Iterative Methods for Network Alignment

David F. Gleich
Computer Science
Purdue University

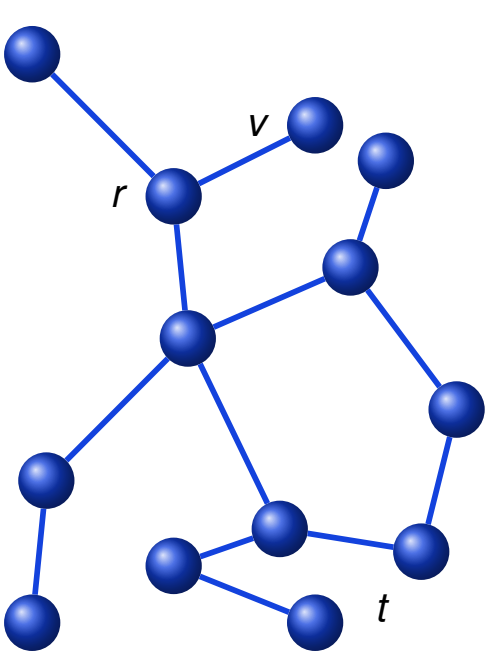
with

Arif Khan, Alex Pothen
Purdue University, Computer Science
Mahantesh Halappanavar
Pacific Northwest National Labs
Mohsen Bayati, Amin Saberi
Stanford University
Ying Wang Google

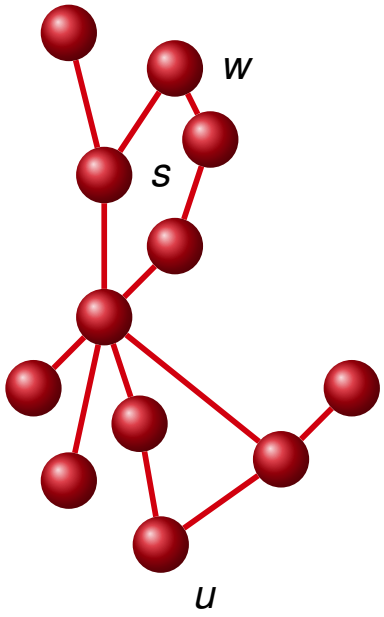
Work supported by DOE CSCAPES Institute grant (DE-FC02-08ER25864), NSF CAREER grant 1149756-CCF, and the Center for Adaptive Super Computing Software Multithreaded Architectures (CASS-MT) at PNNL. Stanford's CADS grant from the Library of Congress. PNNL is operated by Battelle Memorial Institute under contract DE-AC06-76RL01830

Network alignment

What is the best way of matching graph *A* to *B*?

