

Finding the Way for Graph Matching

Daniel Sussman

Boston University
Department of Mathematics and Statistics

April 24, 2019

Supported by
DARPA and MIT Lincoln Labs HDDN Program

The Team



Vince Lyzinski
(UMass Amherst/Univ. MD)

- Jesus Arroyo, Donniel Fishkind, Carey Priebe, Youngser Park (JHU)
- Zihuan Qiao, Fei Fang, Esther Wang (BU)
- Lin Li, Charlie Dagli (MITLL)
- ... many others

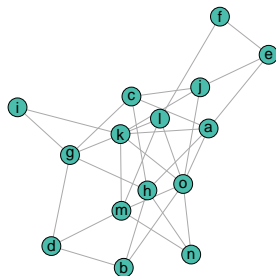
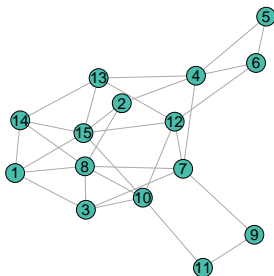
What is graph matching?

Formulation

Consider observing two graphs, $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$.

The classical graph matching formulation is to find a map $\pi : V_1 \mapsto V_2$, that minimizes the symmetric difference between

$$\pi(E_1) = \{(\pi(i), \pi(j)) : (i, j) \in E_1\} \text{ and } E_2.$$



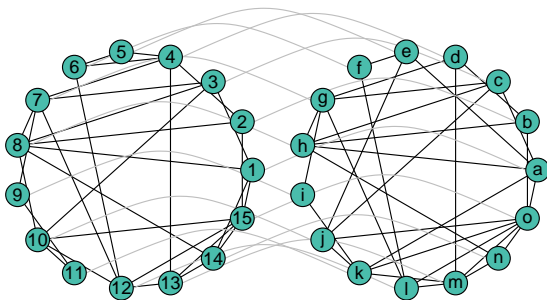
What is graph matching?

Formulation

Consider observing two graphs, $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$.

The classical graph matching formulation is to find a map $\pi : V_1 \mapsto V_2$, that minimizes the symmetric difference between

$$\pi(E_1) = \{(\pi(i), \pi(j)) : (i, j) \in E_1\} \text{ and } E_2.$$



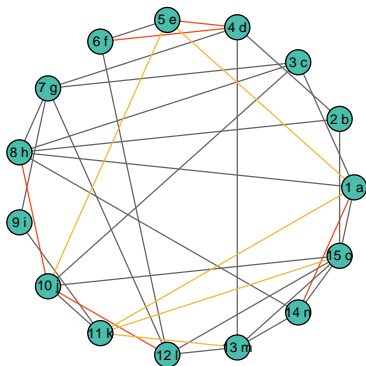
What is graph matching?

Formulation

Consider observing two graphs, $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$.

The classical graph matching formulation is to find a map $\pi : V_1 \mapsto V_2$, that minimizes the symmetric difference between

$$\pi(E_1) = \{(\pi(i), \pi(j)) : (i, j) \in E_1\} \text{ and } E_2.$$



Matrix Formulation

Adjacency Matrix Formulation

Let $A \in \{0,1\}^{n_A \times n_A}$ and $B \in \{0,1\}^{n_B \times n_B}$, with $n_A < n_B$

The graph matching problem is to find

$$\operatorname{argmin}_{P \in \mathcal{P}^{n_A, n_B}} \left\| A - PB P^T \right\|_F^2.$$

i.e. minimize number of edge discrepancies, between A and PBP^T .

Easily extends to weighted graphs, resulting in the QAP.

Sub-permutation Matrices

For $m, n \in \mathbb{N}$, $m < n$, let $\mathcal{P}_{m,n}$ be the set of binary $m \times n$ matrices

- with exactly one non-zero entry per row and
- at most one non-zero entry per column.

When $m = n$ this is the set of permutation matrices.

