Applied and Computational Discrete Algorithms Shine in Making Sense of High-Throughput Biological Network Data: 3 Short Stories

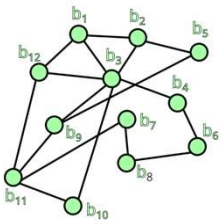
Lenore J. Cowen
Tufts University

First Short Story: Approximate IsoRank

Global Network Alignment Problem

One of the Most Famous Problems in Network Biology:

starts with a classical PPI network



Classical PPI Network:

Vertices: proteins

Edges: these proteins physically interact

Note: undirected, unweighted or confidence weights

Network Alignment Methods Over the Years

2008 **Global alignment of multiple protein interaction networks with application to functional orthology detection**
Rohit Singh, Jinbo Xu, and Bonnie Berger Authors Info & Affiliations

2010 **Topological network alignment uncovers biological function and phylogeny**
Oleksii Kuchaiev, Tijana Milenković, Vesna Memišević, Wayne Hayes and Nataša Pržulj

2012 **AlignNemo: A Local Network Alignment Method to Integrate Homology and Topology**
Giovanni Civetta, Marco Mina, Pietro H. Guzzi, Mario Cannataro, Concettina Guerra

2015 **Natalie 2.0: Sparse Global Network Alignment as a Special Case of Quadratic Assignment**
by Mohammed El-Kebir, Jaap Heringa and Gunnar W. Klau

2016 **FINAL: Fast Attributed Network Alignment**
Authors: Si Zhang, Hanghang Tong
Authors Info & Claims

2017 **Unified Alignment of Protein-Protein Interaction Networks**
Noël Malod-Dognin, Kristina Ban & Nataša Pržulj

2017 **SANA: simulated annealing far outperforms many other search algorithms for biological network alignment**
Nil Mamano, Wayne B Hayes

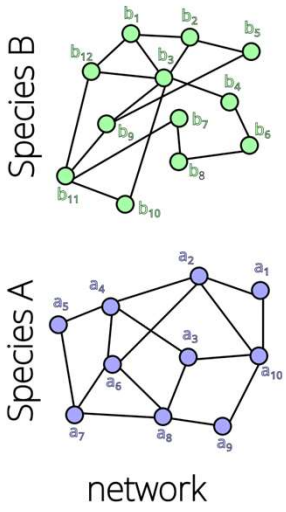
2019 **Functional protein representations from biological networks enable diverse cross-species inference**
Jason Fan, Anthony Cannistra, Inbar Fried, Tim Lim, Thomas Schaffner, Mark Crovella, Benjamin Hescott, Mark D M Leiserson Author Notes

2021 **NetQuilt: deep multispecies network-based protein function prediction using homology-informed network similarity**
Meet Barot, Vladimir Gligorijević, Kyunghyun Cho, Richard Bonneau

2022 **Network alignment and motif discovery in dynamic networks**
Review Article | Published: 03 October 2022