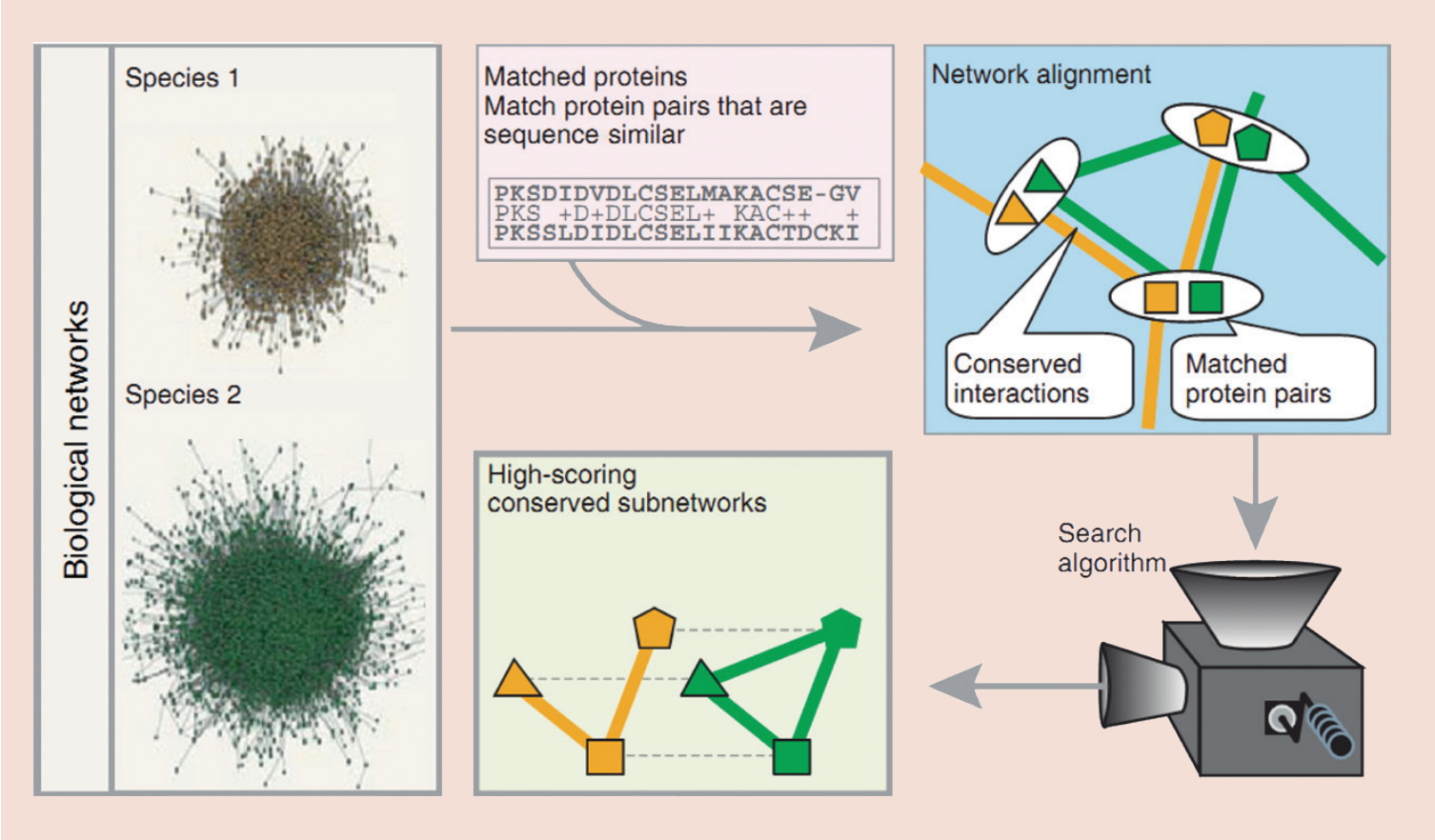


A



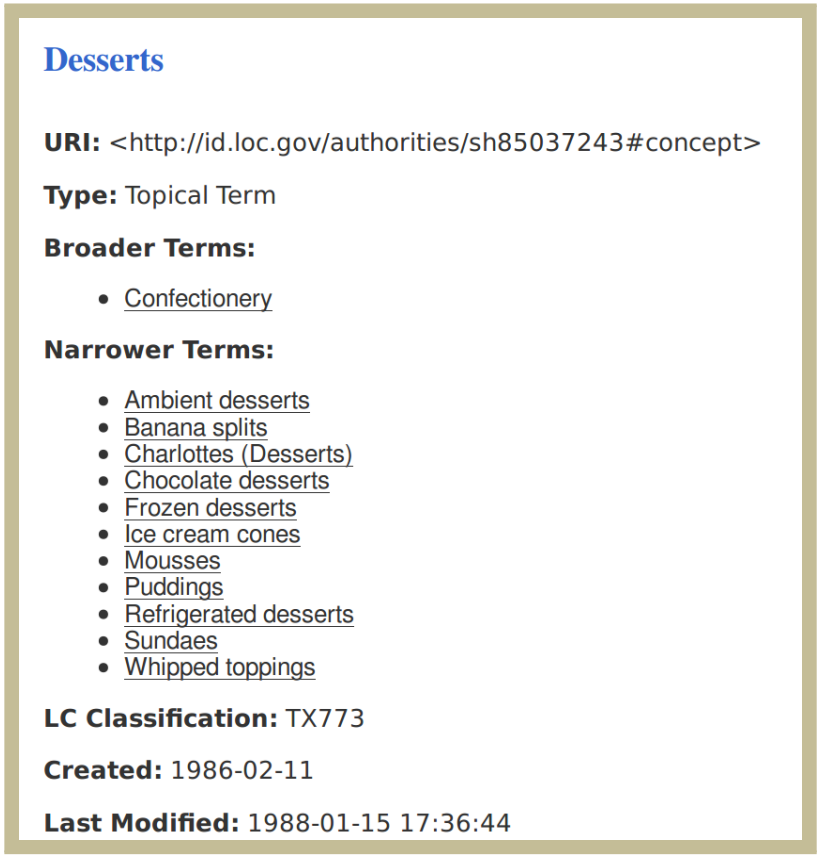
B

Network alignment



From Sharan and Ideker, Modeling cellular machinery through biological network comparison. *Nat. Biotechnol. 24,* 4 (Apr. 2006), 427–433.

Network alignment



Desserts

URI: <http://id.loc.gov/authorities/sh85037243#concept>

Type: Topical Term

Broader Terms:

- Confectionery

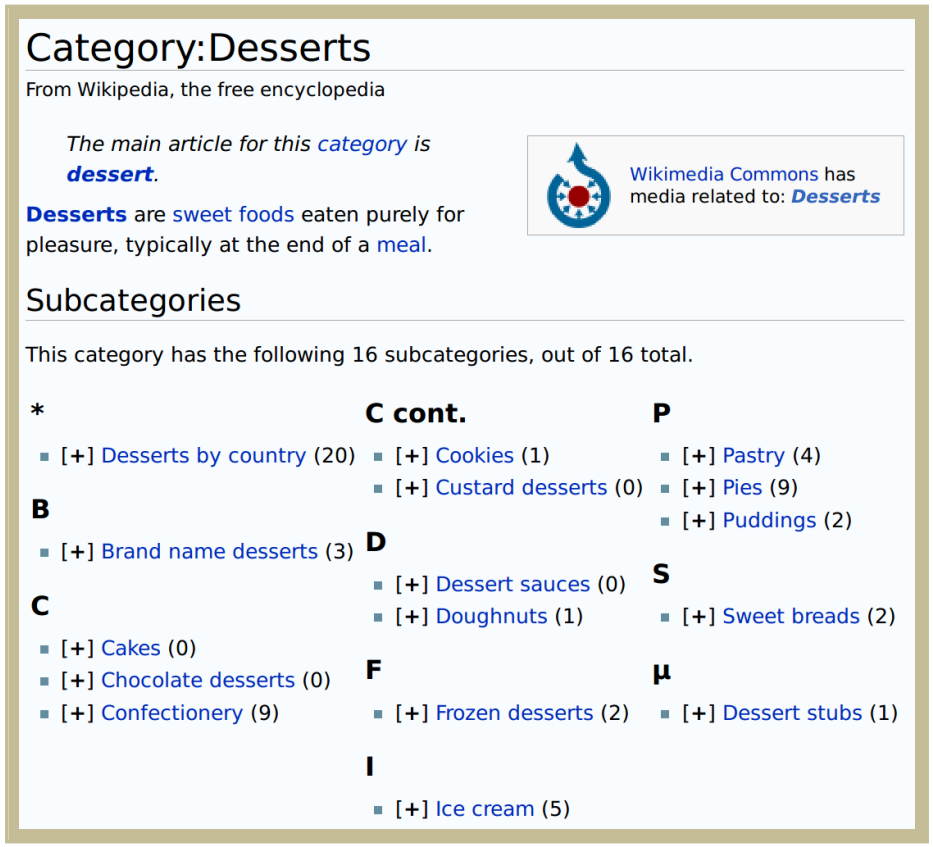
Narrower Terms:

- Ambient desserts
- Banana splits
- Charlottes (Desserts)
- Chocolate desserts
- Frozen desserts
- Ice cream cones
- Mousses
- Puddings
- Refrigerated desserts
- Sundaes
- Whipped toppings

LC Classification: TX773

Created: 1986-02-11

Last Modified: 1988-01-15 17:36:44



Category:Desserts

From Wikipedia, the free encyclopedia

*The main article for this category is **dessert**.*

Desserts are sweet foods eaten purely for pleasure, typically at the end of a meal.

Wikimedia Commons has media related to: ***Desserts***

Subcategories

This category has the following 16 subcategories, out of 16 total.

*****
- [+] Desserts by country (20)

B
- [+] Brand name desserts (3)

C
- [+] Cakes (0)
- [+] Chocolate desserts (0)
- [+] Confectionery (9)

C cont.
- [+] Cookies (1)
- [+] Custard desserts (0)

D
- [+] Dessert sauces (0)
- [+] Doughnuts (1)

F
- [+] Frozen desserts (2)

I
- [+] Ice cream (5)

P
- [+] Pastry (4)
- [+] Pies (9)
- [+] Puddings (2)

S
- [+] Sweet breads (2)

µ
- [+] Dessert stubs (1)

$\mathcal{A}$	LCSH	297,266 vertices, 248,230 edges
$\mathcal{B}$	Wikipedia	205,948 vertices, 382,353 edges
$\mathcal{L}$	links	4,971,629 edges

