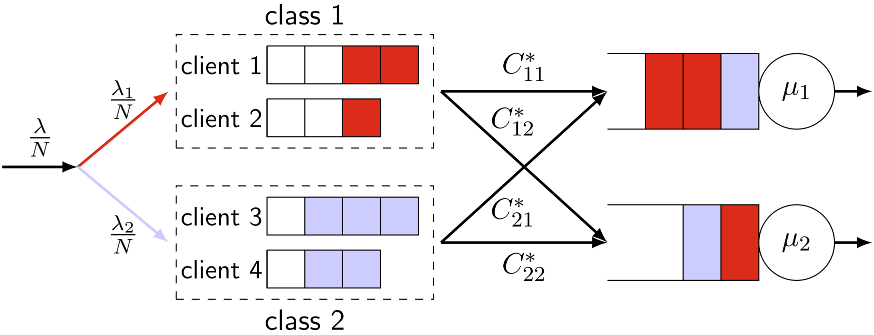
Two key challenges

- unknown payoff $\rightarrow$ learn payoffs from noisy feedback
- uncertain client dynamics $\rightarrow$ adaptively control assignments

Queueing model with uncertain dynamic and payoff

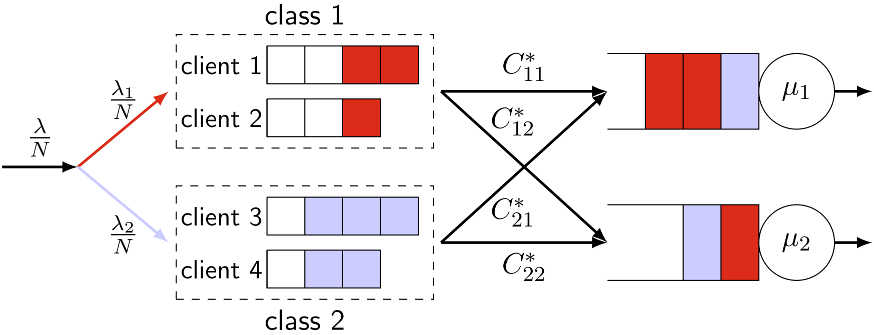


Queueing model with uncertain dynamic and payoff



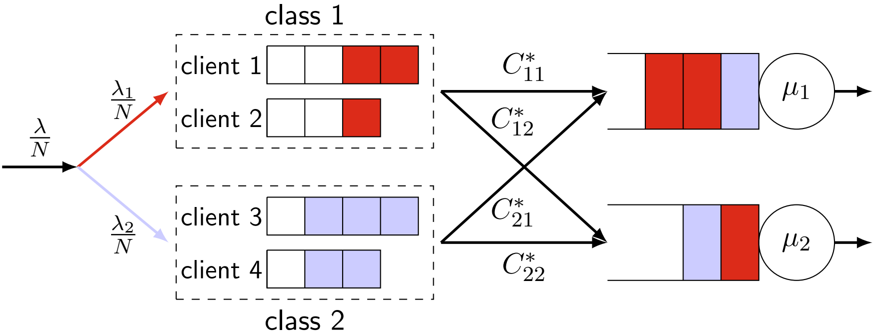
- New client (Advertiser) carries $\mathrm{Geom}(N)$ number of tasks (Ads)

Queueing model with uncertain dynamic and payoff



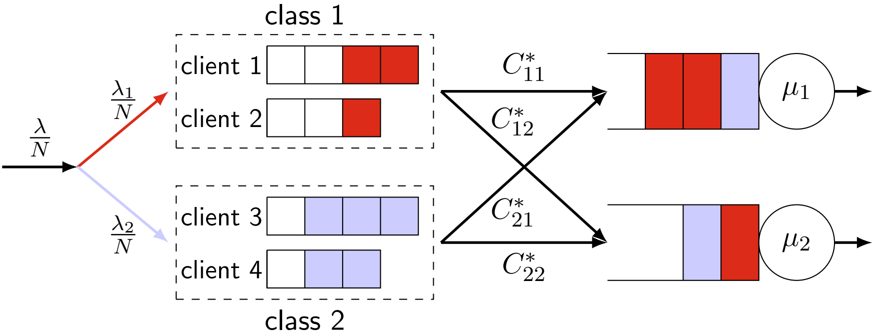
- New client (Advertiser) carries $\mathrm{Geom}(N)$ number of tasks (Ads)
- Server (keyword searches) j serve μ_j tasks per slot

Queueing model with uncertain dynamic and payoff



- New client (Advertiser) carries $\mathrm{Geom}(N)$ number of tasks (Ads)
- Server (keyword searches) j serve μ_j tasks per slot
- Observe $\mathrm{Bern}(C^*_{ij})$ payoff after a task of class i departs at server j

Queueing model with uncertain dynamic and payoff



- New client (Advertiser) carries $\mathrm{Geom}(N)$ number of tasks (Ads)
- Server (keyword searches) j serve μ_j tasks per slot
- Observe $\mathrm{Bern}(C^*_{ij})$ payoff after a task of class i departs at server j
- Do not know λ_i, N, class label, $\#$ of tasks, or payoff vectors

Performance metric and oracle bound

Expected payoff per unit time of a policy Π:

$$R_T(\Pi) = \frac{1}{T} \sum_{t=1}^{T} \sum_{j=1}^{J} \mathbb{E}\left[\sum_{l=1}^{n(t)} p_j^l(t) C^*_{i(l),j} \right]$$

$p_j^l(t)$: mean number of tasks assigned to server j from client l at time t

Performance metric and oracle bound

Expected payoff per unit time of a policy Π:

$$R_T(\Pi) = \frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{J}\mathbb{E}\left[\sum_{l=1}^{n(t)} p_j^l(t) C^*_{i(l),j}\right]$$

$p_j^l(t)$: mean number of tasks assigned to server j from client l at time t

Oracle LP bound with perfect information:

$$\begin{aligned} R^* = \max_{[p_{ij}]\geq 0} \quad & \sum_{i=1}^{I}\lambda_i \sum_{j=1}^{J} p_{ij} C^*_{ij} \\ \text{s.t.} \quad & \sum_{i=1}^{I}\lambda_i p_{ij} \leq \mu_j \text{ for all servers } j = 1,\ldots,J \\ & \sum_{j=1}^{J} p_{ij} = 1 \text{ for all classes } i = 1,\ldots,I \end{aligned}$$