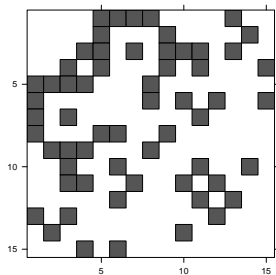
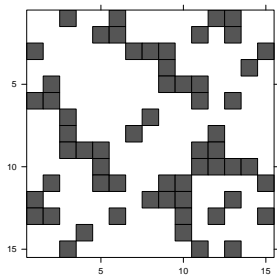
Matrix Formulation

Adjacency Matrix Formulation

Let $A \in \{0, 1\}^{n_A \times n_A}$ and $B \in \{0, 1\}^{n_B \times n_B}$, with $n_A < n_B$

The graph matching problem is to find

$$\operatorname{argmin}_{P \in \mathcal{P}_{n_A, n_B}} \left\| A - PBP^T \right\|_F^2.$$



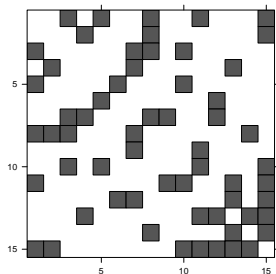
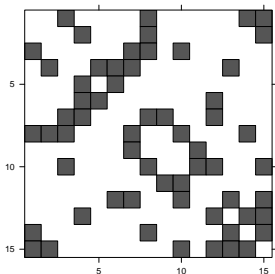
Matrix Formulation

Adjacency Matrix Formulation

Let $A \in \{0, 1\}^{n_A \times n_A}$ and $B \in \{0, 1\}^{n_B \times n_B}$, with $n_A < n_B$

The graph matching problem is to find

$$\operatorname{argmin}_{P \in \mathcal{P}_{n_A, n_B}} \left\| A - PBP^T \right\|_F^2.$$



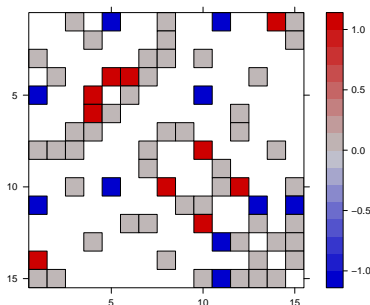
Matrix Formulation

Adjacency Matrix Formulation

Let $A \in \{0, 1\}^{n_A \times n_A}$ and $B \in \{0, 1\}^{n_B \times n_B}$, with $n_A < n_B$

The graph matching problem is to find

$$\operatorname{argmin}_{P \in \mathcal{P}_{n_A, n_B}} \left\| A - PBP^T \right\|_F^2.$$



How do you solve it?

It's really hard!

QAP is a notoriously difficult NP-complete problem.

Many approaches

- 30 Years of Graph Matching (Conte et al., 2004)
- 50 Years of Graph Matching (Emmert-Streib et al., 2016)
- Relaxation and optimization methods (Lyzinski et al., 2016; El-Kebir et al., 2015)
- Seeds and prior information are critical (Zhang and Tong, 2016; Lyzinski et al., 2013)

Equivalent Matrix Formulations

Binary Directed Graphs

$$\begin{aligned} \left\| A - PBP^T \right\|_F^2 &= \text{tr}(A^T A) && \# \text{ edges in } A \\ &\quad - 2\text{tr}(A^T PBP^T) && 2 \times \# \text{ common edges} \\ &\quad + \text{tr}(PB^T P^T PBP^T) && \# \text{ edges in } PBP^T \end{aligned}$$

Equivalent Matrix Formulations

Binary Directed Graphs

$$\begin{aligned} \left\| A - PBP^T \right\|_F^2 &= \text{tr}(A^T A) && \# \text{ edges in } A \\ &\quad - 2\text{tr}(A^T PBP^T) && 2 \times \# \text{ common edges} \\ &\quad + \text{tr}(PB^T P^T PBP^T) && \# \text{ edges in } PBP^T \end{aligned}$$

Equal # Vertices and Undirected

GMP simplifies to

$$\operatorname{argmax}_{P \in \mathcal{P}_{n,n}} \text{tr}(APBP^T). \quad \# \text{ common edges}$$

- $n_A = n_B \implies P$ is a permutation \implies last line doesn't depend on P .
- undirected graph \implies middle line is $-2\text{tr}(APBP^T)$

We'll define this as the default and rebuild others.

Relaxation and Iterative Method

Relaxations and Gradient

Viewed as a function of $n \times n$ real matrices,

- the function $f(M) = \text{tr}(AMB M^T)$ has gradient $\text{vec}(AMB)$.
- Entries $(AMB)_{uv}$ give
 - ▶ a (weighted) count of nodes “currently matched” by M
 - ▶ that are adjacent to u in A and v in B .

Relaxation and Iterative Method

Relaxations and Gradient

Viewed as a function of $n \times n$ real matrices,

- the function $f(M) = \text{tr}(AMB M^T)$ has gradient $\text{vec}(AMB)$.
- Entries $(AMB)_{uv}$ give
 - ▶ a (weighted) count of nodes “currently matched” by M
 - ▶ that are adjacent to u in A and v in B .