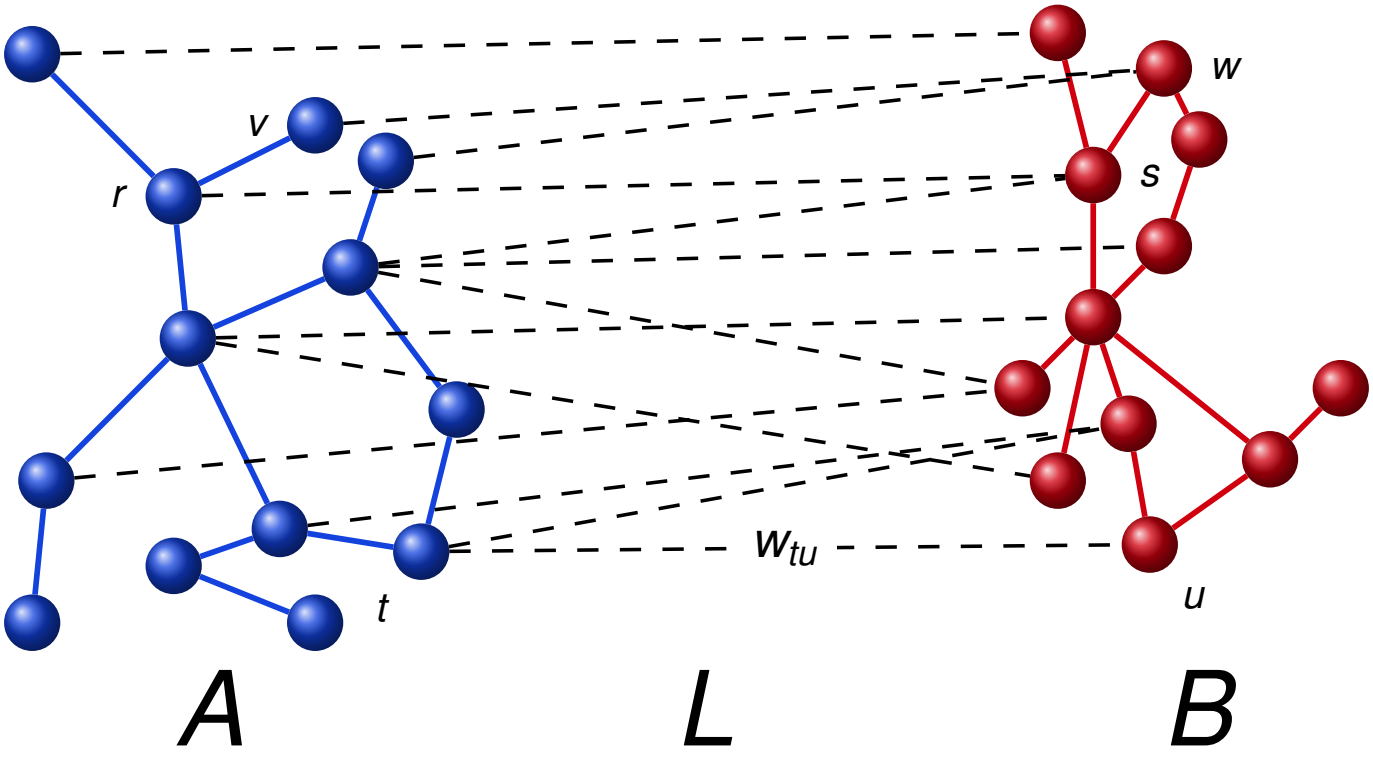
Sometimes small data becomes big …

Dataset	Size	Nonzeros
LCSH-2	59,849	227,464
WC-3	70,509	403,960
Product graph	4,219,893,141	91,886,357,440

… Ananth has some better techniques to work with these large problems …

Network alignment

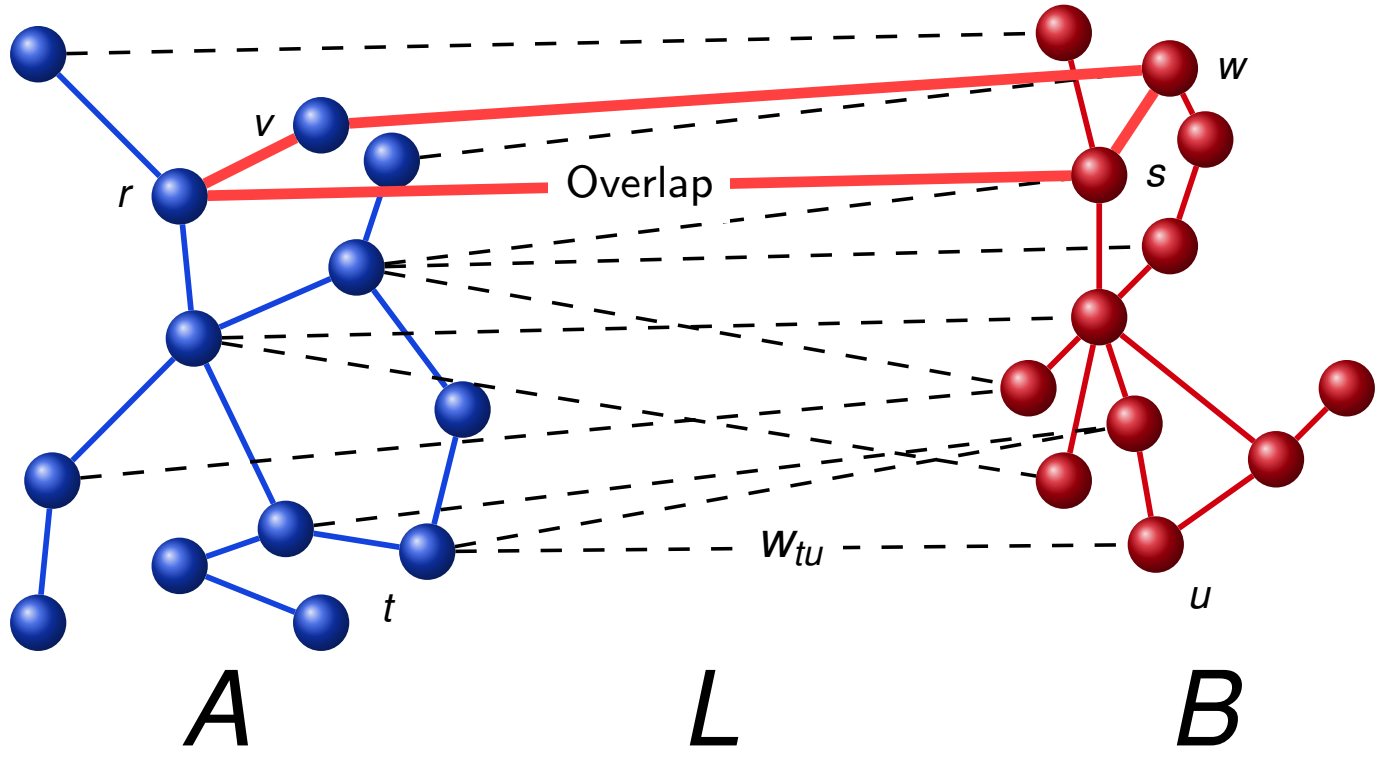
What is the best way of matching graph *A* to *B* using only edges in *L*?