Performance metric and oracle bound

Expected payoff per unit time of a policy Π:

$$R_T(\Pi) = \frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{J}\mathbb{E}\left[\sum_{l=1}^{n(t)} p_j^l(t) C^*_{i(l),j}\right]$$

$p_j^l(t)$: mean number of tasks assigned to server j from client l at time t

Oracle LP bound with perfect information:

$$\begin{aligned} R^* = \max_{[p_{ij}]\geq 0} \quad & \sum_{i=1}^{I}\lambda_i \sum_{j=1}^{J} p_{ij} C^*_{ij} \\ \text{s.t.} \quad & \sum_{i=1}^{I}\lambda_i p_{ij} \leq \mu_j \text{ for all servers } j = 1, \ldots, J \\ & \sum_{j=1}^{J} p_{ij} = 1 \text{ for all classes } i = 1, \ldots, I \end{aligned}$$

Weakness of queue length based control

With known payoff vectors [Tassiulas and Ephremides '92, ...]

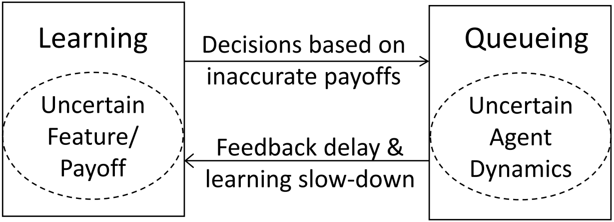
1. Use queue length q_j at server j to capture congestion level
2. Adjust each client's payoff parameter by subtracting q_j/V
3. Assign the next task to the server with the highest adjusted payoff

Weakness of queue length based control

With known payoff vectors [Tassiulas and Ephremides '92, ...]

1. Use queue length q_j at server j to capture congestion level
2. Adjust each client's payoff parameter by subtracting q_j/V
3. Assign the next task to the server with the highest adjusted payoff

With unknown payoff vectors: vicious cycle



Weakness of myopic matching

$$\max_{[p_j^l]\geq 0} \quad \sum_{l=1}^{n(t)} \sum_{j=1}^{J} p_j^l \, C_j^l(t)$$

$$\text{s.t.} \quad \sum_{l=1}^{n(t)} p_j^l \leq \mu_j \ \text{ for all servers } j = 1, \ldots, J$$

- No task-queue at servers $\rightarrow$ no payoff feedback delay

Weakness of myopic matching

$$\max_{[p_j^l]\geq 0} \quad \sum_{l=1}^{n(t)} \sum_{j=1}^{J} p_j^l \, C_j^l(t)$$

$$\text{s.t.} \quad \sum_{l=1}^{n(t)} p_j^l \leq \mu_j \ \text{ for all servers } j = 1, \ldots, J$$

- No task-queue at servers $\rightarrow$ no payoff feedback delay
- Payoff estimate $C_j^l(t)$ based on Upper-Confidence-Bound (UCB) [Lai-Robbins '85]:

$$C_j^l(t) = \min \left\{ \overline{C}_j^l(t-1) + \sqrt{\frac{2 \log h^l(t-1)}{h_j^l(t-1)}}, \ 1 \right\}$$

$\overline{C}_j^l(t-1)$: empirical average payoff of client l;
$h_j^l(t-1)$: # of tasks assigned to server j from client l

Weakness of myopic matching

$$\max_{[p_j^l]\geq 0} \quad \sum_{l=1}^{n(t)} \sum_{j=1}^{J} p_j^l \, C_j^l(t)$$

$$\text{s.t.} \quad \sum_{l=1}^{n(t)} p_j^l \leq \mu_j \quad \text{for all servers } j = 1, \ldots, J$$

- No task-queue at servers $\rightarrow$ no payoff feedback delay
- Payoff estimate $C_j^l(t)$ based on Upper-Confidence-Bound (UCB) [Lai-Robbins '85]:

$$C_j^l(t) = \min \left\{ \overline{C}_j^l(t-1) + \sqrt{\frac{2 \log h^l(t-1)}{h_j^l(t-1)}}, \; 1 \right\}$$

$\overline{C}_j^l(t-1)$: empirical average payoff of client l;
$h_j^l(t-1)$: # of tasks assigned to server j from client l

- Do not look into future and hence long-term payoff is suboptimal!

Our approach based on utility optimization

$$\max_{[p_j^l]\geq 0} \quad \sum_{l=1}^{n(t)} \left\{ \frac{1}{V} \log\left(\sum_{j=1}^{J} p_j^l \right) + \sum_{j=1}^{J} p_j^l \left(C_j^l(t) - \gamma \right) \right\}$$

$$\text{s.t.} \quad \sum_{l=1}^{n(t)} p_j^l \leq \mu_j \quad \text{for all servers } j = 1, \ldots, J \; .$$

- Log utility function promotes fairness $\rightarrow$ every client can learn
- $V > 0$: as V increases
  - clients are more conservative in choosing low-payoff servers
  - more clients are backlogged in the system
- $\gamma > 1$: prevent clients choosing low-payoff servers too aggressively
- Inspired by flow-level congestion control in communication networks [Lin-Shroff-Srikant '08]

Performance guarantees of our policy

λ : total arrival rate $\quad$ μ : total service rate $\quad$ $n(t)$: # of clients

Theorem (mean number of backlogged clients)

$$\mathbb{E}[n(t)] \leq \frac{2\mu}{\mu - \lambda}\left(1 + \frac{\mu^2\gamma}{\gamma - 1}\right) + \mu\gamma V.$$

Performance guarantees of our policy

λ : total arrival rate $\quad \mu$: total service rate $\quad n(t)$: # of clients

Theorem (mean number of backlogged clients)

$$\mathbb{E}\left[n(t)\right] \le \frac{2\mu}{\mu - \lambda}\left(1 + \frac{\mu^2\gamma}{\gamma - 1}\right) + \mu\gamma V.$$

- implies system is stable
- mean number of backlogged clients increases linearly in V
- Proof: couple to a Geom/Geom/μ queue with Bernoulli arrivals and Binomial departures

Performance guarantees of our policy

Theorem (Payoff gap to oracle bound)

$$R^* - R_T \le \frac{\beta_1}{V} + \beta_2\sqrt{\frac{\log N}{N}} + \beta_3\frac{N(V+1)}{T}$$