Gradient Ascent (from D)

- within doubly stochastic matrices (convex hull of P):
Solve LAP, $\text{argmax}_{P \in \mathcal{P}} \text{tr}(ADBP)$
(Vogelstein et al., 2014)

Relaxation and Iterative Method

Relaxations and Gradient

Viewed as a function of $n \times n$ real matrices,

- the function $f(M) = \text{tr}(AMB M^T)$ has gradient $\text{vec}(AMB)$.
- Entries $(AMB)_{uv}$ give
 - ▶ a (weighted) count of nodes “currently matched” by M
 - ▶ that are adjacent to u in A and v in B .

Gradient Ascent (from D)

- within doubly stochastic matrices (convex hull of P):
Solve LAP, $\text{argmax}_{P \in \mathcal{P}} \text{tr}(ADBP)$
(Vogelstein et al., 2014)
- within sub-stochastic matrices:
Set $D_{uv} \rightarrow 1$ for where $(u, v) = \text{argmax}_{(i,j)} (ADB)_{ij}$
(Yartseva and Grossglauser, 2013).

Multiplex Subgraph Matching

Goal:

Find 35-node template graph

World Graph

- FriendFeed: 5,540 nodes and 31,921 edges
- Twitter: 5,702 nodes and 42,327 edges
- YouTube: 663 nodes and 614 edges

574 nodes span all three layers.

Collected by M. Magnani and L. Rossi and provided via PNNL.



Note: Multiplex matching achieved via summing objective over layers/channels.

Finding a Solution

Getting Started

- 1 Use node-filtering to reduce possible matches for each node (Moorman et al., 2018). Eg.
 - ▶ degree in world \geq degree in subgraph;
 - ▶ edge in template must appear between candidate nodes, etc

20 nodes have 1 candidate,
3 have 2, rest 35–1968.
- 2 Perform LAP+line-search where
 - ▶ constrain that $D_{ij} \neq 0$ only if j is a candidate for i .
 - ▶ start at stochastic matrix D^0 with $D_{ij}^0 = 1/n_i$ if j is candidate for i and n_i is # candidates for node i .

Finding a Solution

Getting Started

- 1 Use node-filtering to reduce possible matches for each node (Moorman et al., 2018). Eg.
 - ▶ degree in world \geq degree in subgraph;
 - ▶ edge in template must appear between candidate nodes, etc20 nodes have 1 candidate, 3 have 2, rest 35–1968.
- 2 Perform LAP+line-search where
 - ▶ constrain that $D_{ij} \neq 0$ only if j is a candidate for i .
 - ▶ start at stochastic matrix D^0 with $D_{ij}^0 = 1/n_i$ if j is candidate for i and n_i is # candidates for node i .



layer	Edges		
	Missing	Extra	Common
FFeed	0	43	60
Twitter	1	65	66
Youtube	0	0	31

Alternative Rewards

Using objective

$$\sum_{c \in F, T, Y} \text{tr}(A_c D B_c D^T)$$

Template	Rewards	
	Edge	No-Edge
Edge	1	0
No-Edge	0	0



layer	Edges		
	Missing	Extra	Common
FFeed	0	43	60
Twitter	1	65	66
Youtube	0	0	31

Alternative Rewards

Using objective

$$\sum_{c \in F, T, Y} \text{tr}(A_c D B_c D^T) - \text{tr}(A_c D \bar{B}_c D^T)$$

where \bar{B}_c denotes the graph complement.

Template	Rewards	
	Edge	No-Edge
Edge	1	-1
No-Edge	0	0



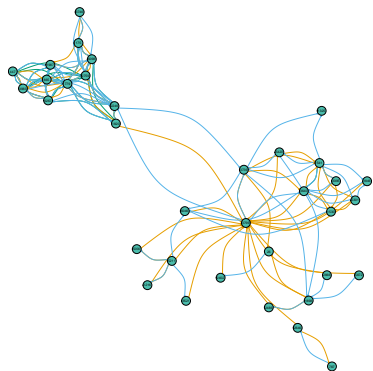
layer	Edges		
	Missing	Extra	Common
FFeed	0	45	60
Twitter	0	73	67
Youtube	0	0	31

Alternative Rewards

Using objective

$$\begin{aligned} & \sum_{c \in F, T, Y} \text{tr}(A_c D B_c D^T) \\ & - \text{tr}(A_c D \bar{B}_c D^T) \\ & - 0.25 \text{tr}(\bar{A}_c D B_c D^T) \\ & + 0.25 \text{tr}(\bar{A}_c D \bar{B}_c D^T) \end{aligned}$$

Template	Rewards	
	Edge	No-Edge
Edge	1	-1
No-Edge	-0.25	0.25



layer	Edges		
	Missing	Extra	Common
FFeed	2	0	58
Twitter	2	5	65
Youtube	0	0	31

Finding the “Truth”

Subgraph Finding

- UCLA Team found that there exist 10^{14} exact subgraph mappings which result in 0 missing edges.