Global Network Alignment Problem: Description

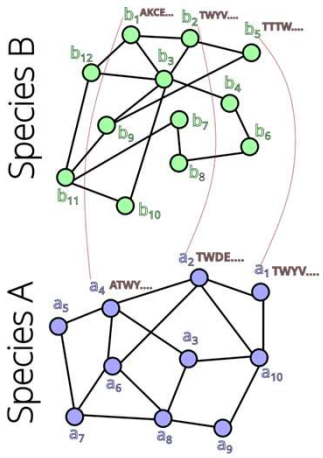
Input



network

Network Alignment Problem

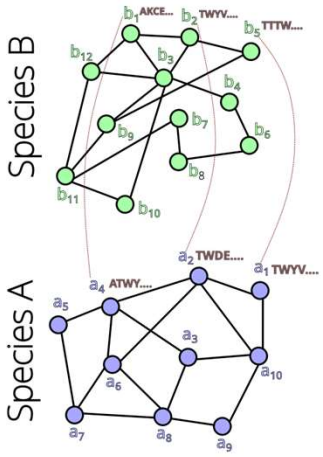
Input



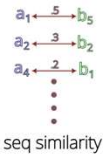
network

Network Alignment Problem

Input

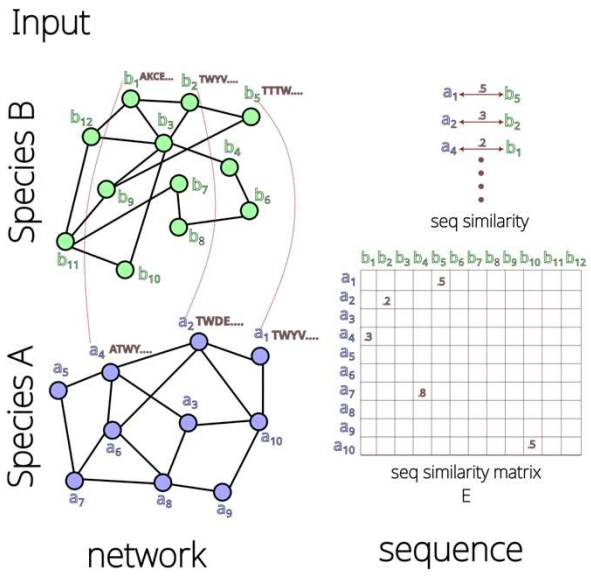


network

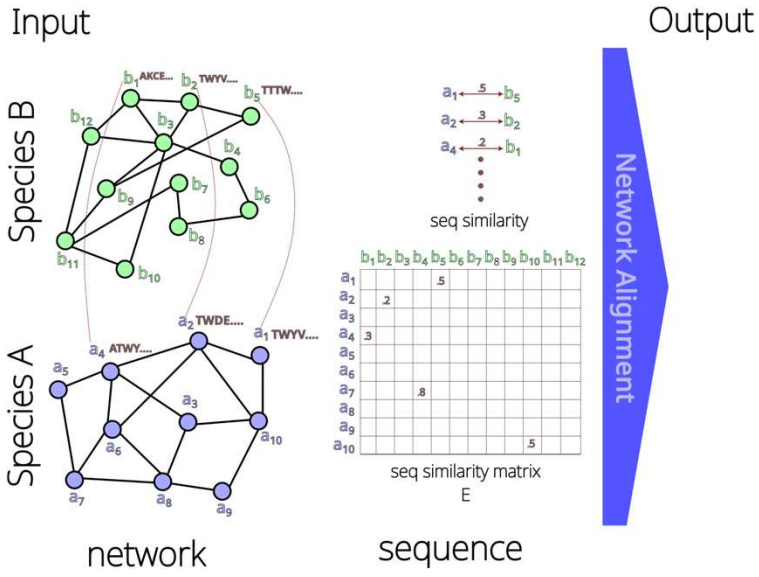


sequence

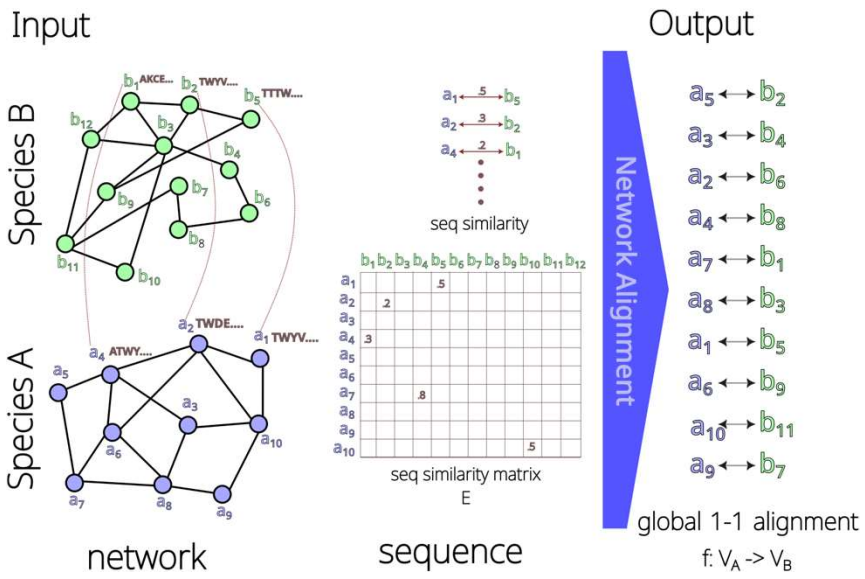
Network Alignment Problem



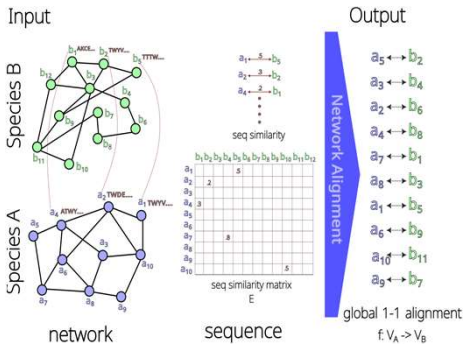
Network Alignment Problem



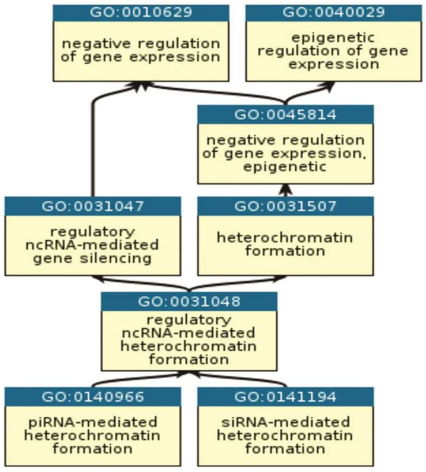
Network Alignment Problem



Network Alignment Problem: for functional annotation transfer

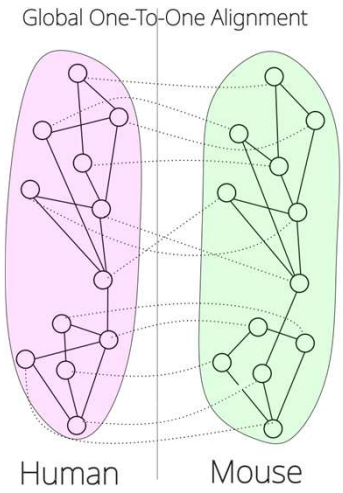


Functional annotation from GO

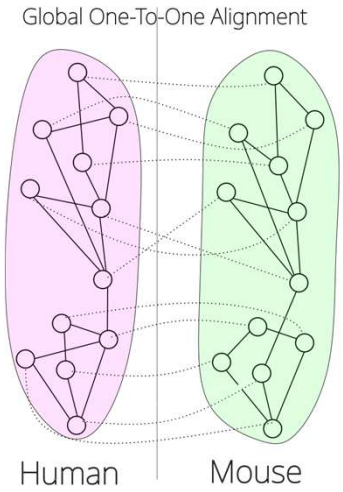


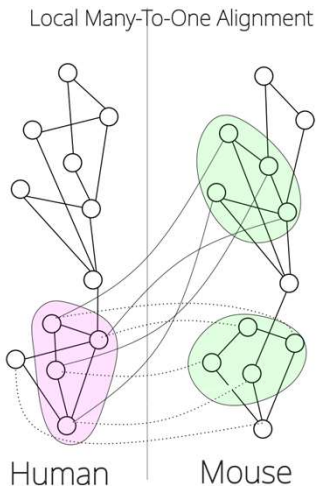
"The Gene Ontology knowledgebase in 2026." *Nucleic Acids Research* 54, no. D1 (2026): D1779-D1792.

Global vs Local Network Alignment



Global vs Local Network Alignment





Classical tool: IsoRank[1]

- Won the 2019 RECOMB test of time award (presented in RECOMB 2007)
- Popular global alignment algorithm that uses network structure and sequence.
- Uses spectral graph properties to find one to one alignment. **Uses the global network topology to find alignments.**
- Based on the idea that, if a node of a source PPI is aligned with node from a target PPI, their neighbors should be aligned as well

[1]Singh, Rohit, Jinbo Xu, and Bonnie Berger. "Global alignment of multiple protein interaction networks with application to functional orthology detection." *Proceedings of the National Academy of Sciences* 105, no. 35 (2008): 12763-12768.

IsoRank Algorithm: Greedy Alignment

Input:

- $G_1 = (V_1, E_1)$,

  $G_2 = (V_2, E_2)$, where

  $n_1 = |V_1|, n_2 = |V_2|$

- Sequence Similarity **(Optional)** : $E = \mathrm{R}^{n_1 \times n_2}$

Output:

- Network alignment matrix:

  $R \in \mathrm{R}^{n_1 \times n_2}$

Greedy alignment on R to produce $f : V_1 \rightarrow V_2$

IsoRank Algorithm: Greedy Alignment

Input:

- $G_1 = (V_1, E_1)$,

  $G_2 = (V_2, E_2)$, where

  $n_1 = |V_1|, n_2 = |V_2|$

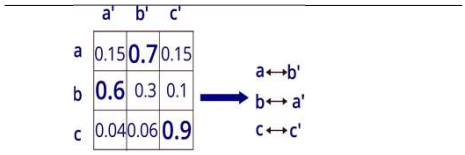
- Sequence Similarity **(Optional)** : $E = \mathrm{R}^{n_1 \times n_2}$

Output:

- Network alignment matrix:

  $R \in \mathrm{R}^{n_1 \times n_2}$

Greedy alignment on R to produce $f : V_1 \rightarrow V_2$



IsoRank Algorithm: Greedy Alignment

Input:

- $G_1 = (V_1, E_1)$,

  $G_2 = (V_2, E_2)$, where

  $n_1 = |V_1|, n_2 = |V_2|$

- Sequence Similarity **(Optional)** : $E = \mathrm{R}^{n_1 \times n_2}$

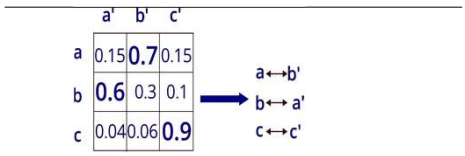
- Two ways of formulating IsoRank:
  1. sequence similarity not used
  2. with sequence similarity

Output:

- Network alignment matrix:

  $R \in \mathrm{R}^{n_1 \times n_2}$

Greedy alignment on R to produce $f : V_1 \rightarrow V_2$



IsoRank Algorithm: Without Sequence Similarity

IsoRank Algorithm: Without Sequence Similarity

- To find the similarity between $i \in V_1, j \in V_2$ (i.e. R_{ij})

$$R_{ij} = \sum_{u \in N(i)} \sum_{v \in N(j)} \frac{1}{|N(u)N(v)|} R_{uv} \qquad i \in V_1, j \in V_2$$

IsoRank Algorithm: Without Sequence Similarity

- To find the similarity between $i \in V_1, j \in V_2$ (i.e. R_{ij})

$$R_{ij} = \sum_{u \in N(i)} \sum_{v \in N(j)} \frac{1}{|N(u)N(v)|} R_{uv} \qquad i \in V_1, j \in V_2$$

- The problem becomes $R = AR$ (1)

$$A_{ij}^{uv} = \begin{cases} \frac{1}{|N(u)N(v)|} & \text{if } (i,u) \in E_1,\ (j,v) \in E_2 \\ 0 & \text{o.w.} \end{cases}$$

IsoRank Algorithm: Without Sequence Similarity

- To find the similarity between $i \in V_1, j \in V_2$ (i.e. R_{ij})

$$R_{ij} = \sum_{u \in N(i)} \sum_{v \in N(j)} \frac{1}{|N(u)N(v)|} R_{uv} \qquad i \in V_1, j \in V_2$$

- The problem becomes $R = AR$ (1)

$$A_{ij}^{uv} = \begin{cases} \frac{1}{|N(u)N(v)|} & \text{if } (i,u) \in E_1,\ (j,v) \in E_2 \\ 0 & \text{o.w.} \end{cases}$$

- Original IsoRank obtained this by initializing and iterating till convergence.

IsoRank Algorithm: Without Sequence Similarity

- To find the similarity between $i \in V_1, j \in V_2$ (i.e. R_{ij})

$$R_{ij} = \sum_{u \in N(i)} \sum_{v \in N(j)} \frac{1}{|N(u)N(v)|} R_{uv} \qquad i \in V_1, j \in V_2$$

- The problem becomes $R = AR$ (1)

We recognize 1 as the problem of finding dominant Eigenvectors

$$A_{ij}^{uv} = \begin{cases} \frac{1}{|N(u)N(v)|} & \text{if } (i,u) \in E_1,\ (j,v) \in E_2 \\ 0 & \text{o.w.} \end{cases}$$

- Original IsoRank obtained this by initializing and iterating till convergence.

IsoRank Algorithm: With Sequence Similarity

IsoRank Algorithm: With Sequence Similarity

- The problem becomes

$$R = (1 - \alpha)AR + \alpha E \qquad (2)$$

where, A is the **same** 4-tensor

$$A_{ij}^{uv} = \begin{cases} \frac{1}{|N(u)N(v)|} & \text{if } (i,u) \in E_1,\ (j,v) \in E_2 \\ 0 & \text{o.w.} \end{cases}$$

IsoRank Algorithm: With Sequence Similarity

- The problem becomes

$$R = (1 - \alpha)AR + \alpha E \qquad (2)$$

where, A is the **same** 4-tensor

$$A_{ij}^{uv} = \begin{cases} \frac{1}{|N(u)N(v)|} & \text{if } (i,u) \in E_1,\ (j,v) \in E_2 \\ 0 & \text{o.w.} \end{cases}$$

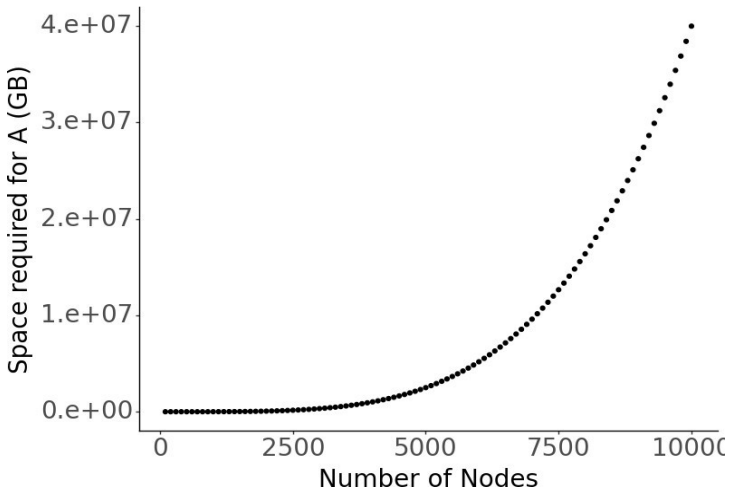
- Linear system. Iterative linear solver.

Space consideration for *A*

- As $A \in \mathbb{R}^{(n_1 \times n_2) \times (n_1 \times n_2)}$, the dense computation of *A* is unfeasible in time and space.