- $\beta_1, \beta_2, \beta_3$: functions of λ_i, μ_j, γ
- $1/V$: Impact of the uncertainty in client dynamics
- $\sqrt{\log N/N}$: Impact of the uncertainty in payoffs
- $N(V+1)/T$: Payoff loss incurred by backlogged tasks
- Captures the transient behavior in finite T in contrast to study of stationary regime in [Johari-Kamble-Kanoria '17]

Proof ideas

1. Use Lyapunov drift analysis to show

$$R^* - R_T \lesssim \frac{1}{V} + \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[A(t)\right] + \frac{N}{T}\mathbb{E}\left[n(T)\right],$$

$$\text{where } A(t) = \sum_{l=1}^{n(t)} \sum_{j=1}^{J} \underbrace{\left(C_j^l(t) - C^*_{i(l),j}\right)}_{\text{learning error}} \underbrace{\left(p_j^l(t) - \widetilde{p}_j^l(t)\right)}_{\text{controlling error}}$$

$\widetilde{p}_j^l(t)$: optimal assignment if our policy knew true payoffs

2. Use duality +UCB regret analysis + martingale argument to show

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[A(t)\right] \lesssim \sqrt{\frac{\log N}{N}}$$

To show

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\sum_{l=1}^{n(t)}\sum_{j=1}^{J}\left(C_j^l(t)-C^*_{i(l),j}\right)\left(p_j^l(t)-\widetilde{p}_j^l(t)\right)\right]\lesssim\frac{\sqrt{N\log N}}{N}$$

1. Convex duality

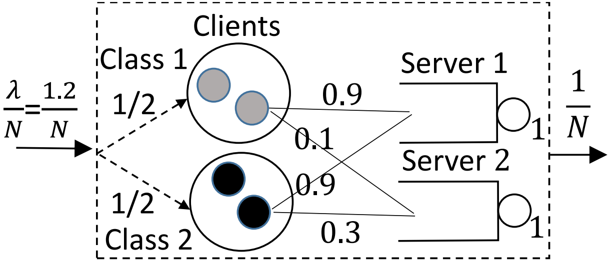
$$\frac{\sum_{j=1}^{J}\widetilde{p}_j^l(t)}{\sum_{j=1}^{J}p_j^l(t)}\le\left(\frac{\gamma}{\gamma-1}\right)^2$$

2. UCB regret analysis

$$\mathbb{E}\left[\left(C_j^l(t)-C^*_{i(l),j}\right)p_j^l(t)\right]\lesssim\sum_{k=1}^{N}\sqrt{\frac{\log k}{k}}\lesssim\sqrt{N\log N}$$

Use martingale argument to take care of dependency between $p_j^l(t)$ and $\{C_j^l(s):s\le t\}$

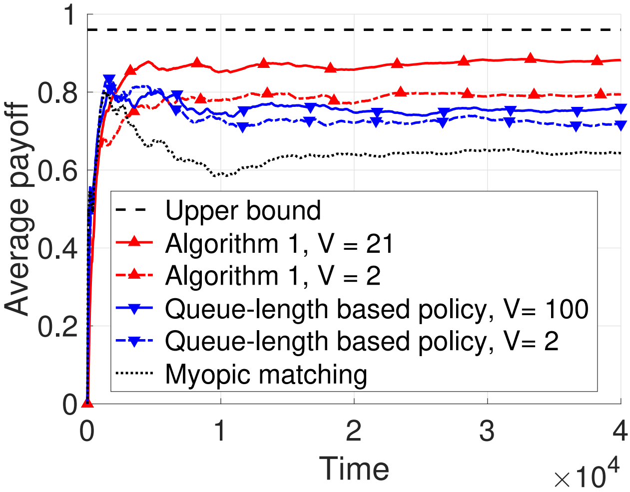
Numerical results: simulation setup



Oracle solution:

$$p_{11}^* = 1, \quad p_{12}^* = 0, \quad p_{21}^* = \frac{2}{3}, \quad p_{22}^* = \frac{1}{3}, \quad R^* = 0.96$$

Numerical results: performance comparison



$N = 100$ and $\gamma = 1.1$ and $R^* = 0.96$

Related literature

Learning and adaptive control seperately

- Multi-armed bandits: [Lai-Robbins '85], [Auer-Cesa-Bianchi-Fischer '02],...
- Adaptive control: [Tassiulas-Ephremides '92], [Neely-Modiano-Li '05],...

Related literature

Learning and adaptive control seperately

- Multi-armed bandits: [Lai-Robbins '85], [Auer-Cesa-Bianchi-Fischer '02],...
- Adaptive control: [Tassiulas-Ephremides '92], [Neely-Modiano-Li '05],...

Integrate learning and adaptive control

- Online matching while learning: [Johari-Kamble-Kanoria '17]
  - Stationary setting with known arrival rates and class-dependent payoff vectors
  - Divide learning and adaptive control into two stages
- Learning unknown labels with capacity constraints: [Xu-Massoulié '16]
- Processing tasks of unknown types with capacity constraints: [Bimpikis-Markakis '15], [Shah-Gulikers-Massoulié-Vojnovic '17]

Conclusion remarks

Propose an online learning and adaptive control policy based on utility optimization:

$$\text{payoff gap} \lesssim \underbrace{\frac{1}{V}}_{\text{uncertain dynamics}} + \underbrace{\sqrt{\frac{\log N}{N}}}_{\text{uncertain payoffs}} + \underbrace{\frac{N(V+1)}{T}}_{\text{backlogged tasks}}$$

Conclusion remarks

Propose an online learning and adaptive control policy based on utility optimization:

$$\text{payoff gap} \lesssim \underbrace{\frac{1}{V}}_{\text{uncertain dynamics}} + \underbrace{\sqrt{\frac{\log N}{N}}}_{\text{uncertain payoffs}} + \underbrace{\frac{N(V+1)}{T}}_{\text{backlogged tasks}}$$