Network alignment

Matching? 1-1 relationship

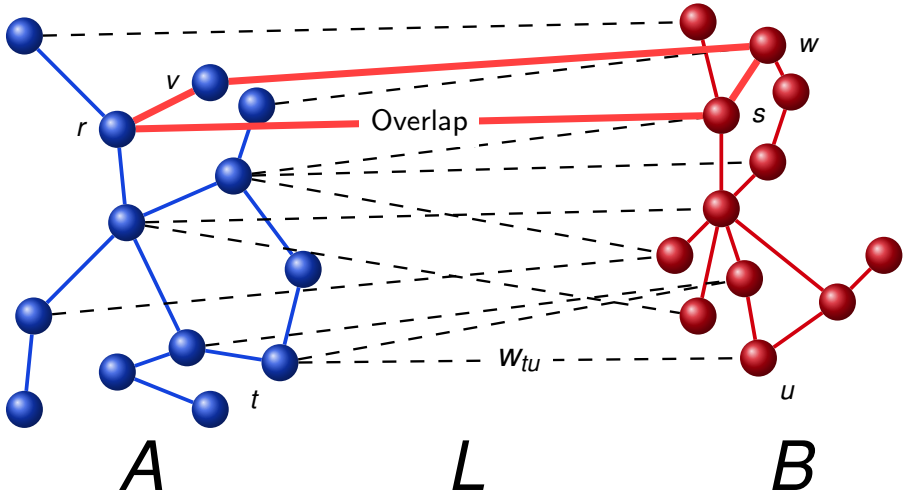
Best? highest weight and overlap



Network alignment

… is NP-hard

… has no approximation algorithm

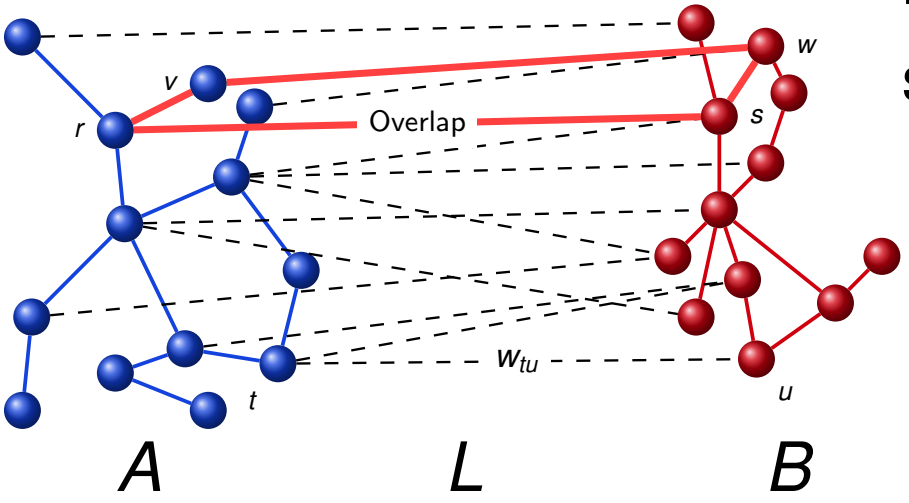


- Computer Vision
- Ontology matching
- Database matching
- Bioinformatics

objective = α matching + β overlap

Network alignment

via mathematical programming



maximize $\alpha \mathbf{w}^T \mathbf{x} + \frac{\beta}{2} \mathbf{x}^T \mathbf{S} \mathbf{x}$

subject to $\mathbf{A}\mathbf{x} \leq \mathbf{e}, x_i \in \{0, 1\}$

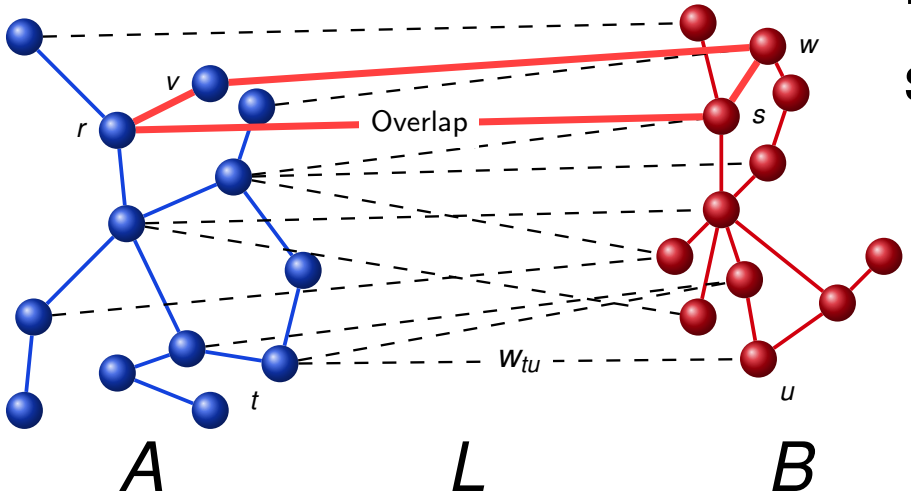
Let x_i be an indicator over edges in L

If A is the node-edge incidence matrix for L, then x is a 1-1 matching

Find a 1-1 matching between vertices with as many overlaps as possible.

Network alignment

via mathematical programming



maximize $\alpha \mathbf{w}^T \mathbf{x} + \frac{\beta}{2} \mathbf{x}^T \mathbf{S} \mathbf{x}$

subject to $\mathbf{A}\mathbf{x} \leq \mathbf{e}, x_i \in \{0, 1\}$

Let x_i be an indicator over edges in L

Let $S_{ij} = 1$ when x_i and x_j overlap, then $x^T S x$ is twice the overlapped count.

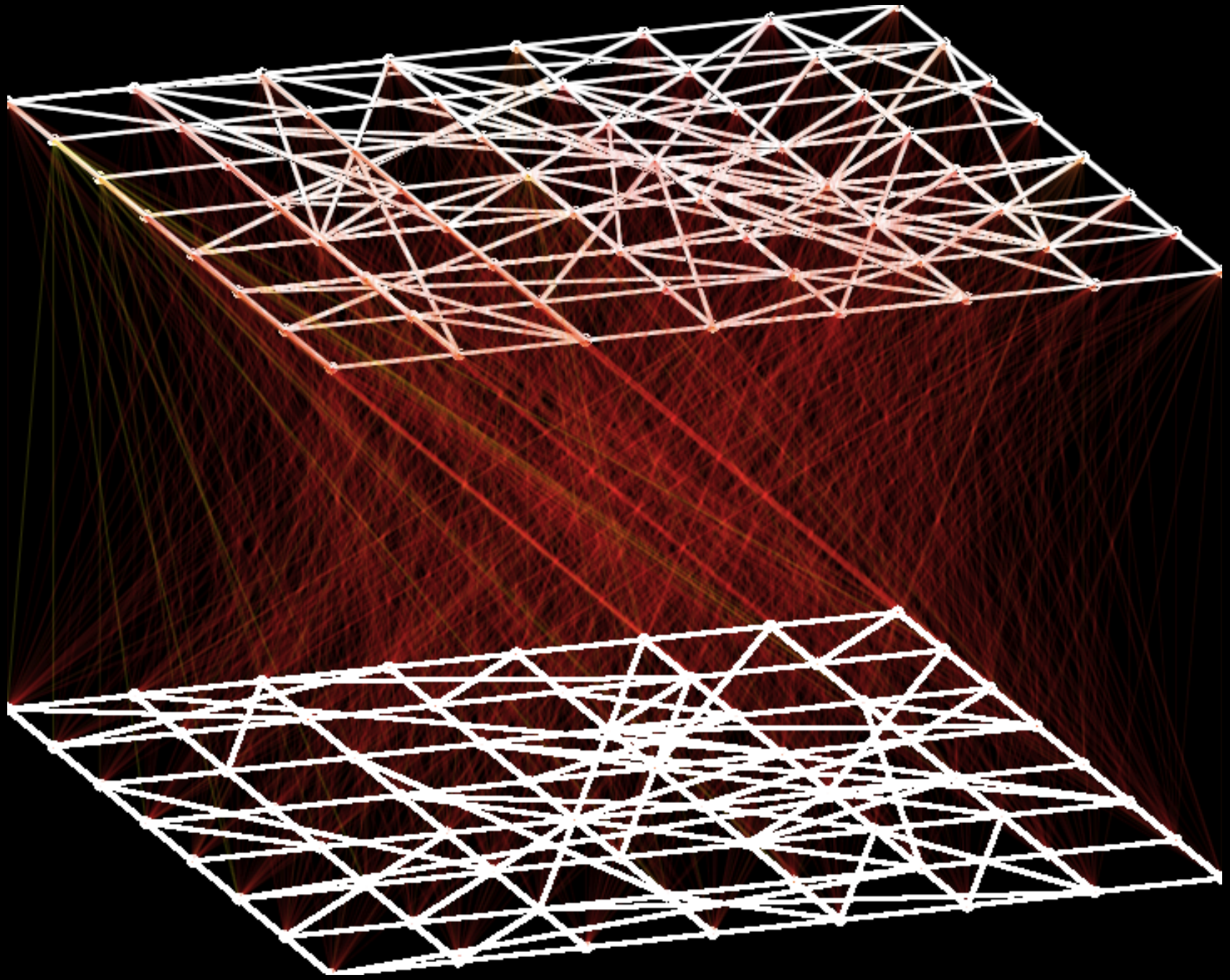
Find a 1-1 matching between vertices with as **many overlaps** as possible.