Space consideration for *A*

- As $A \in \mathbb{R}^{(n_1 \times n_2) \times (n_1 \times n_2)}$, the dense computation of *A* is unfeasible in time and space.



network	#nodes	#edges	avg degree	avg cc
human	32431	301529	18.595	0.069
fly	11247	42555	7.567	0.027
mouse	13967	36541	5.232	0.051
rat	10792	23315	4.320	0.075
yeast	6478	70638	21.808	0.285

Tensor Reformulation of IsoRank and its fast computation

Exact solution when no sequence similarity

- Let $A_1 \in \mathbb{R}^{n_1 \times n_1}$, $A_2 \in \mathbb{R}^{n_2 \times n_2}$ be the adjacency matrices of G_1 and G_2 respectively

Exact solution when no sequence similarity

- Let $A_1 \in \mathbb{R}^{n_1 \times n_1}$, $A_2 \in \mathbb{R}^{n_2 \times n_2}$ be the adjacency matrices of G_1 and G_2 respectively
- Construct P_1, P_2 as

$$P_1 = A_1 D_1^{-1}$$
$$P_2 = A_2 D_2^{-1}$$

Exact solution when no sequence similarity

- Let $A_1 \in \mathbb{R}^{n_1 \times n_1}$, $A_2 \in \mathbb{R}^{n_2 \times n_2}$ be the adjacency matrices of G_1 and G_2 respectively
- Construct P_1, P_2 as

$$P_1 = A_1 D_1^{-1}$$
$$P_2 = A_2 D_2^{-1}$$

- Then, $A = P_1 \otimes P_2$.

Exact solution when no sequence similarity

- Let $A_1 \in R^{n_1 \times n_1}$, $A_2 \in R^{n_2 \times n_2}$ be the adjacency matrices of G_1 and G_2 respectively
- Construct P_1, P_2 as

$$P_1 = A_1 D_1^{-1}$$
$$P_2 = A_2 D_2^{-1}$$

- Then, $A = P_1 \otimes P_2$.

$$\begin{pmatrix} a_0 & a_1 \\ a_2 & a_3 \end{pmatrix} \otimes \begin{pmatrix} b_0 & b_1 \\ b_2 & b_3 \end{pmatrix} = \begin{pmatrix} a_0b_0 & a_0b_1 & a_1b_0 & a_1b_1 \\ a_0b_2 & a_0b_3 & a_1b_2 & a_1b_3 \\ a_2b_0 & a_2b_1 & a_3b_0 & a_3b_1 \\ a_2b_2 & a_2b_3 & a_3b_2 & a_3b_3 \end{pmatrix}$$

Exact solution when no sequence similarity

- Let $A_1 \in R^{n_1 \times n_1}$, $A_2 \in R^{n_2 \times n_2}$ be the adjacency matrices of G_1 and G_2 respectively
- Construct P_1, P_2 as

$$P_1 = A_1 D_1^{-1}$$
$$P_2 = A_2 D_2^{-1}$$

- Then, $A = P_1 \otimes P_2$.

$$\begin{pmatrix} a_0 & a_1 \\ a_2 & a_3 \end{pmatrix} \otimes \begin{pmatrix} b_0 & b_1 \\ b_2 & b_3 \end{pmatrix} = \begin{pmatrix} a_0b_0 & a_0b_1 & a_1b_0 & a_1b_1 \\ a_0b_2 & a_0b_3 & a_1b_2 & a_1b_3 \\ a_2b_0 & a_2b_1 & a_3b_0 & a_3b_1 \\ a_2b_2 & a_2b_3 & a_3b_2 & a_3b_3 \end{pmatrix}$$

- If R = vec (R), R=AR (3)

Contribution 1: Exact solution when only graph similarity considered (sequence similarity not used)

- If $d_1 = A_1 e$, $d_2 = A_2 e$, where e is an all-1 column vector

Exact solution when only graph similarity considered (sequence similarity not used)

- If $d_1 = A_1 e$, $d_2 = A_2 e$, where e is an all-1 column vector
- If $\pi = d_1 \otimes d_2$, then

$$A(d_1 \otimes d_2) = (P_1 \otimes P_2)(d_1 \otimes d_2)$$

$$= (P_1 d_1) \otimes (P_2 d_2)$$

$$= d_1 \otimes d_2 \qquad (4)$$

Exact solution when only graph similarity considered (sequence similarity not used)

- If $d_1 = A_1 e$, $d_2 = A_2 e$, where e is an all-1 column vector
- If $\pi = d_1 \otimes d_2$, then

$$A(d_1 \otimes d_2) = (P_1 \otimes P_2)(d_1 \otimes d_2)$$
$$= (P_1 d_1) \otimes (P_2 d_2)$$
$$= d_1 \otimes d_2 \qquad (4)$$

- So $\pi = d_1 \otimes d_2$ is a closed-form solution to the problem

Main Contribution: R^0: A very good guess for IsoRank, when sequence similarity given

- IsoRank with sequence similarity:

$$R = \alpha E + (1 - \alpha) AR$$
$$\| E \|_1 = 1$$

Contribution 2: R^0: A very good guess for IsoRank, when sequence similarity given

- IsoRank with sequence similarity:

$$R = \alpha E + (1 - \alpha)AR \qquad \|E\|_1 = 1$$

- We use the close-form solution ($\boldsymbol{\pi}$) to guess the IsoRank solution, which we call R^0

Contribution 2: R^0: A very good guess for IsoRank, when sequence similarity given

- IsoRank with sequence similarity:

$$R = \alpha E + (1 - \alpha)AR \qquad \|E\|_1 = 1$$

- We use the close-form solution ($\boldsymbol{\pi}$) to guess the IsoRank solution, which we call R^0

$$R^0 = \alpha E + (1 - \alpha)\boldsymbol{\pi} \qquad (5)$$

Where recall $\boldsymbol{\pi}$ is the steady state solution of the tensor product graph with $\boldsymbol{\pi} = \boldsymbol{d}/\boldsymbol{d}_{total}$

And $\mathbf{d}=D\mathbf{1}_{|V_1||V_2|}$ And $d_{total} = \mathbf{d}^T\mathbf{1}_{|V_1||V_2|}$

Contribution 3: Apply 1-step IsoRank update to get intermediate IsoRank result

- Given our original guess R^0

Contribution 3: Apply 1-step IsoRank update to get intermediate IsoRank result

- Given our original guess R^0
- Apply 1-step IsoRank update to get R^1

$$R^1 = \alpha E + (1 - \alpha)AR^0$$

Contribution 3: Apply 1-step IsoRank update to get intermediate IsoRank result

- Given our original guess R^0
- Apply 1-step IsoRank update to get R^1

$$R^1 = \alpha E + (1 - \alpha) A R^0$$

$R^1, R^2, R^3, \ldots, R^n$

Contribution 3: Apply 1-step IsoRank update to get intermediate IsoRank result

- Given our original guess R^0
- Apply 1-step IsoRank update to get R^1

$$R^1 = \alpha E + (1 - \alpha) A R^0$$

$R^1, R^2, R^3, \ldots, R^n$

- To compute AR^0, we apply the identity

$$AR^0 = P_1 \otimes P_2 R^0$$

$$= P_2 \, vec^{-1}(R^0) P_1^T \qquad (6)$$

- Note that, applying (6) until convergence gives us the true IsoRank R

Speedup: Construct a Barabasi-Albert random graph with ~5x edges as vertices and a random permutation

V1, V2 = 2,500; 5,000**; 10,000**

E1,E2 ~ 5V

Speedup

V1, V2=10000

> 7 hrs

Original Implementation

Speedup

V1, V2=10000

> 7 hrs

Original Implementation

41 seconds

~ 600 times faster

Our **R** implementation

Speedup

V1, V2=10000

> 7 hrs

Original Implementation

41 seconds

~ 600 times faster

Our **R** implementation

6 seconds

~4200 times faster

R1

Speedup

V1, V2=10000

> 7 hrs	41 seconds
	~ 600 times faster
Original Implementation	Our **R** implementation
6 seconds	3 seconds
~4200 times faster	**~8400 times faster**
R1	R0

How good are R^0 and R^1 approximations?
Analysis using Synthetic Networks.

Synthetic Networks Construction

- We know the true alignment

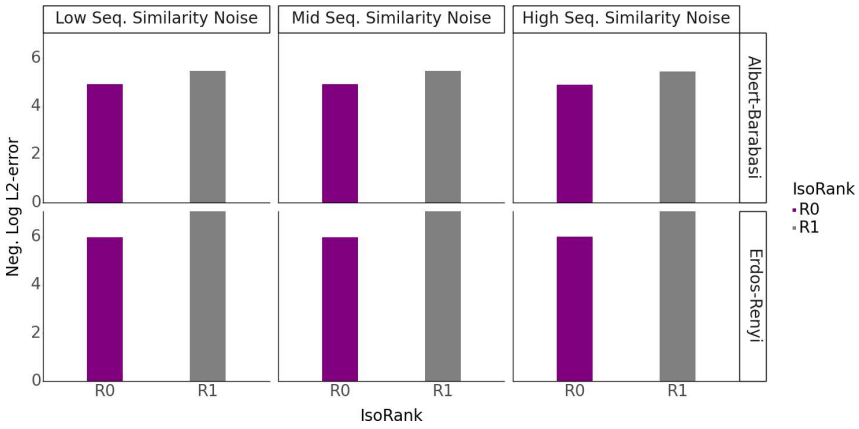
Synthetic Networks Construction

- We know the true alignment
- Introduce noise to the seq. similarity to obtain performances at different noise level.