Finding the “Truth”

Subgraph Finding

- UCLA Team found that there exist 10^{14} exact subgraph mappings which result in 0 missing edges.

Subgraph Isomorphism

- By randomizing the start matrix matrix,
 - ▶ we ran 4000 restarts of our procedure
 - ▶ with the last objective.
 - ▶ Each restart takes $\approx 6 - 10s$.
- Two best were distinct exact subgraph isomorphisms.
- Neither of these correspond with nodes originally picked by PNNL.

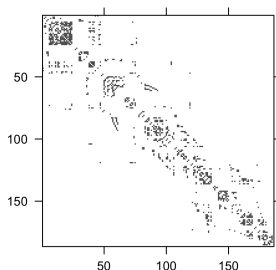
Which is/are the truth?

DTMRI Brain Example

Data (Sussman et al., 2018), neurodata.io

- 40k nodes divided into 2 hemispheres with 35 matched regions each.
- Want to match
 - ▶ 186-node graph for region 7 in the left hemisphere
 - ▶ to 20k-node graph for entire right hemisphere.

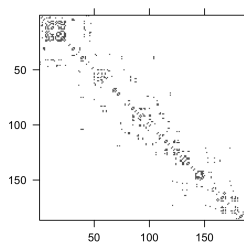
region 7 hemisphere left



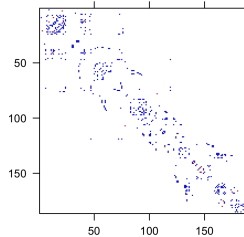
nodes: 186, # edges: 611

DTMRI Brain Example

most matches

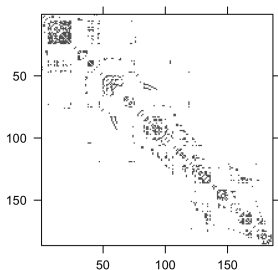


matches: 56



edge errors: 285

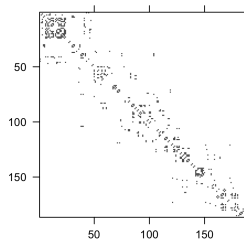
region 7 hemisphere left



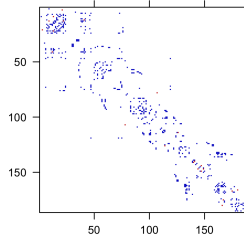
nodes: 186, # edges: 611

DTMRI Brain Example

most matches

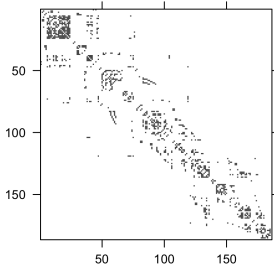


matches: 56



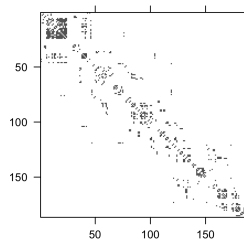
edge errors: 285

region 7 hemisphere left

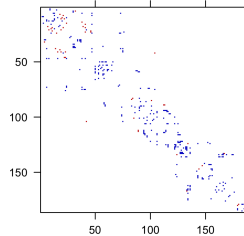


nodes: 186, # edges: 611

fewest edge errors



matches: 9



edge errors: 176

Heterogeneity and Centering

Correlated Erdős-Rényi Model

For R , Λ_A and Λ_B symmetric, hollow matrices in $[0, 1]^{n \times n}$, we say $A, B \sim \text{CorrER}(\Lambda_A, \Lambda_B, R)$ if:

- i. A (resp. B) are marginally $\text{ER}(\Lambda_A)$ (resp. $\text{ER}(\Lambda_B)$); i.e.
- ▶ for all $u, v \in [n]$, $u < v$,

$$A_{uv} \stackrel{\text{ind}}{\sim} \text{Bern}(\Lambda_{A,uv}) \text{ and } B_{uv} \stackrel{\text{ind}}{\sim} \text{Bern}(\Lambda_{B,uv}),$$

- ▶ with $A_{uv} = A_{vu}$ and $B_{uv} = B_{vu}$.