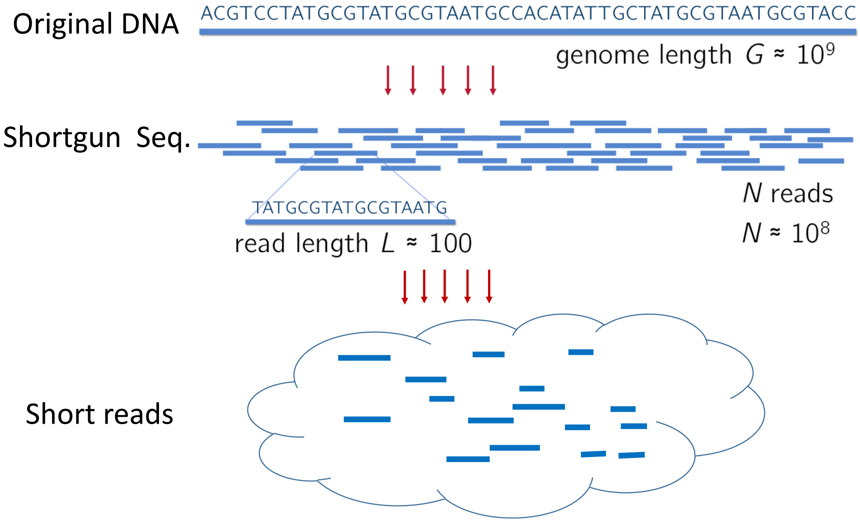
Future work

- Improve the exploration-exploitation tradeoff to $\log N/N$
- Adapt to random service time or unknown service rates

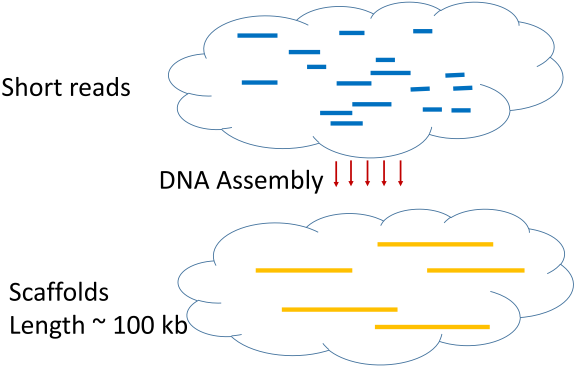
Recover a Hidden Hamiltonian Cycle via Linear Programming

joint work with V. Bagaria, David Tse (Stanford), J. Ding (Wharton), Y. Wu (Yale)

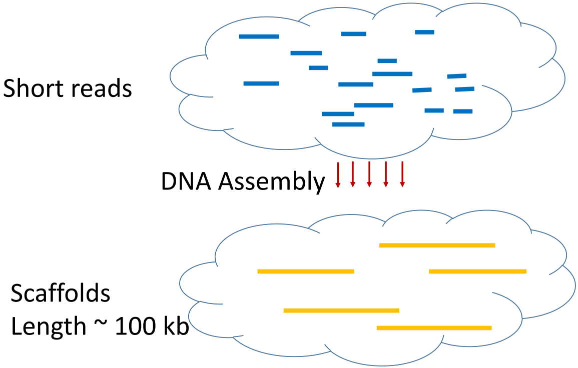
DNA high-throughput sequencing



Key challenge in DNA high-throughput sequencing



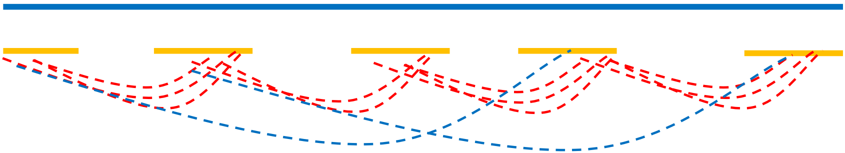
Key challenge in DNA high-throughput sequencing



High-throughput sequencing has low contiguity!

Boost contiguity: cross-links in Chicago datasets

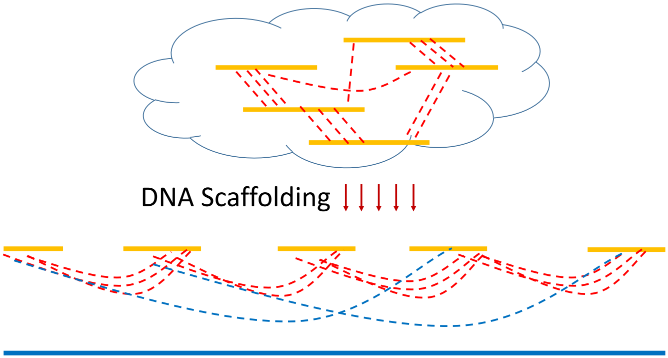
1. Reconstitute chromatin in vitro upon naked DNA
2. Produce cross-links by fixing chromatin with formaldehyde



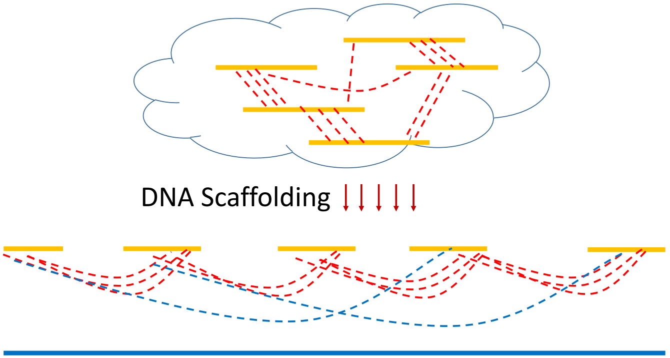
Chicago datasets generate cross-links among contigs [Putnam et al. '16]

On average more cross-links exist between adjacent contigs

Ordering DNA contigs with Chicago cross-links



Ordering DNA contigs with Chicago cross-links



Reduces to travelling salesman problem (TSP):

Find a path (tour) to visit every contig exactly once with the maximum number of cross-links

Key challenges for DNA scaffolding with Chicago data

- Computational: TSP is NP-hard in the worst-case
- Statistical: spurious cross-links between contigs far apart

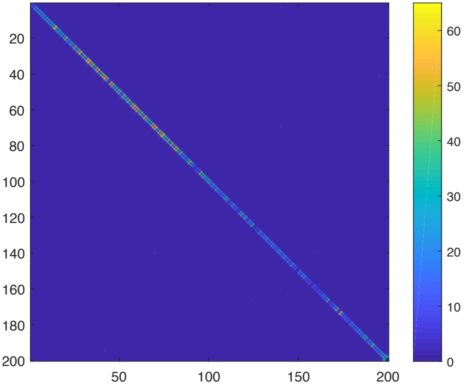
Key challenges for DNA scaffolding with Chicago data