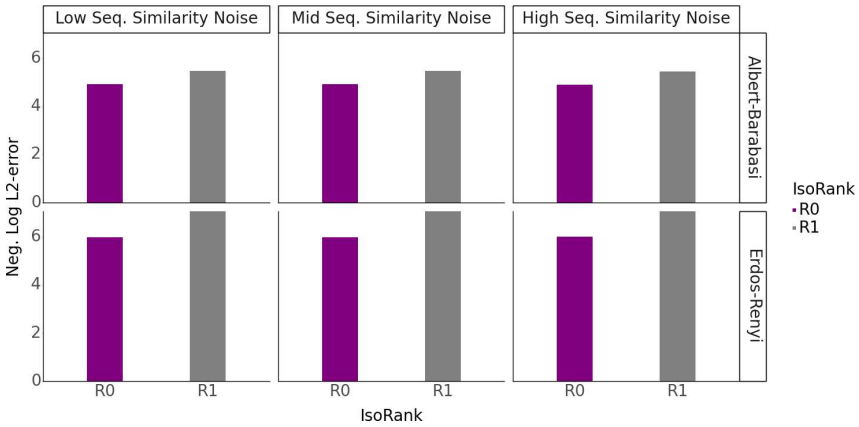
Synthetic Networks Construction

- We know the true alignment
- Introduce noise to the seq. similarity to obtain performances at different noise level.
- Used Erdős Rényi (ER) and Barabási-Albert (BA) graphs.
- How different are R^0, R^1 compared to the true R in terms of *L*2 norm?

Error Results: number of nodes = 10,000



Error Results: number of nodes = 10,000



- Turns out the approximations are really close to the true IsoRank.

Biological Networks

Biological networks

- Used IntAct database [2]

network	#nodes	#edges	avg degree	avg cc
human	32431	301529	18.595	0.069
fly	11247	42555	7.567	0.027
mouse	13967	36541	5.232	0.051
rat	10792	23315	4.320	0.075
yeast	6478	70638	21.808	0.285

[2] Noemi del Toro et al. The IntAct database: efficient access to fine-grained molecular interaction data, Nucleic Acids Research, 7 January 2022.

Evaluation Metrics for Biological Experiments

- From the R-approximations, used greedy method to find the top 2000 one-to-one mappings for analysis

Evaluation Metrics for Biological Experiments

- From the R-approximations, used greedy method to find the top 2000 one-to-one mappings for analysis
- Two types of metrics used:
  1. **Graph-theoretic**
     - How similar are the local neighborhoods produced by the global mappings?
  2. **Function-based**
     - Uses GO functional labels

Graph-theoretic methods

- Edge Correctness[3]: Measures if an edge in one network is also present in another network

[3]Hashemifar S, Xu J. HubAlign: an accurate and efficient method for global alignment of protein-protein interaction networks. Bioinformatics. 2014 Sep 1;30(17):i438-44. doi: 10.1093/bioinformatics/btu450. PMID: 25161231; PMCID: PMC4147903.

Graph-theoretic methods

- Edge Correctness[3]: Measures if an edge in one network is also present in another network
- Largest Common Connected Subgraph (LCCS): Measures the largest connected component of the transferred edges

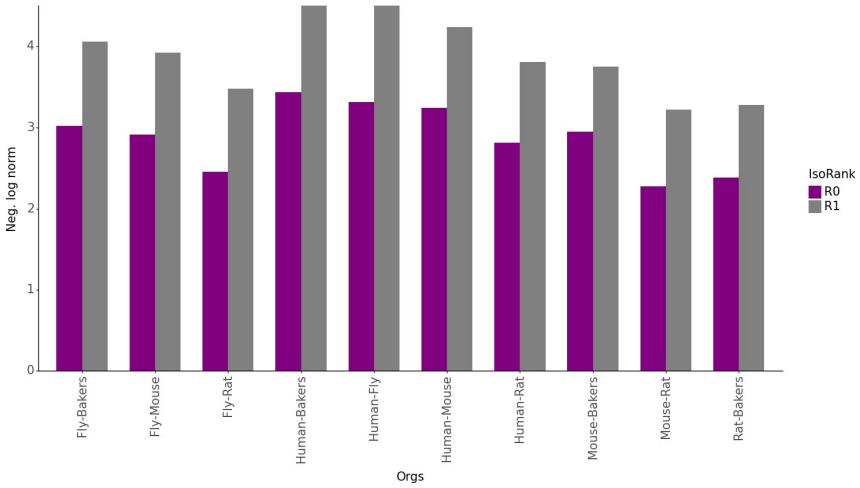
[3]Hashemifar S, Xu J. HubAlign: an accurate and efficient method for global alignment of protein-protein interaction networks. Bioinformatics. 2014 Sep 1;30(17):i438-44. doi: 10.1093/bioinformatics/btu450. PMID: 25161231; PMCID: PMC4147903.

Function-based methods

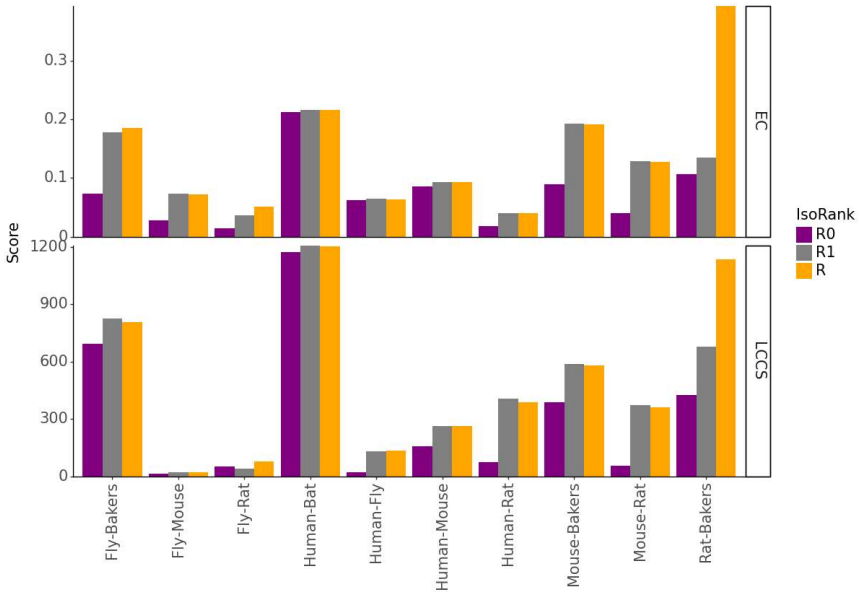
- Average Functional Similarity (AFS): Measures the average Schlicker's similarity[4] between the one-one couplings obtained.
- We used all three GO hierarchies.

[4]A. Schlicker, F. S. Domingues, J. Rahnenf¨uhrer, and T. Lengauer. A new measure for functional similarity of gene products based on gene ontology. BMC bioinformatics, 7(1):1–16, 2006.

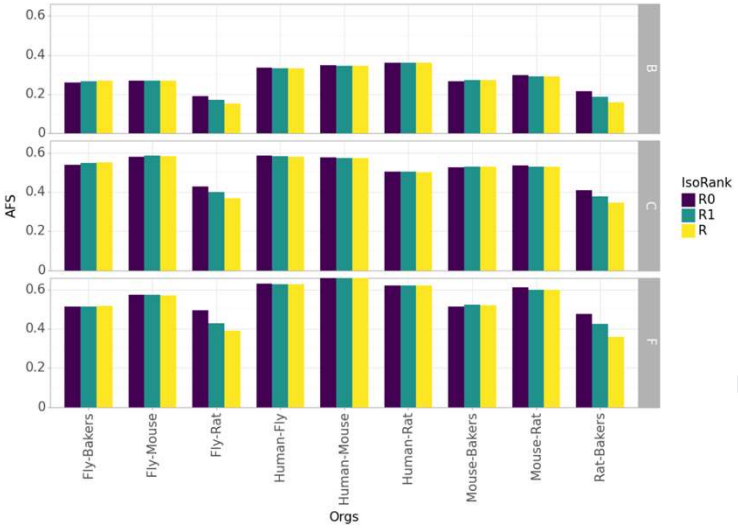
Approximation Results on Biological Network - Norms



Network measures on Biological Network - EC & LCCS



Function-based measures on Biological Network - *AFS*

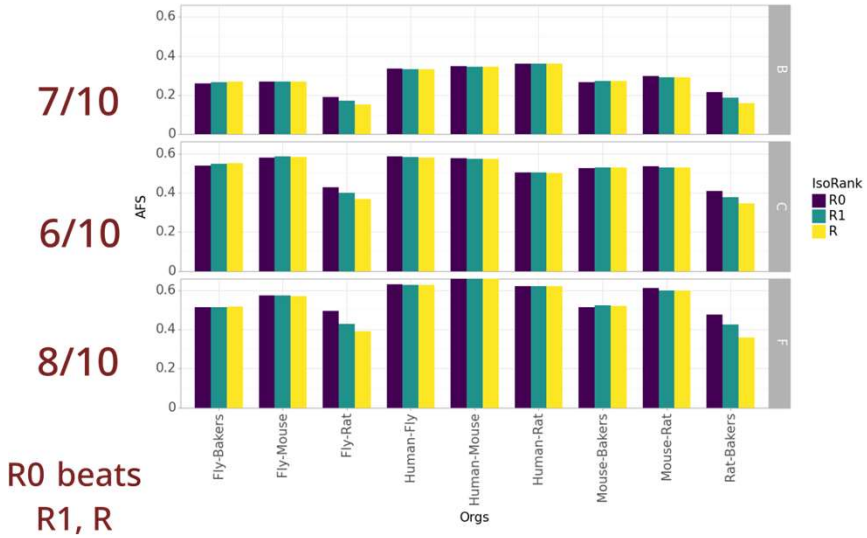


Biological Process

Cellular Component

Molecular function

Function-based measures on Biological Network -*AFS*



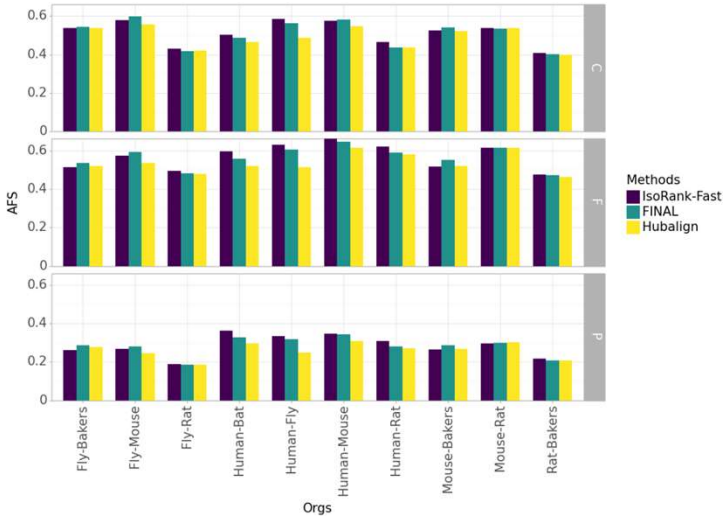
Comparisons of R^0 with other methods

- Comparison methods:
  - Hubalign[5]
  - Fast Attributed Network Alignment (FINAL)[6]