- ii. For all $u, v, w, r \in [n]$, $u < v$, $w < r$, it holds that

- ▶ A_{uv} and B_{wr} are independent unless $(u, v) = (w, r)$,
- ▶ in which case the correlation between A_{uv} and B_{uv} is $R_{u,v} \geq 0$.

The true vertex correspondence in the identity mapping.

Heterogeneity and Centering

Correlated Erdős-Rényi Model

- $A_{uv} \stackrel{ind}{\sim} \text{Bern}(\Lambda_{A,uv})$ and $B_{uv} \stackrel{ind}{\sim} \text{Bern}(\Lambda_{B,uv})$,
- correlation between A_{uv} and B_{uv} is $R_{u,v} \geq 0$.

Matchability (Lyzinski and Sussman, 2017)

- When Λ_A and Λ_B have matching structure, the matching is easier.

Heterogeneity and Centering

Correlated Erdős-Rényi Model

- $A_{uv} \stackrel{ind}{\sim} \text{Bern}(\Lambda_{A,uv})$ and $B_{uv} \stackrel{ind}{\sim} \text{Bern}(\Lambda_{B,uv})$,
- correlation between A_{uv} and B_{uv} is $R_{u,v} \geq 0$.

Matchability (Lyzinski and Sussman, 2017)

- When Λ_A and Λ_B have matching structure, the matching is easier.
- When Λ_A and Λ_B have different structures,
 - ▶ standard graph matching cannot find the truth because
 - ★ matching similarities between Λ_A and Λ_B
 - ★ will overwhelm the correlation R .

Heterogeneity and Centering

Correlated Erdős-Rényi Model

- $A_{uv} \stackrel{ind}{\sim} \text{Bern}(\Lambda_{A,uv})$ and $B_{uv} \stackrel{ind}{\sim} \text{Bern}(\Lambda_{B,uv})$,
- correlation between A_{uv} and B_{uv} is $R_{u,v} \geq 0$.

Matchability (Lyzinski and Sussman, 2017)

- When Λ_A and Λ_B have matching structure, the matching is easier.
- When Λ_A and Λ_B have different structures,
 - ▶ standard graph matching cannot find the truth because
 - ★ matching similarities between Λ_A and Λ_B
 - ★ will overwhelm the correlation R .
 - ▶ Instead, maximize empirical covariance
 $\text{argmax}_{P \in \mathcal{P}} \text{tr}((A - \Lambda_A)P(B - \Lambda_B)P^T)$.
 - ▶ In practice, would need to estimate Λ_A and Λ_B ,
eg. by singular value thresholding, fit an SBM, etc.

Heterogeneity and Centering

Correlated Erdős-Rényi Model

- $A_{uv} \stackrel{ind}{\sim} \text{Bern}(\Lambda_{A,uv})$ and $B_{uv} \stackrel{ind}{\sim} \text{Bern}(\Lambda_{B,uv})$,
- correlation between A_{uv} and B_{uv} is $R_{u,v} \geq 0$.

Matchability (Lyzinski and Sussman, 2017)

- When Λ_A and Λ_B have matching structure, the matching is easier.
- When Λ_A and Λ_B have different structures,
 - ▶ standard graph matching cannot find the truth because
 - ★ matching similarities between Λ_A and Λ_B
 - ★ will overwhelm the correlation R .
 - ▶ Instead, maximize empirical covariance
 $\text{argmax}_{P \in \mathcal{P}} \text{tr}((A - \Lambda_A)P(B - \Lambda_B)P^T)$.
 - ▶ In practice, would need to estimate Λ_A and Λ_B ,
eg. by singular value thresholding, fit an SBM, etc.

*Difficult balance between exploiting matching structure
and avoiding spurious commonalities.*

Matching FriendFeed and Twitter

True matching

4854 nodes with edges in both graphs.

- 9.3k edges in common,
- 22k extra edges in FriendFeed, and
- 33k extra edges in Twitter.

Matching FriendFeed and Twitter

True matching

4854 nodes with edges in both graphs.

- 9.3k edges in common,
- 22k extra edges in FriendFeed, and
- 33k extra edges in Twitter.