Our contributions

A new belief propagation method (Bayati et al. 2009, 2013)
Outperformed state-of-the-art PageRank and optimization-based heuristic methods

High performance C++ implementations (Khan et al. 2012)
40 times faster (C++ ~ 3, complexity ~ 2, threading ~ 8)
5 million edge alignments ~ 10 sec

`www.cs.purdue.edu/~dgleich/codes/netalignmc`



Iterative methods

for network alignment

Each iteration involves

Matrix-vector-ish computations
with a sparse matrix, e.g. sparse matrix vector products in a semi-ring, dot-products, axpy, etc.

Bipartite max-weight matching
using a different weight vector at each iteration

No "convergence"
100-1000 iterations

```
Let x[i] be the score for
each pair-wise match in L


for i=1 to ...
  update x[i] to y[i]
  compute a
    max-weight match
    with y
  update y[i] to x[i]
    (using match in MR)
```

Open question 1

Any sort of property of
these methods beyond ...

(i) Principled derivation and
(ii) “David and Ananth say they work”?

Belief propagation methods

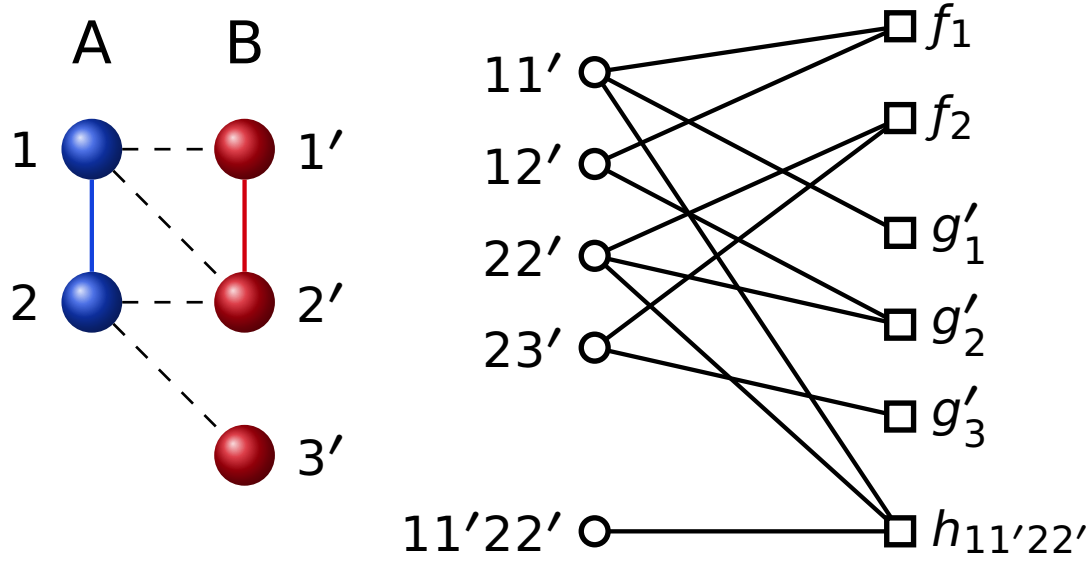
Summary

- Construct a probability model where the most likely state is the solution!
- Locally update information
- Like a generalized dynamic program
- It works
- *Most likely, it won't converge*

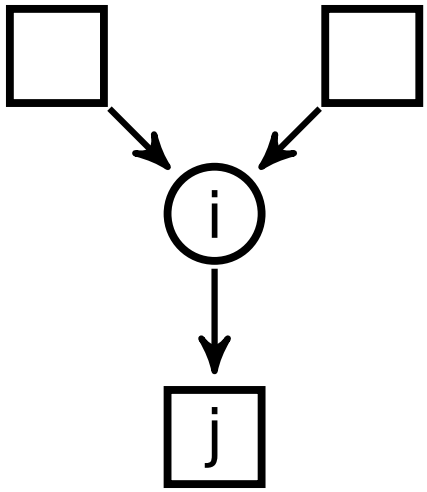
History

- BP used for computing marginal probabilities and maximum aposterori probability
- Wildly successful at solving satisfiability problems
- Convergent algorithm for max-weight matching

Belief propagation for network alignment

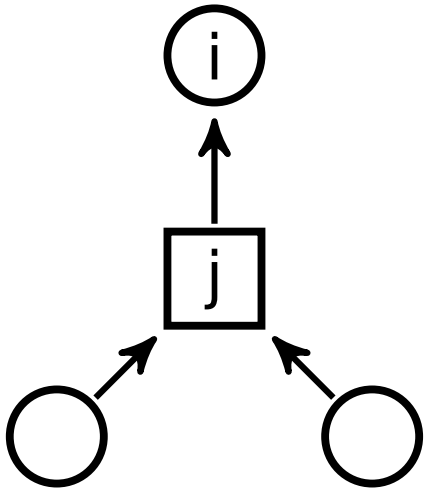


$$p(\boldsymbol{x}_L, \boldsymbol{x}_S) = \frac{1}{Z}\left[\prod_{i=1}^{n} f_i(\boldsymbol{x}_{\partial f_i}) \prod_{j=1}^{m} g_j(\boldsymbol{x}_{\partial g_j}) \prod_{ijrs \in V_S} h_{ijrs}(\boldsymbol{x}_{\partial h_{ijrs}})\right] e^{\alpha \boldsymbol{w}^T \boldsymbol{x}_L + \frac{\beta}{2} \mathbf{1}_{|S|}^T \boldsymbol{x}_S}$$



$$M_{i\to j}\{x_i = s\} = \prod_{j' \in \{N(i)\setminus j\}} M_{j'\to i}\{x_i = s\}$$

variable i tells function j what it thinks about being in state s. This is just the product of what all the other functions tell i about being in state s.



$$M_{j\to i}\{x_i = s\} = \max_{\substack{\mathbf{y}:\text{all possible choices} \\ \text{for variables } i' \in N(j)}} \left[f_j(\mathbf{y}) \prod_{i' \in \{N(j)\setminus i\}} M_{i'\to j}\{x'_i = y'_i\} \right]$$

function j tells variable i what it thinks about being in state s. This means that we have to locally maxamize f_j among all possible choices. Note $y_i = s$ always (too cumbersome to include in notation.)