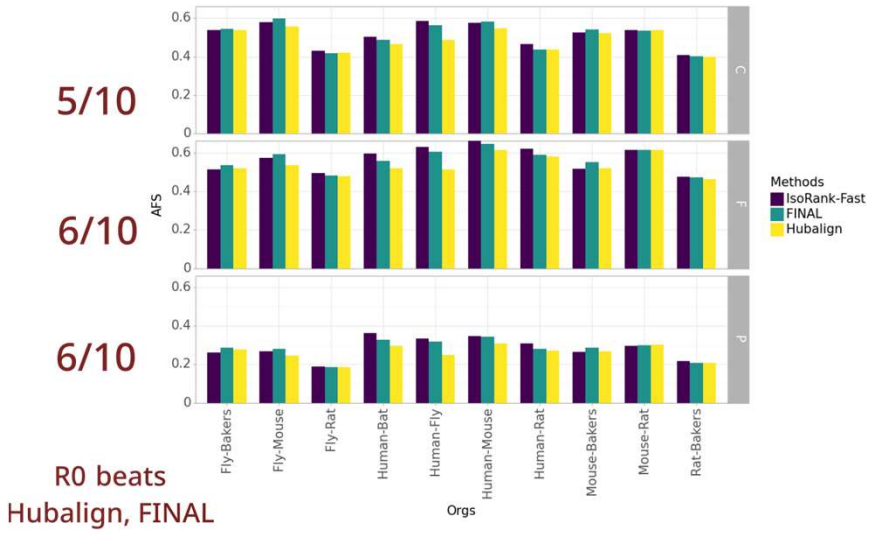
[5]Hashemifar S et al. An accurate and efficient method for global alignment of protein-protein interaction networks. 2014 Sept.

[6]Zhang, Si, and Hanghang Tong. "Final: Fast attributed network alignment." In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1345-1354. 2016.

Function-based comparisons with competing methods- *AFS*



Function-based comparisons with competing methods- *AFS*



R0 beats Hubalign, FINAL

Summary

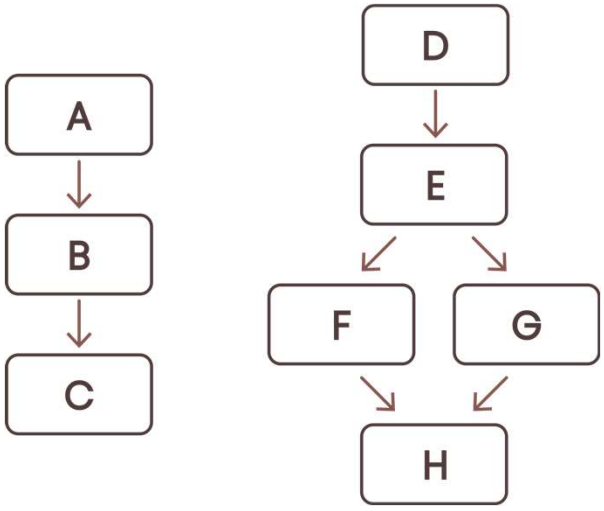
- The scalability of original IsoRank deteriorated with the number of nodes
- We propose an exact very-fast algorithm for IsoRank (when sequence information not used)
- Very fast and functionally informative guess R^0, when sequence similarity provided.
- One step update on R^0 to produce an approximation closer to IsoRank (called R^1)
- R^0 outperforming R^1 and true IsoRank R on biological networks, when function based metric was used for comparison.

Compensatory Pathways in Genetic Interaction Networks:

A different kind of interaction network

What is a compensatory pathway?

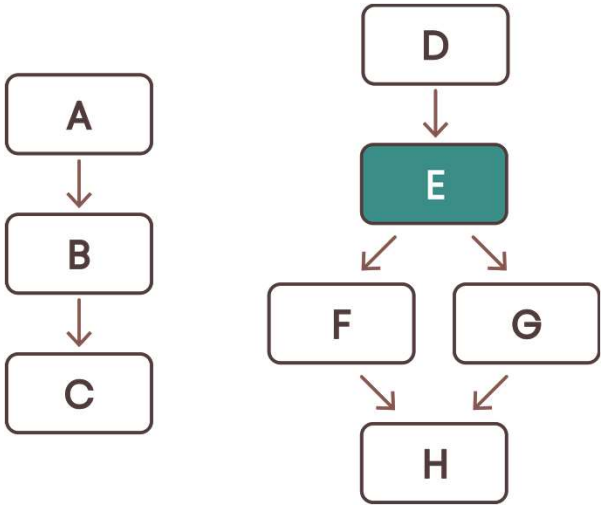
Imagine that **essential DNA repair** can be performed by **two distinct** cell pathways



What is a compensatory pathway?

Imagine that **essential DNA repair** can be performed by **two distinct** cell pathways

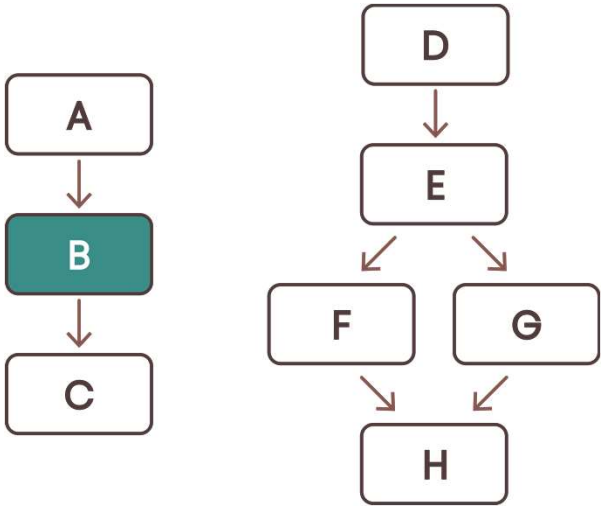
- If a gene in **one** pathway is mutated, the cell can still perform that essential function (70% capacity without E



What is a compensatory pathway?

Imagine that **essential DNA repair** can be performed by **two distinct** cell pathways

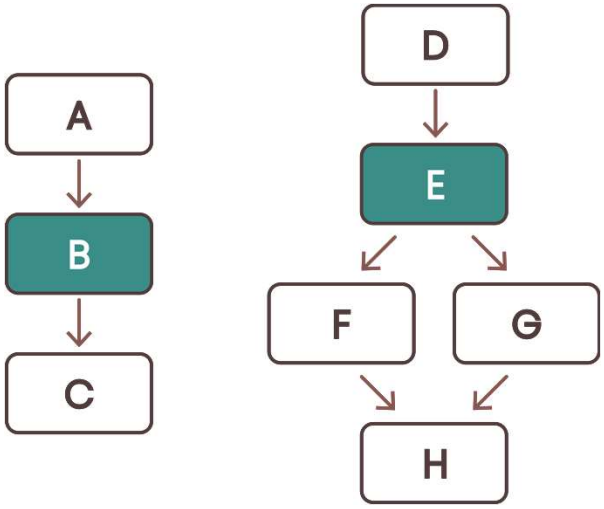
- If a gene in **one** pathway is mutated, the cell can still perform that essential function (70% capacity without E and 50% without B)



What is a compensatory pathway?

Imagine that **essential DNA repair** can be performed by **two distinct** cell pathways

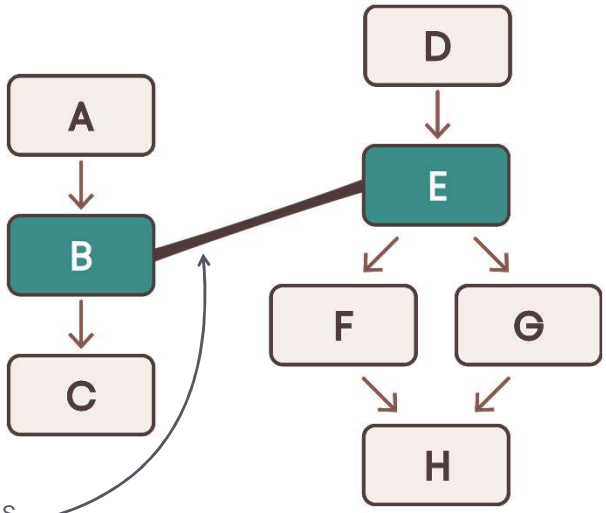
- If a gene in **one** pathway is mutated, the cell can still perform that essential function (**70% capacity without** E and 50% without B)
- If a gene in **both** pathways is mutated, the cell dies without being able to repair DNA



What is a compensatory pathway?