- Computational: TSP is NP-hard in the worst-case
- Statistical: spurious cross-links between contigs far apart

Key questions:

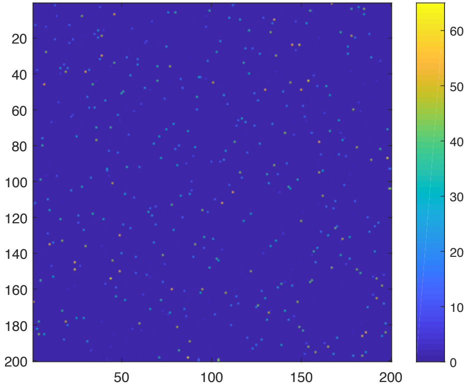
- How to efficiently order hundreds of thousands of contigs?
- How much noise can be tolerated for accurate DNA scaffolding?

Our mathematical model for DNA scaffolding



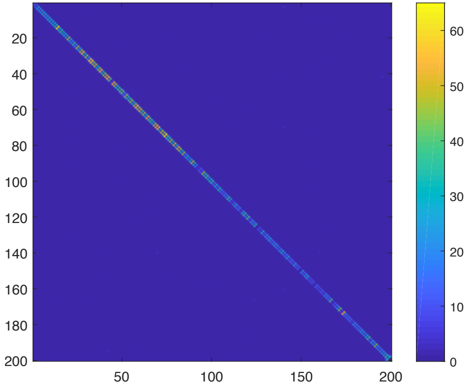
Real DNA data [Putnam et al. '16]

Our mathematical model for DNA scaffolding

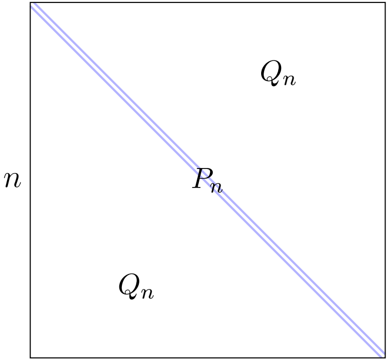


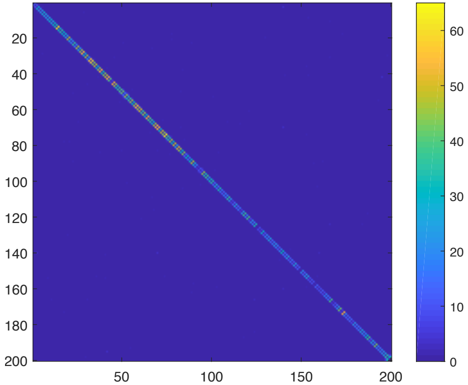
Real DNA data [Putnam et al. '16]

Our mathematical model for DNA scaffolding

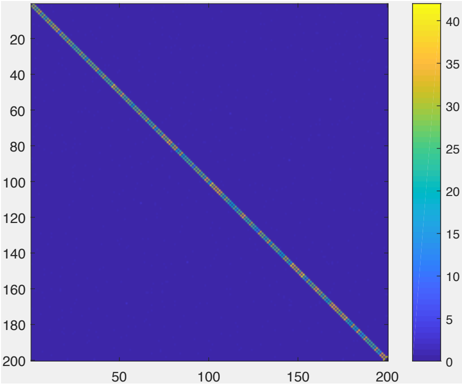


Real DNA data [Putnam et al. '16]



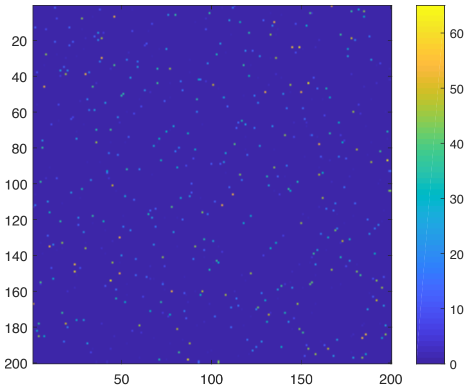


Real DNA data [Putnam et al. '16]

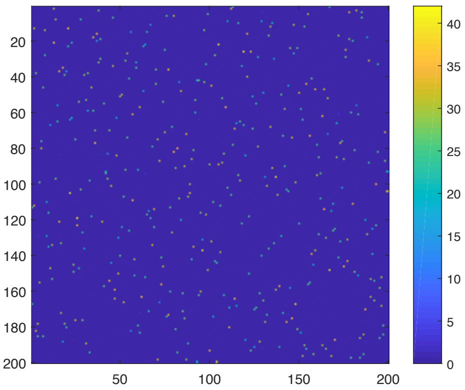


$P_n = \mathrm{Pois}(\lambda_1)$, $Q_n = \mathrm{Pois}(\lambda_2)$

Our mathematical model for DNA scaffolding



Real DNA data [Putnam et al. '16]



recover hidden Hamiltonian cycle

What is known information-theoretically

Consider the Gaussian case $P = \mathcal{N}(\mu, 1)$ and $Q = \mathcal{N}(0, 1)$

Theorem (Bagaria-Ding-Tse-Wu-X. '18)

If

$$\frac{\mu^2}{\log n} > 4,$$

exact recovery is information-theoretically possible.
Conversely, if

$$\frac{\mu^2}{\log n} < 4,$$

then exact recovery is impossible.

What is known algorithmically

- Spectral methods fails miserably:
  - $\mu^2 \gg n^5$ (spectral gap of cycle is too small)

What is known algorithmically

- Spectral methods fails miserably:
  - $\mu^2 \gg n^5$ (spectral gap of cycle is too small)

- Thresholding or nearest-neighbor method:
  - $\mu^2 > 8 \log n$

What is known algorithmically

- Spectral methods fails miserably:
  - $\mu^2 \gg n^5$ (spectral gap of cycle is too small)
- Thresholding or nearest-neighbor method:
  - $\mu^2 > 8 \log n$
- Greedy merging [Motahari-Bresler-Tse '13]:
  - $\mu^2 > 6 \log n$

What is known algorithmically

- Spectral methods fails miserably:
  - $\mu^2 \gg n^5$ (spectral gap of cycle is too small)
- Thresholding or nearest-neighbor method:
  - $\mu^2 > 8 \log n$
- Greedy merging [Motahari-Bresler-Tse '13]:
  - $\mu^2 > 6 \log n$

Suboptimal comparing to IT-limit $\mu^2 > 4 \log n$!