Belief propagation for network alignment

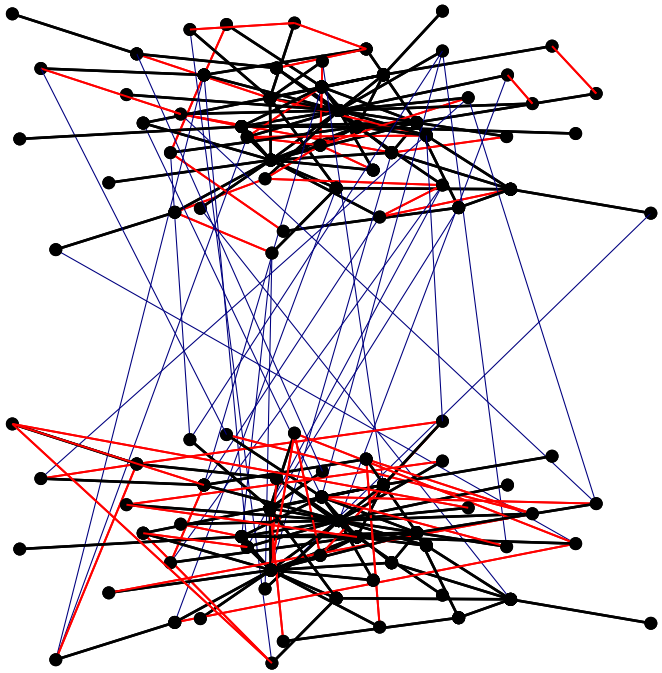
For $t \geq 1$, the messages in iteration t are obtained from the messages in iteration $t-1$ recursively. In particular for all $ii' \in E_L$

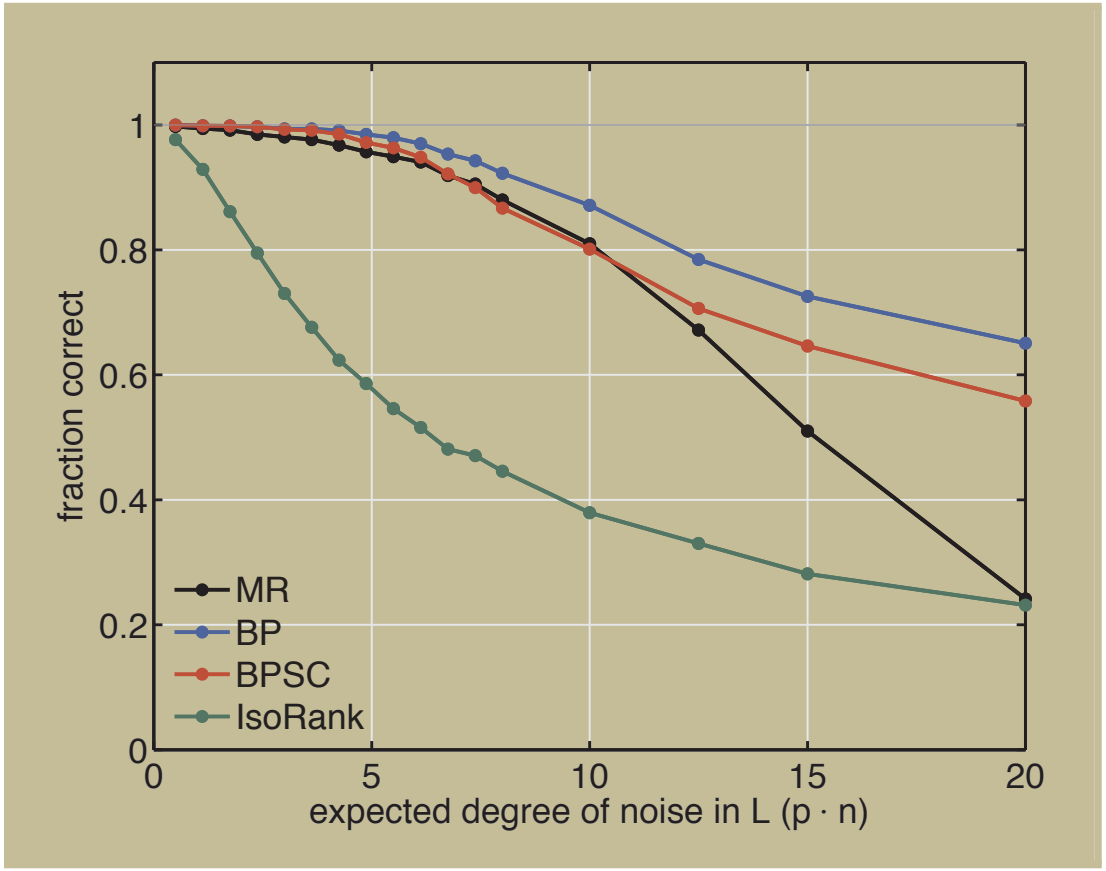
$$m^{(t)}_{ii' \to f_i} = \alpha w_{ii'} - \left(\max_{k \neq i} \left[m^{(t-1)}_{ki' \to g_{i'}} \right] \right)^+ + \sum_{ii'jj' \in V_S} \min \left(\frac{\beta}{2}, \max(0, \frac{\beta}{2} + m^{(t-1)}_{jj' \to h_{ii'jj'}}) \right). \quad (1)$$

The update rule for $m^{(t)}_{ii' \to g_{i'}}$ is similar, and

$$m^{(t)}_{ii' \to h_{ii'jj'}} = \alpha w_{ii'} - \left(\max_{k \neq i} \left[m^{(t-1)}_{ki' \to g_{i'}} \right] \right)^+ - \left(\max_{k' \neq i'} \left[m^{(t-1)}_{ik' \to f_i} \right] \right)^+ + \sum_{\substack{kk' \neq jj' \\ ii'kk' \in V_S}} \min \left(\frac{\beta}{2}, \max(0, m^{(t-1)}_{kk' \to h_{ii'kk'}} + \frac{\beta}{2}) \right). \quad (2)$$

Synthetic evaluation of network alignment





Open question 2

When could we hope to solve such synthetic problems in asymptotic regimes?

Does it work?

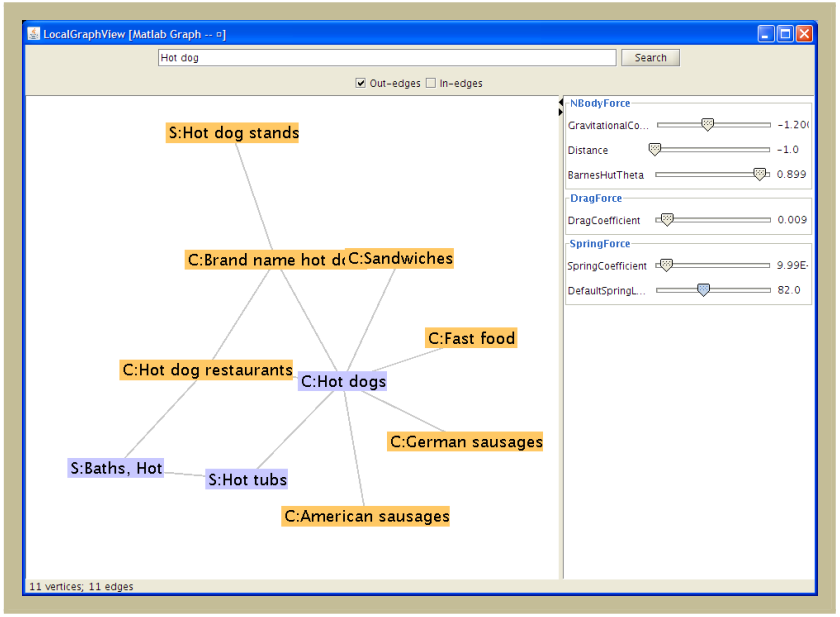
LCSH – Library of Congress subject headings
Rameau – French National Library subject headings
Manually matched