Imagine that **essential DNA repair** can be performed by **two distinct** cell pathways

- If a gene in **one** pathway is mutated, the cell can still perform that essential function (**70% capacity without** E and 50% without B)
- If a gene in **both** pathways is mutated, the cell dies without being able to repair DNA

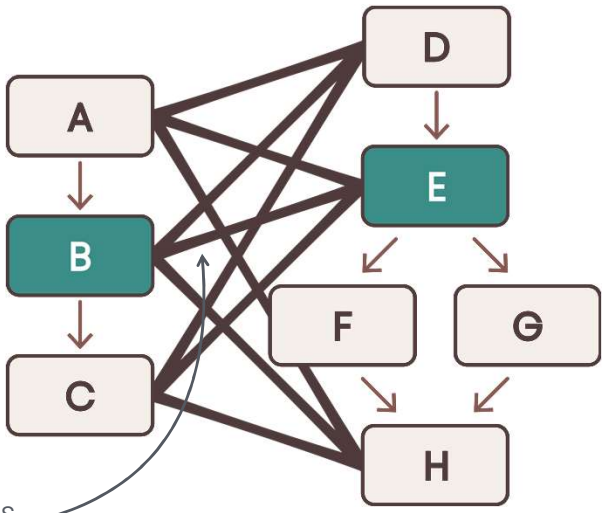


These are **synthetic lethality** interactions
(gene pairs that only cause cell death together)

What is a compensatory pathway?

Imagine that **essential DNA repair** can be performed by **two distinct** cell pathways

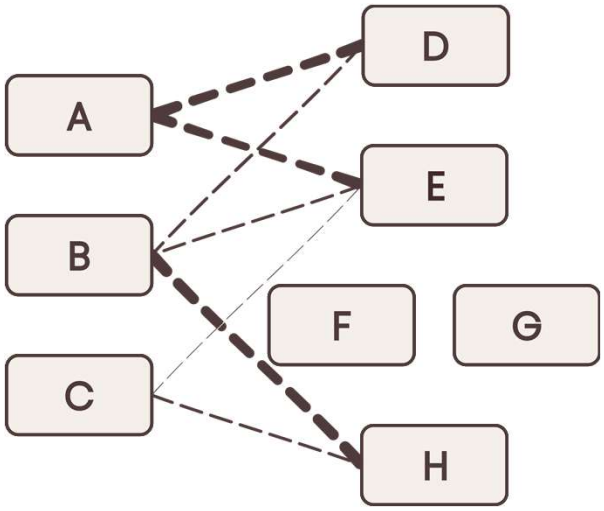
- If a gene in **one** pathway is mutated, the cell can still perform that essential function (70% capacity without E and 50% without B)
- If a gene in **both** pathways is mutated, the cell dies without being able to repair DNA



These are **synthetic lethality** interactions
(gene pairs that only cause cell death together)

Synthetic Lethality → Synthetic Sickness

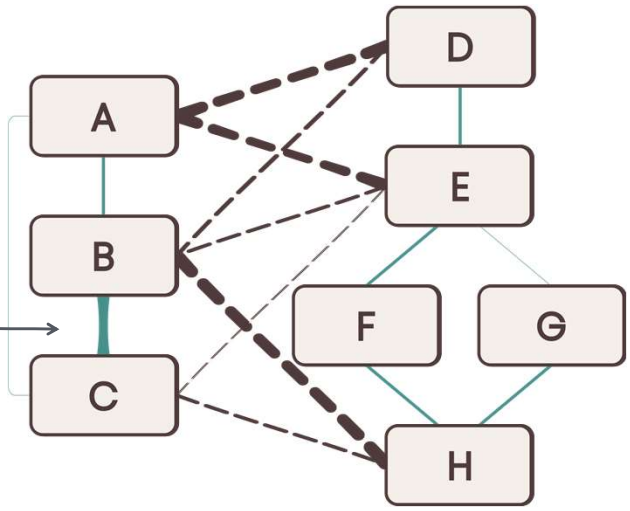
We can expand from binary interactions to *synthetic sickness*, a weighted edge where the cell is **sicker** than expected



Synthetic Lethality → Synthetic Sickness

We can expand from binary interactions to *synthetic sickness*, a weighted edge where the cell is **sicker** than expected

…and **genetic suppression**, for when the cell is healthier than expected (which are milder)

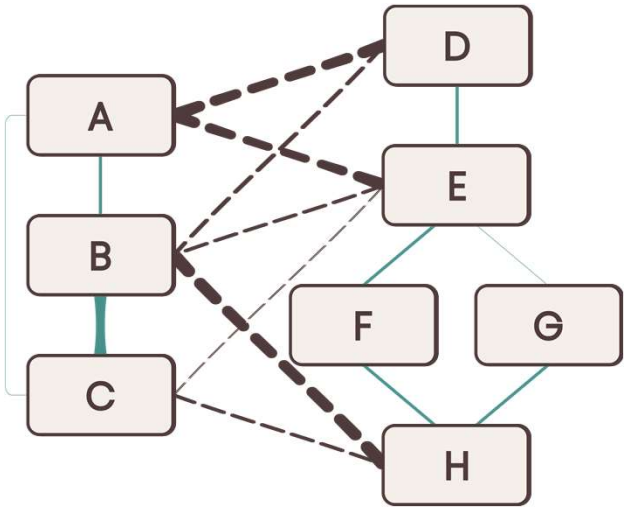


The Between Pathway Model

We call this motif a **BPM** because, if two **modules** (genes sets) represent compensatory pathways, they should have:

- Many synthetic sickness **(negative)** edges across pathways
- Many genetic suppression **(positive)** edges within pathways

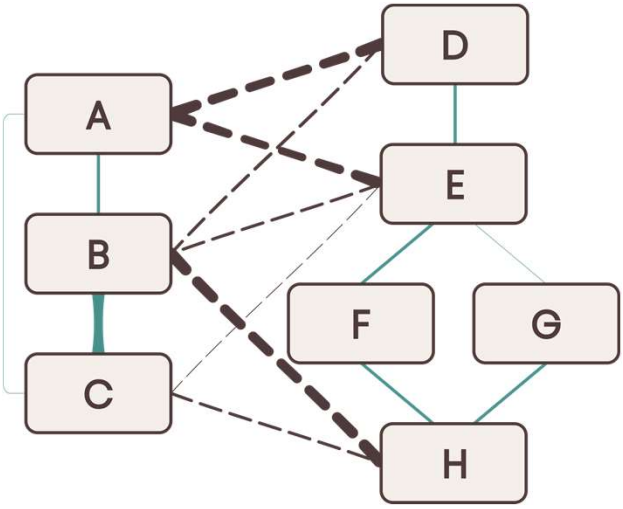
A *cohesive* BPM should contain genes in each module with **related functions**



Genetic Interaction Network → BPM

Our goal is to **identify BPMs** from a **genetic interaction network**, which is a graph where

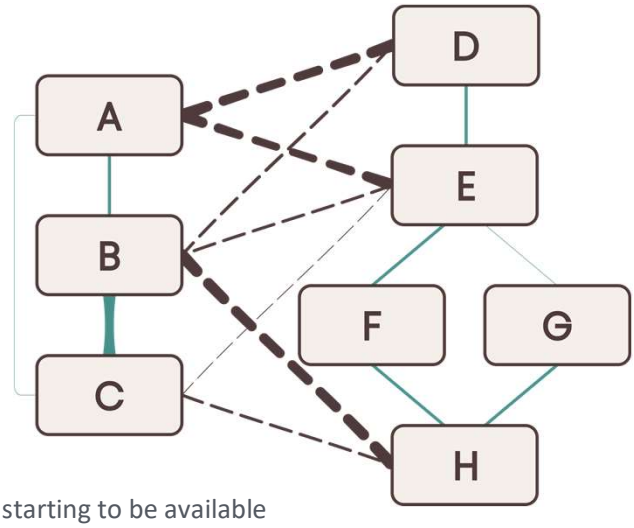
- **Nodes** are nonessential genes
- **Edges** are weighted negative/positive genetic interactions



Genetic Interaction Network → BPM

Our goal is to **identify BPMs** from a **genetic interaction network**, which is a graph where

- **Nodes** are nonessential genes
- **Edges** are weighted negative/positive genetic interactions



Currently we have high-throughput experiments on

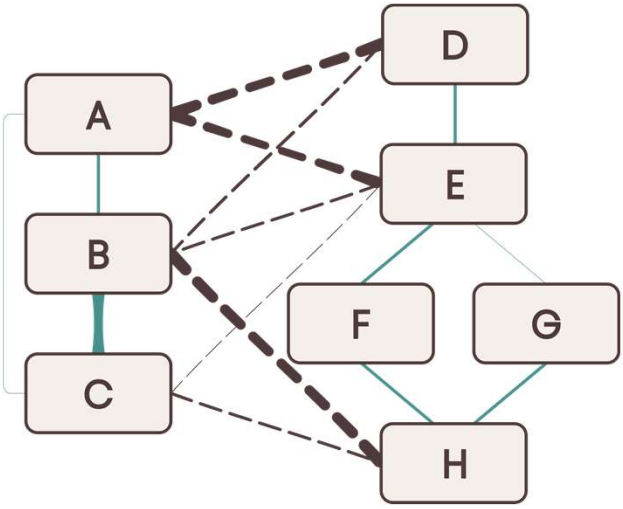
Non-essential yeast genes; other organisms more data is starting to be available