Maximum likelihood estimation reduces to TSP

$$\widehat{x}_{\mathrm{ML}} = \arg\max_{x} \langle w, x \rangle$$
$$\text{s.t.} \quad x \text{ is an adjacency vector of a Hamiltonian cycle}$$

- Find a maximum weighted Hamiltonian cycle $\Longleftrightarrow$ TSP
- NP hard!

Fractional 2-factor linear relaxation of TSP

$$\begin{aligned} \widehat{x}_{\mathrm{F2F}} = \arg\max_{x} \ & \langle w, x \rangle \\ \text{s.t.} \ & \sum_{e \in \delta(v)} x_e = 2 \quad \forall \text{ vertex } v \\ & x_e \in [0,1] \quad \forall \text{ edge } e \end{aligned}$$

- Extensively studied in worst case [Schalekamp-Williamson-van Zuylen '14]
- The integrality gap $\frac{\mathrm{2F}}{\mathrm{F2F}} \leq \frac{4}{3}$ for metric TSP [Boyd-Carr '99]
- What is the integrality gap in our planted TSP?

Optimal recovery via fractional 2-factor LP

Theorem (Bagaria-Ding-Tse-Wu-X. '18)

If

$$\mu^2 - 4\log n \to +\infty,$$

then $\min_{x^*} \mathbb{P}\left\{\widehat{x}_{\mathrm{F2F}} = x^*\right\} \to 1.$

Optimal recovery via fractional 2-factor LP

Theorem (Bagaria-Ding-Tse-Wu-X. '18)

If

$$\mu^2 - 4\log n \to +\infty,$$

then $\min_{x^*} \mathbb{P}\left\{\widehat{x}_{\text{F2F}} = x^*\right\} \to 1.$

Remarks:

- Achieving the IT-limit $\mu^2 = 4\log n$
- When above IT-limit, the integrality gap is 1 whp!

General distributions P_n and Q_n

Threshold determined by Battacharyya distance (a.k.a. Rényi divergence of order $\frac{1}{2}$):

$$B(P, Q) \triangleq -2 \log \int \sqrt{\mathrm{d}P \mathrm{d}Q}$$

General distributions P_n and Q_n

Threshold determined by Battacharyya distance (a.k.a. Rényi divergence of order $\frac{1}{2}$):

$$B(P,Q) \triangleq -2\log \int \sqrt{\mathrm{d}P\mathrm{d}Q}$$

Theorem (Bagaria-Ding-Tse-Wu-X. '18)

If

$$B(P,Q) - \log n \to +\infty,$$

then $\min_{x^*} \mathbb{P}\left\{\widehat{x}_{\mathrm{F2F}} = x^*\right\} \to 1.$

General distributions P_n and Q_n

Threshold determined by Battacharyya distance (a.k.a. Rényi divergence of order $\frac{1}{2}$):

$$B(P,Q) \triangleq -2\log \int \sqrt{\mathrm{d}P\mathrm{d}Q}$$

Theorem (Bagaria-Ding-Tse-Wu-X. '18)

If

$$B(P,Q) - \log n \to +\infty,$$

then $\min_{x^*} \mathbb{P}\left\{\widehat{x}_{\mathrm{F2F}} = x^*\right\} \to 1.$

Remarks

- $B(P,Q) \geq (1+o(1))\log n$ is necessary for any estimator to succeed
- F2F achieves the optimal recovery threshold:

$$\liminf_{n\to\infty} \frac{B(P,Q)}{\log n} = 1.$$

Proof attempt via dual certificate argument

- KKT conditions (Farkas' lemma): $\widehat{x}_{\mathrm{F2F}} = x^* \iff \exists u \in \mathbb{R}^n$ (dual certificate):

$$\begin{aligned} u_i + u_j &\le w_{ij}, \qquad \text{if } x^*_{ij} = 1 \\ u_i + u_j &\ge w_{ij}, \qquad \text{if } x^*_{ij} = 0 \end{aligned}$$

- One feasible choice of dual:

$$u_i = \frac{1}{2} \min_j \left\{ w_{ij} : x^*_{ij} = 1 \right\}$$

- This certificate shows correctness if $\mu^2 > 6 \log n$ (same as greedy merging), unable to get to IT limit $\mu^2 > 4 \log n$!

Our proof based on primal analysis

General recipe: show whp for all extremal points $x \neq x^*$ of

$$\text{F2F polytope } \triangleq \left\{ x \in [0,1]^{\binom{n}{2}} : x(\delta(v)) = 2, \forall v \in [n] \right\},$$

it holds that

$$\langle w, x - x^* \rangle < 0$$

Our proof based on primal analysis

General recipe: show whp for all extremal points $x \neq x^*$ of

$$\text{F2F polytope } \triangleq \left\{x \in [0,1]^{\binom{n}{2}} : x(\delta(v)) = 2, \forall v \in [n]\right\},$$

it holds that

$$\langle w, x - x^* \rangle < 0$$

The proof heavily exploits the characterization of extremal points

- F2F polytope is not integral: fractional extremal points exist
- Half integrality [Balinski '65]: for any extremal point x,

$$x_e \in \{0, 1/2, 1\}$$

Proof outline

1. Encode the perturbation: for any extremal point x, represent $2(x - x^*)$ as a bicolored multigraph G_x with

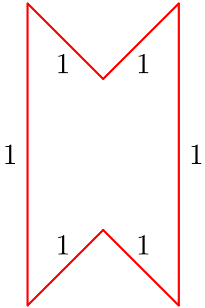
$$w(G_x) = \sum_e w_e (x_e - x_e^*)$$

2. Divide and conquer: decompose G_x as a union of graphs in family $\mathcal{F}$

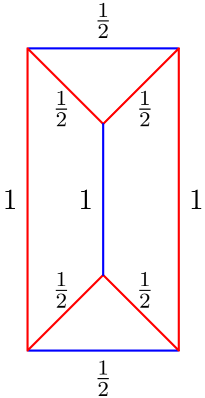
$$w(G_x) = \sum_i w(F_i), \quad F_i \in \mathcal{F}$$