Obj.	Alg.	Weight	Overlap	Time (s)	Correct	Rec.	Prec.	Triangles
	Sol.	36332.42	39847	—	57645	100%	100%	2073
	MWM	93279.0	16990	29.6	29098	50.5%	23.3%	350
$\alpha = 1, \beta = 1$	BP	84622.0	46400	23522.0	32585	56.5%	27.6%	1515
	BP++	85810.1	46942	27115.6	32857	57.0%	27.4%	1548
	MR	87588.6	48367	33366.9	33225	57.6%	27.0%	1617
$\alpha = 1, \beta = 2$	BP	81752.6	46569	23427.1	31724	55.0%	27.6%	1483
	BP++	84615.7	46656	26673.1	31952	55.4%	26.7%	1531
	MR	85438.4	48934	56961.6	32303	56.0%	26.3%	1604
$\alpha = 0, \beta = 1$	BP	60617.9	45247	14284.8	24794	43.0%	23.2%	1467
	BP++	60502.8	41592	13979.5	24498	42.5%	23.0%	1484
	MR	65994.2	46163	10384.4	25455	44.2%	21.5%	1602

Open question 3

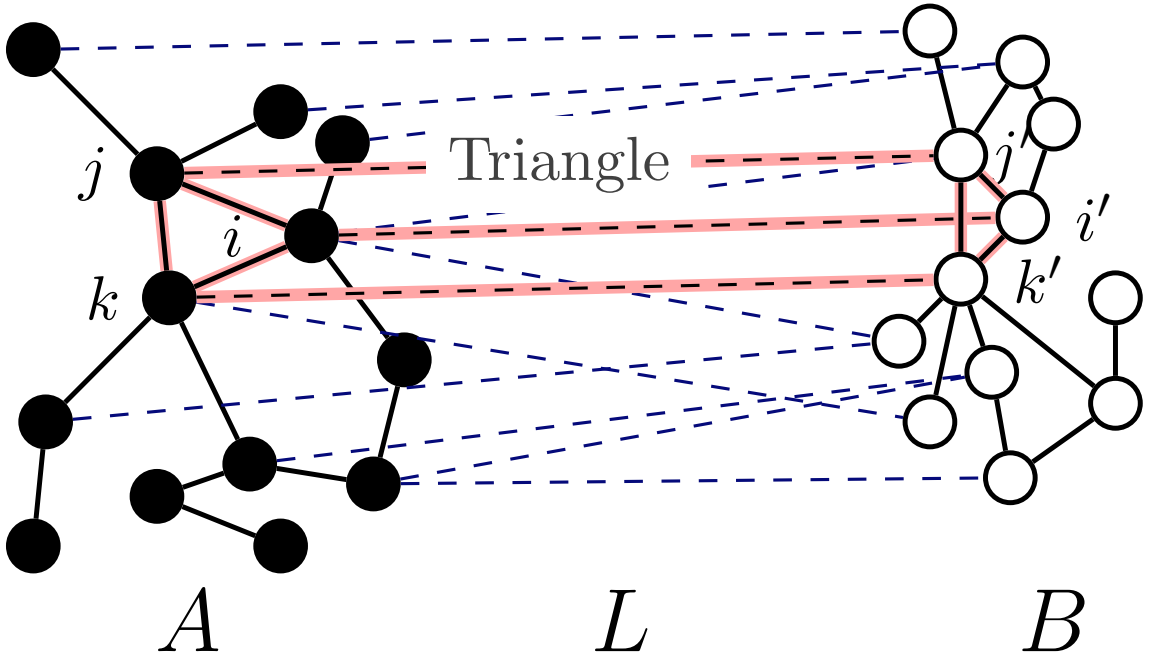
How can we evaluate alignments?
What are possible null-models?

LCSH	WC
Science fiction television series	Science fiction television programs
Turing test	Turing test
Machine learning	Machine learning
Hot tubs	Hot dog



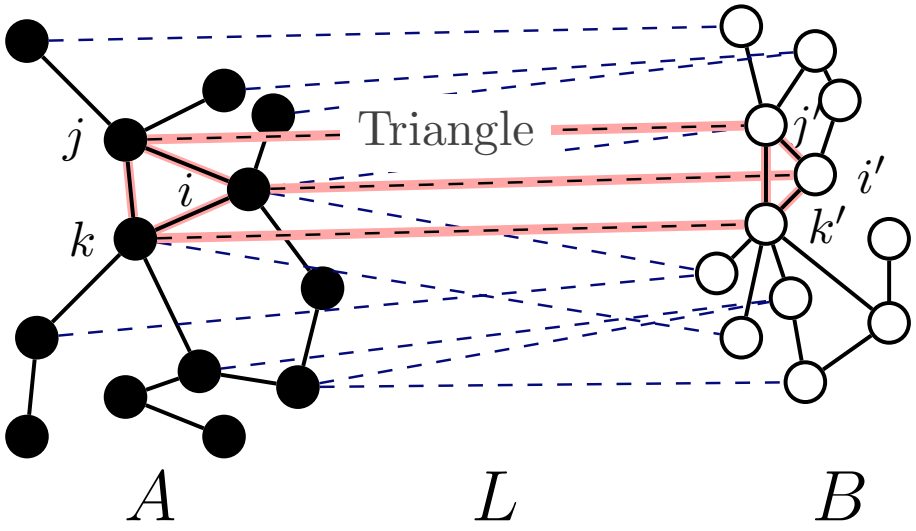
Higher-order network alignment

Joint with S. Mohammadi, T. Kolda



Network alignment

via mathematical programming



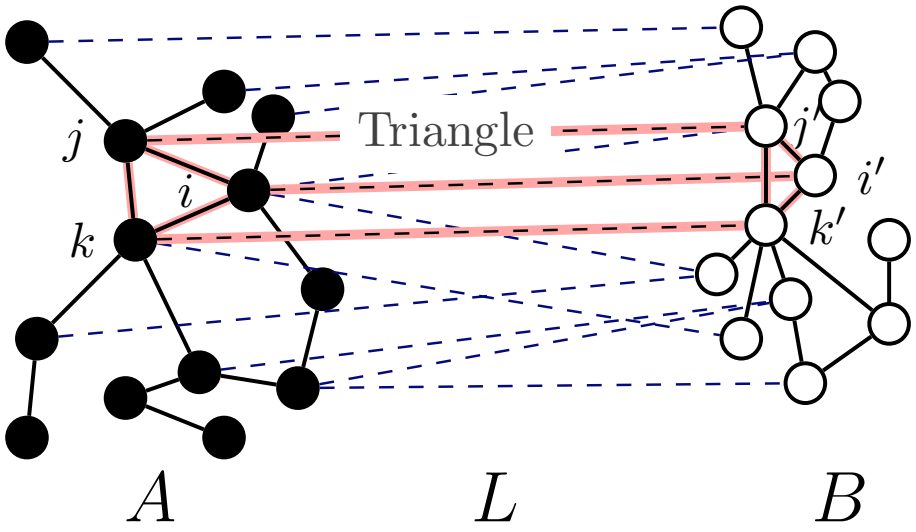
maximize $\alpha \mathbf{w}^T \mathbf{x} + \frac{\beta}{2} \mathbf{x}^T \mathbf{S} \mathbf{x}$

subject to $\mathbf{A}\mathbf{x} \leq \mathbf{e}, x_i \in \{0, 1\}$

Find a 1-1 matching between vertices with as many overlaps as possible.