Genetic Interaction Network → BPM

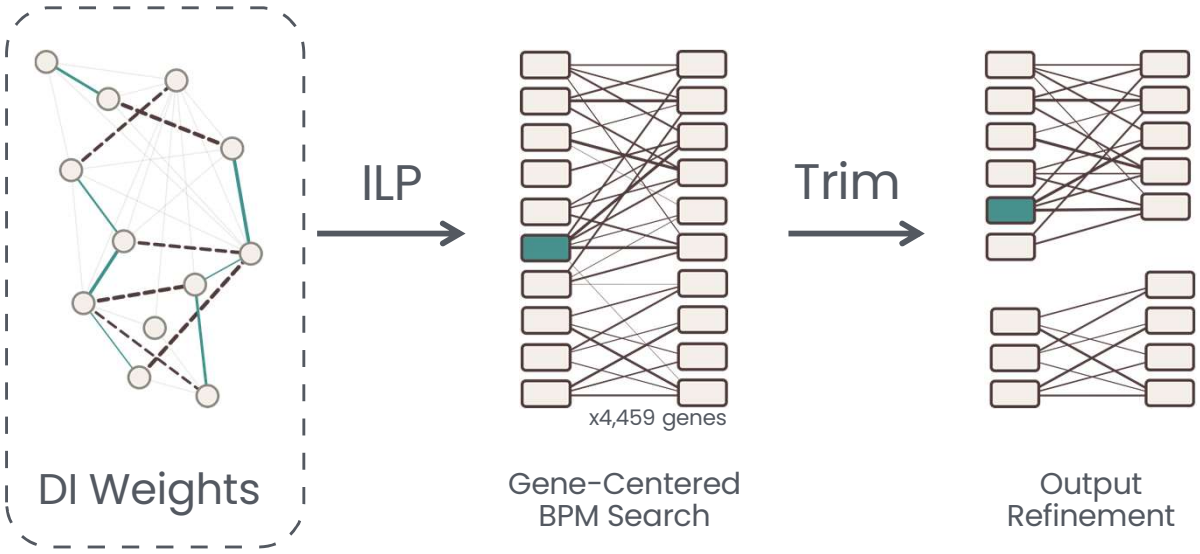
Our goal is to **identify BPMs** from a **genetic interaction network**, which is a graph where

- **Nodes** are nonessential genes
- **Edges** are weighted negative/positive genetic interactions

...but identified BPMs are only as good as the underlying network

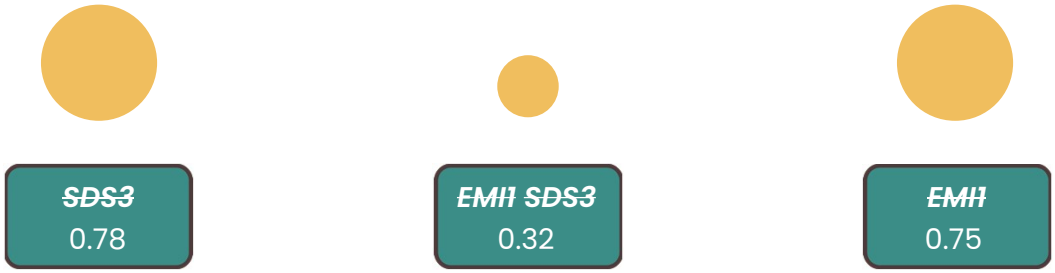


GIDEON Method Outline



Defining Genetic Interactions

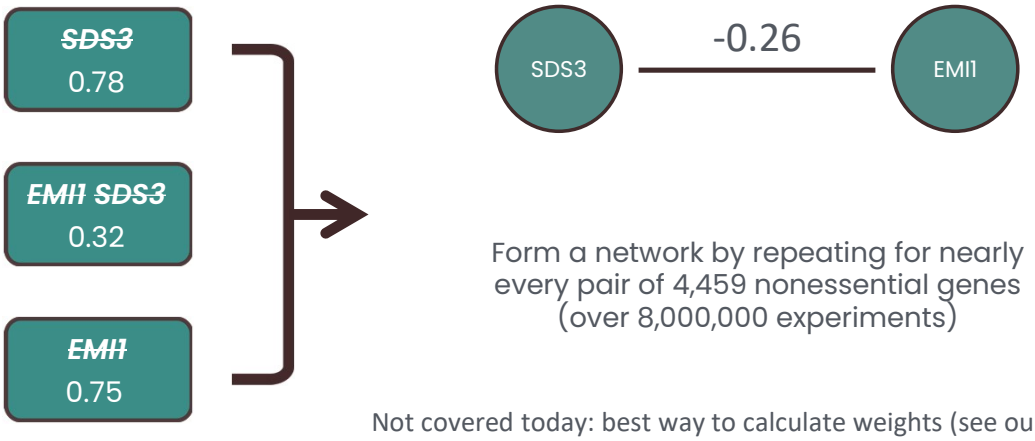
How do we quantify "sicker/healthier than expected" for the edge weights between a pair of genes? Consider *SDS3* and *EMI1*:



Compare the fitness (size) of the yeast colony under the three knockout conditions

Defining Genetic Interactions

How do we quantify "sicker/healthier than expected" for the edge weights between a pair of genes? Consider *SDS3* and *EMI1*:



Form a network by repeating for nearly every pair of 4,459 nonessential genes (over 8,000,000 experiments)

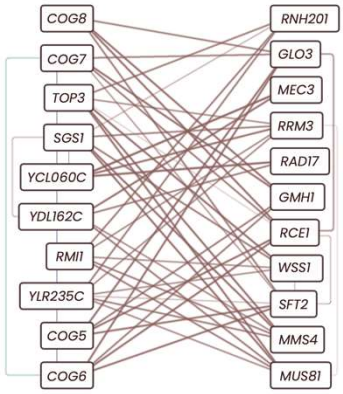
Not covered today: best way to calculate weights (see our paper)

Costanzo, Michael, et al. "A global genetic interaction network maps a wiring diagram of cellular function." *Science* 353.6306 (2016): aaf1420.

Goal: from the signed weighted genetic interaction network, discover BPMs

Two ways to measure BPM quality:

1. *Graph theoretic*
2. *Functional ENRICHMENT*



Biologists have **manually annotated** genes with GO terms that represent gene **functions** (i.e. apoptotic process)
If a module has more genes with a GO term than expected by random chance, the module is **enriched for that GO term**

Translation
(GO:0007049)

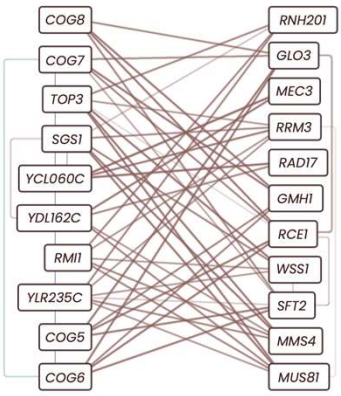
Evaluating BPMs Using Enrichment

Group BPM into **four categories:**

- **Modules enriched for the same term**
- Modules enriched for different terms
- One module enriched
- Neither modules enriched

DNA Repair (GO:0006281)

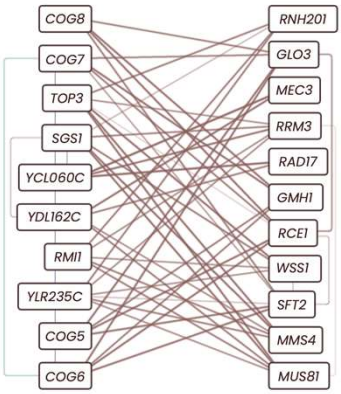
Translation (GO:0007049)



Evaluating BPMs Using Enrichment

Group BPM into **four categories:**

- **Modules enriched for the same term**
- Modules enriched for different terms
- One module enriched
- Neither modules enriched

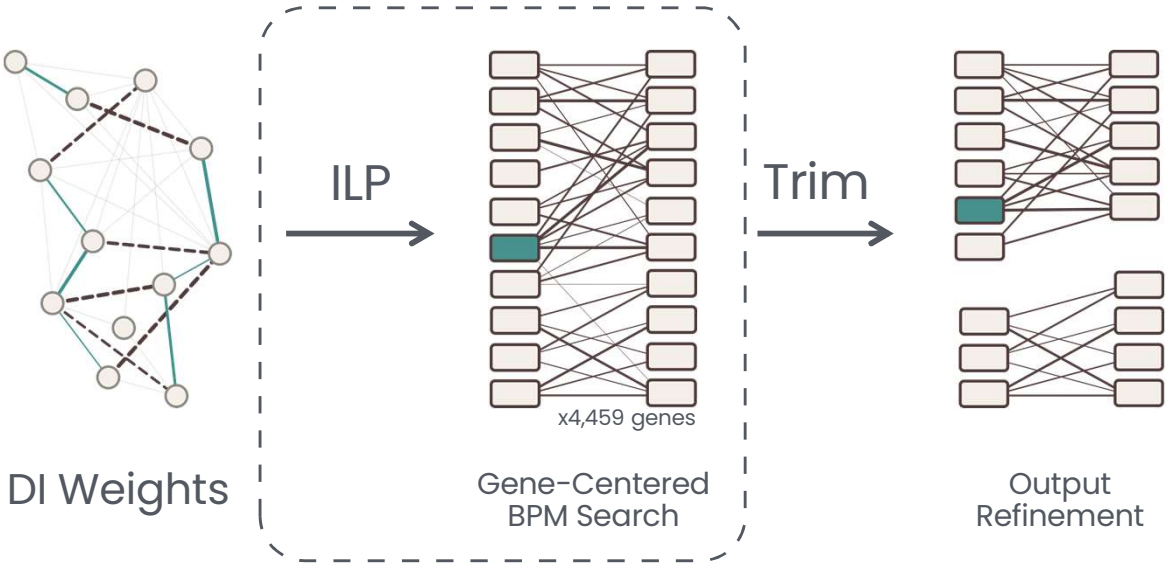


DNA Repair (GO:0006281)

Translation (GO:0007049)

Note: Annotations are **incomplete** and unenriched modules or unannotated genes may suggest **new biological functions**