Using standard graph matching, starting at the true correspondence, we find a match with

- 12.5k edges in common,
- 19k extra edges in FriendFeed, and
- 29k extra edges in Twitter.

2092 node matches at true correspondence.

Matching FriendFeed and Twitter

True matching

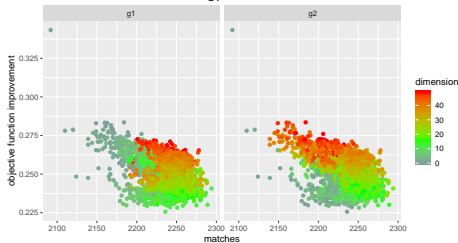
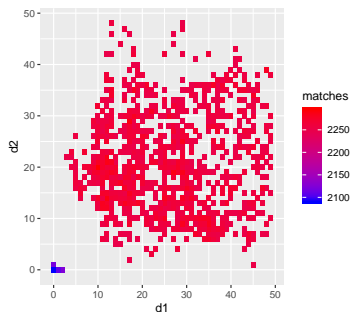
4854 nodes with edges in both graphs.

- 9.3k edges in common,
- 22k extra edges in FriendFeed, and
- 33k extra edges in Twitter.

Using standard graph matching, starting at the true correspondence, we find a match with

- 12.5k edges in common,
- 19k extra edges in FriendFeed, and
- 29k extra edges in Twitter.

2092 node matches at true correspondence.



Looking Forward

Prior information is critical

- Leveraging partially known needed for algorithmic tractability.
- Even more crucial to learn what a match means.
- Need techniques to estimate joint graph distributions and propagate matched patterns to unmatched vertices.
- What about negative correlations?

Finding Subgraphs

- Filtering methods work well for subgraph detection but need refinement for for noisy cases.
- Knowing which edges/non-edges need to be present and which are more fuzzy/noisy can significantly impact performance.

Please use our R package iGraphMatch in development at <https://github.com/dpmcsuss/iGraphMatch/tree/dev>

Thanks!

Supported by
DARPA and MIT Lincoln Labs HDDN Program

References

- Athreya, A., Fishkind, D. E., Levin, K., Lyzinski, V., Park, Y., Qin, Y., Sussman, D. L., Tang, M., Vogelstein, J. T., and Priebe, C. E. (2017). Statistical inference on random dot product graphs: a survey.
- Conte, D., Foggia, P., Sansone, C., and Vento, M. (2004). Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(03):265–298.
- El-Kebir, M., Heringa, J., and Klau, G. (2015). Natalie 2.0: Sparse global network alignment as a special case of quadratic assignment. *Algorithms*, 8(4):1035–1051.
- Emmert-Streib, F., Dehmer, M., and Shi, Y. (2016). Fifty years of graph matching, network alignment and network comparison. *Information sciences*, 346–347:180–197.
- Lyzinski, V., Fishkind, D. E., Fiori, M., Vogelstein, J. T., Priebe, C. E., and Sapiro, G. (2016). Graph matching: Relax at your own risk. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):60–73.
- Lyzinski, V. and Sussman, D. L. (2017). Matchability of heterogeneous networks pairs.
- Lyzinski, V., Sussman, D. L., Fishkind, D. E., Pao, H., and Priebe, C. E. (2013). Seeded graph matching for large stochastic block model graphs. *arXiv preprint arXiv:1310.1297*.
- Moorman, J. D., Chen, Q., Tu, T. K., Boyd, Z. M., and Bertozzi, A. L. (2018). Filtering methods for subgraph matching on multiplex networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 3980–3985.
- Sussman, D. L., Lyzinski, V., Park, Y., and Priebe, C. E. (2018). Matched filters for noisy induced subgraph detection.
- Tang, R., Ketcha, M., Vogelstein, J. T., Priebe, C. E., and Sussman, D. L. (2016). Law of large graphs.
- Vogelstein, J. T., Conroy, J. M., Lyzinski, V., Podrazik, L. J., Kratzer, S. G., Harley, E. T., Fishkind, D. E., Vogelstein, R. J., and Priebe, C. E. (2014). Fast Approximate Quadratic Programming for Graph Matching. *PLoS one*, 10(04).
- Yartseva, L. and Grossglauser, M. (2013). On the performance of percolation graph matching. In *Proceedings of the First ACM Conference on Online Social Networks, COSN '13*, pages 119–130, New York, NY, USA. ACM.
- Zhang, S. and Tong, H. (2016). Final: Fast attributed network alignment. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1345–1354. ACM.