Triangle alignment

via mathematical programming



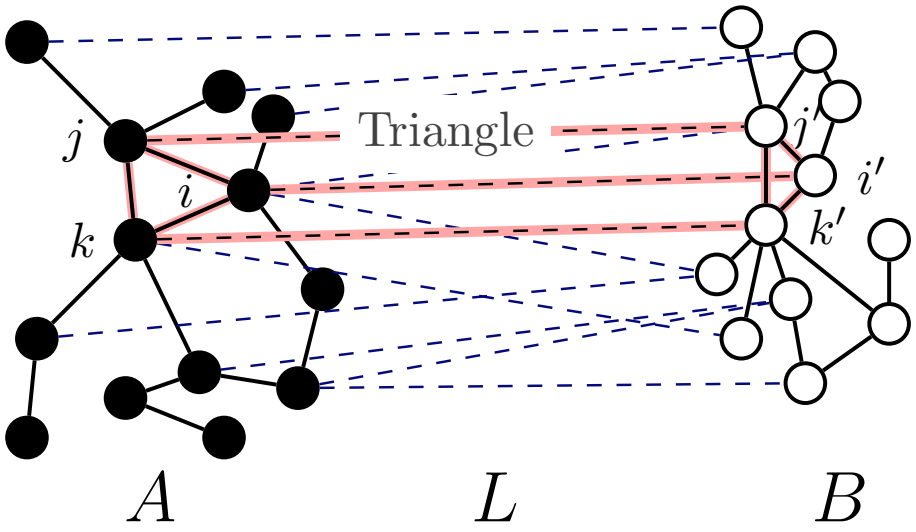
maximize $\alpha \mathbf{w}^T \mathbf{x} + \frac{\beta}{2} \mathbf{x}^T \mathbf{S} \mathbf{x} + \sum_{ijk} T_{ijk} x_i x_j x_k$

subject to $\mathbf{A}\mathbf{x} \leq \mathbf{e}, x_i \in \{0, 1\}$

Find a 1-1 matching between vertices with as many overlaps **and triangles** as possible.

Tensor eigenvalues

and a power method



$$\text{maximize} \quad \sum_{ijk} T_{ijk} x_i x_j x_k$$

$$\text{subject to} \quad \|\mathbf{x}\|_2 = 1$$

$$[\mathbf{x}^{(\text{next})}]_i = \rho \cdot \left(\sum_{jk} T_{ijk} x_j x_k + \gamma x_i\right)$$

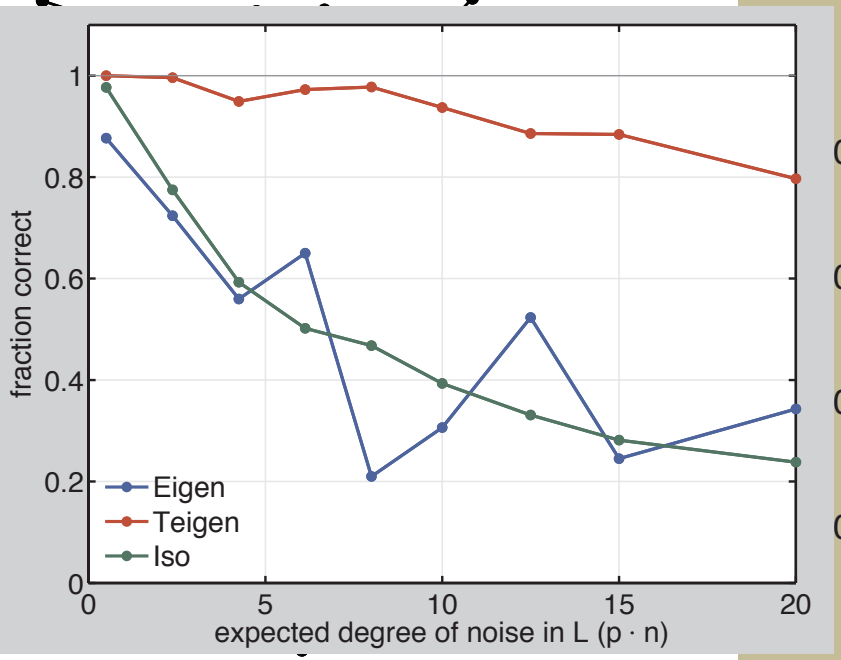
where ρ ensures the 2-norm

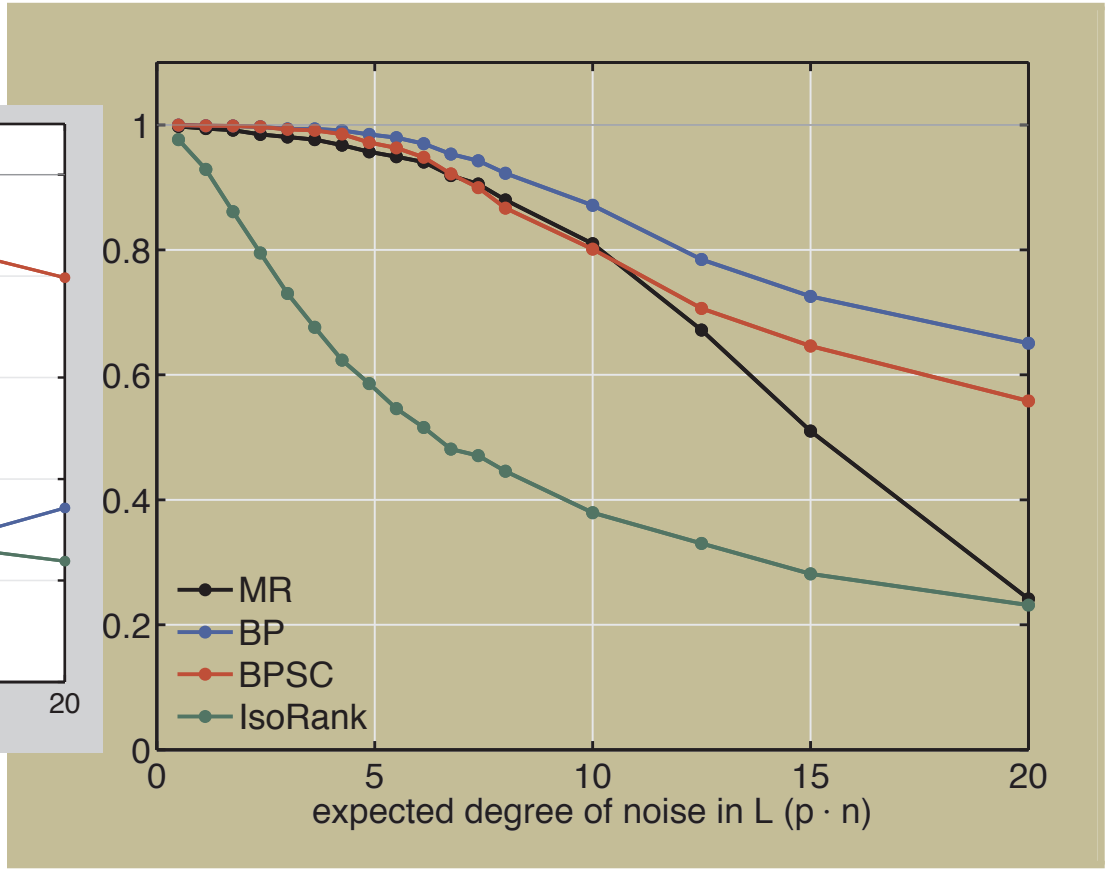
SSHOPM method due to Kolda and Mayo

Human protein interaction networks 48,228 triangles
Yeast protein interaction networks 257,978 triangles
The tensor T has ~100,000,000,000 nonzeros

We work with it implicitly

Synthetic evaluation of network alignment





Open question 4

When do we need triangles?