3. Counting and large dev. bounds: show whp $w(F) < 0$ for all $F \in \mathcal{F}$

Step 1: Bicolored multigraph representation

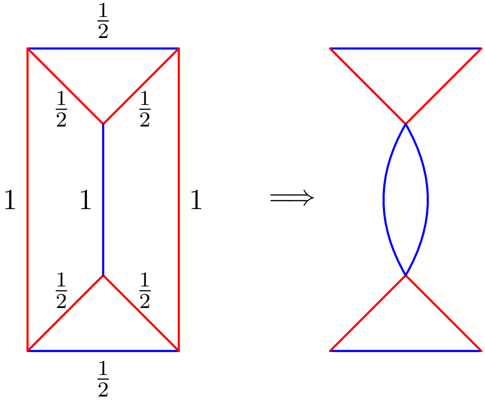


X^*: true cycle



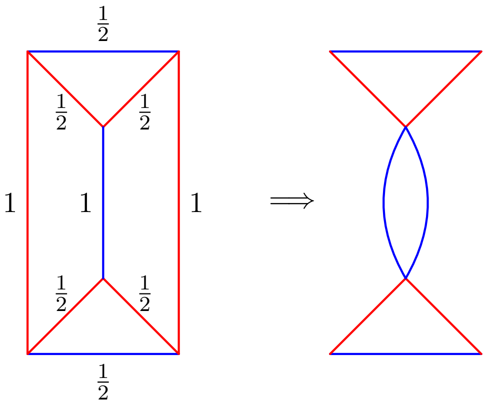
X: extremal solution

Step 1: Bicolored multigraph representation



X: extremal solution $\quad\quad$ G_X

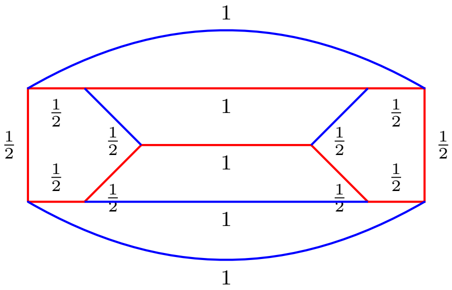
Step 1: Bicolored multigraph representation



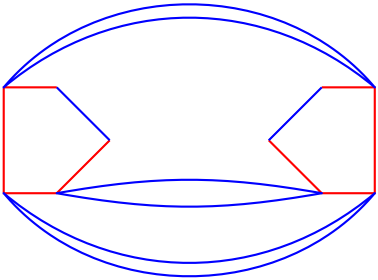
X: extremal solution $\qquad$ G_X

key observation

G_X is always balanced: red degree $=$ blue degree



$\Downarrow$



Step 2: Edge decomposition

Theorem (Kotzig '68)

Every connected balanced bicolored multigraph has an alternating Eulerian circuit.

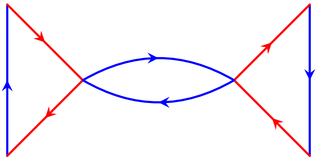
Step 2: Edge decomposition

Theorem (Kotzig '68)

Every connected balanced bicolored multigraph has an alternating Eulerian circuit.

Remarks

- An Eulerian circuit may traverse a double edge twice

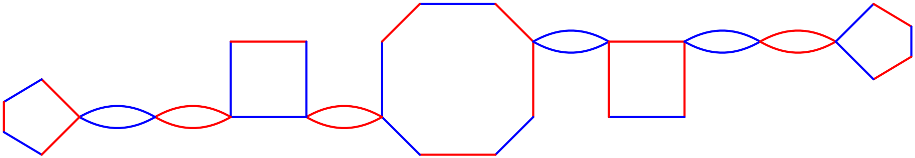


"Dumbbell" structure

Step 2: Edge decomposition

$\mathcal{U}$: collection of graphs recursively constructed

1. Start with an even cycle in alternating colors
2. **Blossoming procedure**: At each step, contract an edge in any cycle and attach a **flower** (path of double edges followed by an alternating odd cycle)

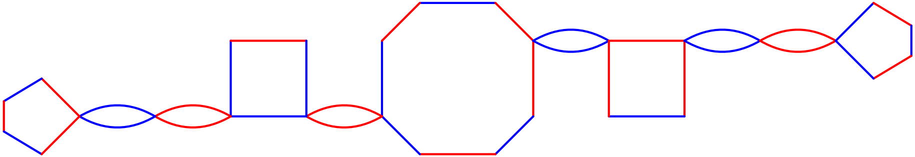


Obtained by starting with an 10-cycle and blossoming 4 times

Step 2: Edge decomposition

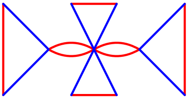
$\mathcal{U}$: collection of graphs recursively constructed

1. Start with an even cycle in alternating colors
2. **Blossoming procedure**: At each step, contract an edge in any cycle and attach a **flower** (path of double edges followed by an alternating odd cycle)



Obtained by starting with an 10-cycle and blossoming 4 times

However, not every G_X is of this form...

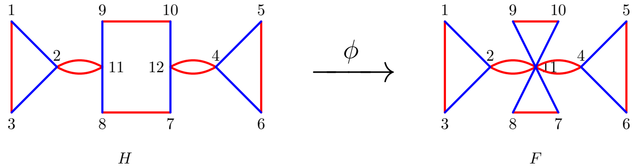


- Graph homomorphism $\phi : H \to F$ is a vertex map that preserves edges

$$\mathcal{F} = \{F : \text{there exists } H \in \mathcal{U} \text{ such that } H \to F\}$$

Lemma (Decomposition)

Every balanced bicolored multigraph G with edge multiplicity at most 2 can be decomposed as an union of elements in $\mathcal{F}$.

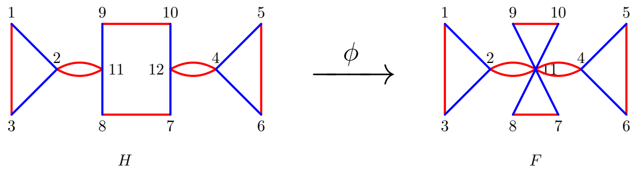
- 

- Graph homomorphism $\phi : H \to F$ is a vertex map that preserves edges

$$\mathcal{F} = \{F : \text{there exists } H \in \mathcal{U} \text{ such that } H \to F\}$$

Lemma (Decomposition)

Every balanced bicolored multigraph G with edge multiplicity at most 2 can be decomposed as an union of elements in $\mathcal{F}$.

- 

- remains to show $\min_{F \in \mathcal{F}} w(F) < 0$ whp

Step 3: Counting and probabilistic arguments

$$\mathcal{F}_{k,\ell} = \{F \in \mathcal{F} : E(F) \text{ consists of } k \text{ double edges and } \ell \text{ single edges } \}$$

Lemma

For any $k \geq 0$ and $\ell \geq 3$. With probability at least $1 - n^{-\Theta(k+\ell)}$,

$$\max_{F \in \mathcal{F}_{k,\ell}} \left(w(F) - \mathbb{E}\left[w(F)\right]\right) \leq (1+\epsilon)\,(2k+\ell)\sqrt{\log n}$$

Step 3: Counting and probabilistic arguments

$\mathcal{F}_{k,\ell} = \{F \in \mathcal{F} : E(F)$ consists of k double edges and ℓ single edges $\}$

Lemma

For any $k \geq 0$ and $\ell \geq 3$. With probability at least $1 - n^{-\Theta(k+\ell)}$,

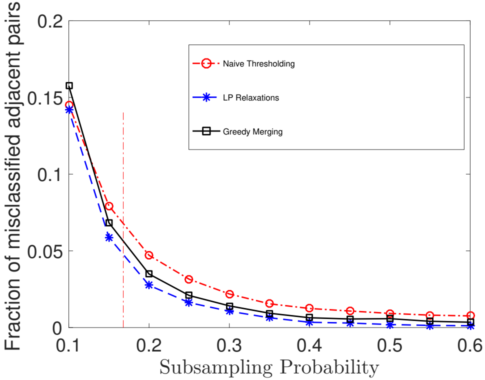
$$\max_{F \in \mathcal{F}_{k,\ell}} (w(F) - \mathbb{E}[w(F)]) \leq (1+\epsilon)\,(2k+\ell)\,\sqrt{\log n}$$

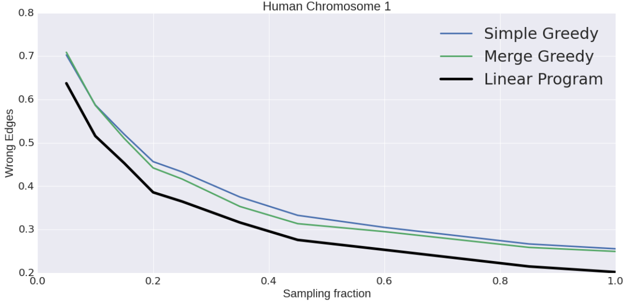
Remarks

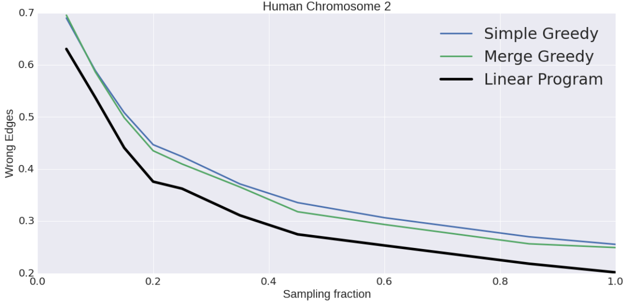
- Total: $2k + \ell$ edges, half are red (by balancedness). Weights on red edges: $N(\mu, 1)$
$$\mathbb{E}[w(F)] = -(2k+\ell)\mu/2$$
- Proof: Counting $\mathcal{F}_{k,\ell}$ and large deviation bounds
- Key observation for counting: condition on one end of a red edge, the other end has at most 2 choices

Real data experiment

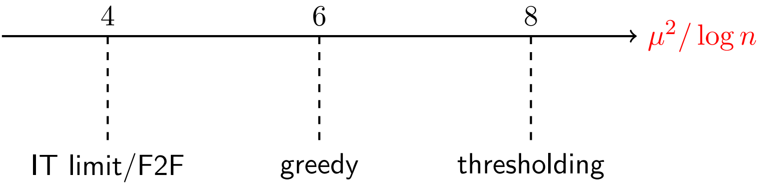
- 1000 DNA contigs of size 45 kb
- 0.45 million Chicago cross-links
- Subsample each cross-link with probability p



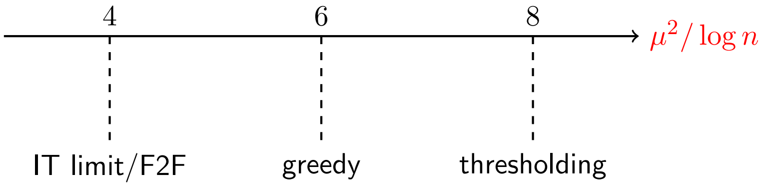

Human Chromosome 1
Simple Greedy
Merge Greedy
Linear Program
Wrong Edges
Sampling fraction
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.0
0.2
0.4
0.6
0.8
1.0


Human Chromosome 2
Simple Greedy
Merge Greedy
Linear Program
Wrong Edges
Sampling fraction
0.2
0.3
0.4
0.5
0.6
0.7
0.0
0.2
0.4
0.6
0.8
1.0

Conclusion and remarks

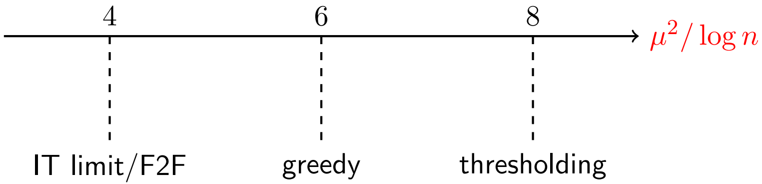


Conclusion and remarks



- Curse of high-dim. $\Longrightarrow$ MLE is computationally intractable
- Bless of high-dim. $\Longrightarrow$ convex relaxations are statistically optimal

Conclusion and remarks



- Curse of high-dim. $\Longrightarrow$ MLE is computationally intractable
- Bless of high-dim. $\Longrightarrow$ convex relaxations are statistically optimal

Connections to other work

- graph partioning (community detection): [Hajek-Wu-Xu '14]
- graph isomorphism (network alignment)