GIDEON Method Outline

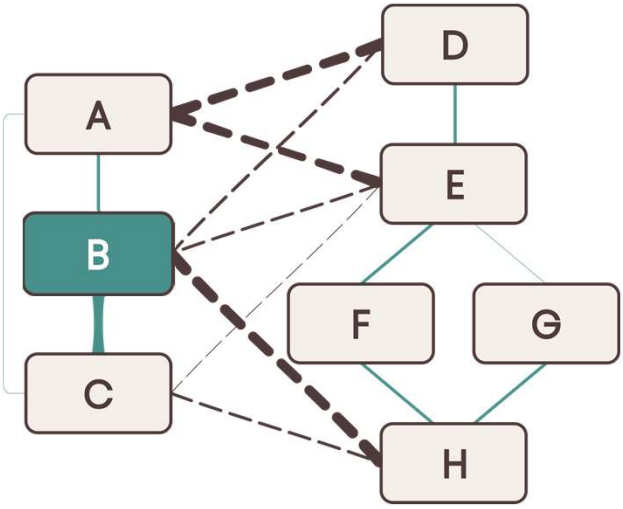


BPM Finding as Optimization

Objective: For **every gene**, maximize the interaction weight of a BPM **containing the gene** where each module has 25 genes

Interaction Weight: LocalCut approach to quantify "negative across and positive within" while accounting for BPM size

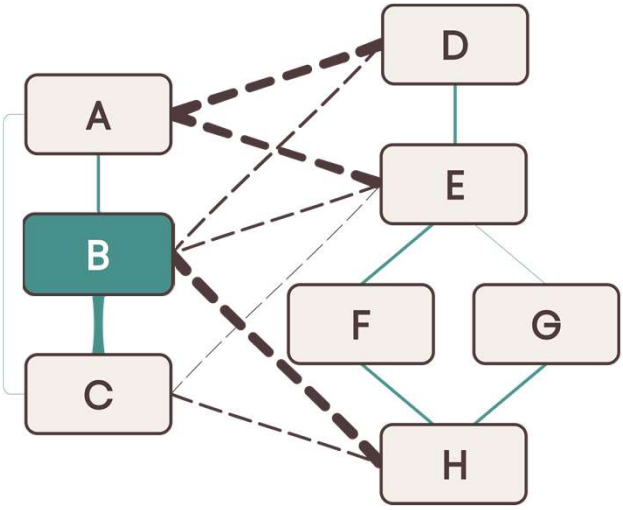
$$\frac{\sum \textit{Interactions within sets} - \sum \textit{Interactions between sets}}{\textit{Number of Genes in BPM}}$$



Defining the ILP for BPM Search

Let's view the location of a node ***n*** using a set of three binary variables:
$o_n, \ell_n, r_n \in \{0,1\}$

representing if ***n*** is in the ***left*** module, ***right*** module, or ***outside*** of the BPM



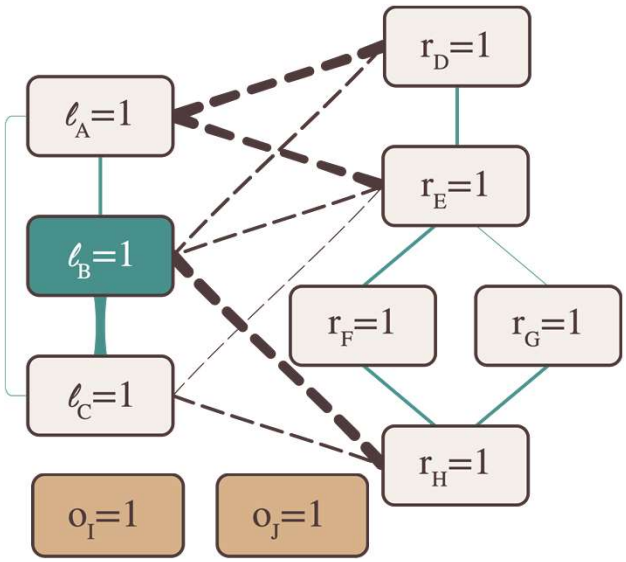
Defining the ILP for BPM Search

Let's view the location of a node *n* using a set of three binary variables:
$o_n, \ell_n, r_n \in \{0,1\}$

representing if *n* is in the *left* module, *right* module, or *outside* of the BPM

We also need the following constraints:

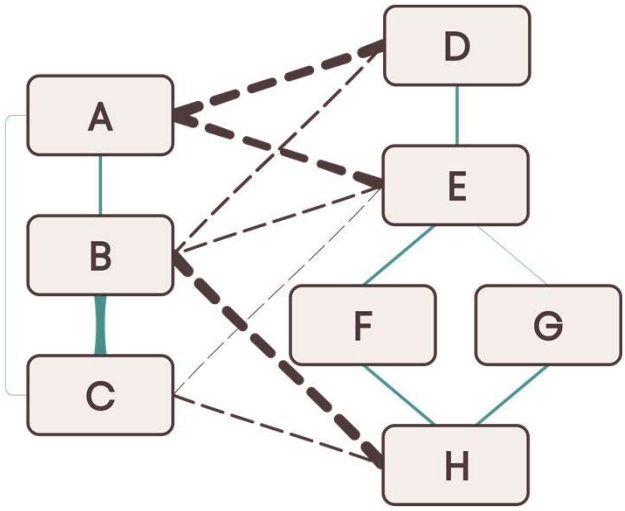
$$o_n + \ell_n + r_n = 1 \qquad \ell_B = 1$$



Defining the ILP for BPM Search

To calculate the objective function, we need binary variables for every edge:
$s_e, b_e \in \{0,1\}$

representing if *e* is in an edge within the *same* BPM modules or *between* BPM modules



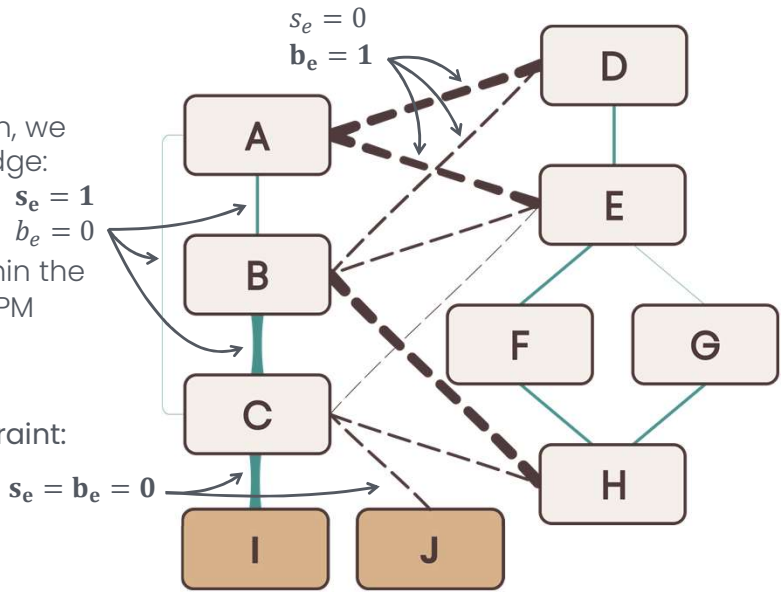
Defining the ILP for BPM Search

To calculate the objective function, we need binary variables for every edge:
$s_e, b_e \in \{0,1\}$

representing if ***e*** is in an edge within the *same* BPM modules or *between* BPM modules

We also need the following constraint:

$$s_e + b_e \leq 1$$



Edge Variables ⟷ Nodes Variables

Edge variables must correspond to the **node** variables. For an edge $e = (u, v)$:

- If the nodes are in **different modules**, then $s_e = 0$ *and* $b_e = 1$
- If **both** nodes are in the **same** module, then $s_e = 1$ *and* $b_e = 0$
- If **either** node is **outside** the bpm, then $b_e = s_e = 0$

Actual ILP Constraints

$$r_u + \ell_v - 1 \leq b_{(u,v)}$$
$$r_v + \ell_u - 1 \leq b_{(u,v)}$$

$$r_u + r_v - 1 \leq s_{(u,v)}$$
$$\ell_u + \ell_v - 1 \leq s_{(u,v)}$$

$$w_{(u,v)} + b_{(u,v)} \leq 1 - o_u$$
$$w_{(u,v)} + b_{(u,v)} \leq 1 - o_v$$

The ILP for BPM Searching

maximize interaction weight

Variables

For $n \in V$:

$r_n \in \{0,1\}$

$\ell_n \in \{0,1\}$

$o_n \in \{0,1\}$

For $e \in E$:

$b_e \in \{0,1\}$

$s_e \in \{0,1\}$

Constraints

Each module has 25 genes, each gene has 1 location, & the gene center is included

$$\sum_{e \in E_n} w_e \cdot (s_e - b_e) \leq \sum_{e \in E_g} w_e \cdot (s_e - b_e)$$

$$\sum_{e \in E_g} w_e \cdot b_e \leq \sum_{e \in E_n} w_e \cdot b_e$$

Edge variables align with node locations

The ILP for BPM Searching

$$\textit{maximize} \sum_{(u,v) \in E} w_{(u,v)} \cdot (s_{(u,v)} - b_{(u,v)})$$

Variables

For $n \in V$:

$r_n \in \{0,1\}$

$\ell_n \in \{0,1\}$

$o_n \in \{0,1\}$

For $e \in E$:

$b_e \in \{0,1\}$

$s_e \in \{0,1\}$

Constraints

$$\ell_n + r_n + o_n = 1$$

$$\sum_{n \in V} \ell_n = 25$$

$$\sum_{n \in V} r_n = 25$$

$$\ell_g = 1$$

$$\sum_{e \in E_n} w_e \cdot (s_e - b_e) \leq \sum_{e \in E_g} w_e \cdot (s_e - b_e)$$

$$\sum_{e \in E_g} w_e \cdot b_e \leq \sum_{e \in E_n} w_e \cdot b_e$$

$$r_u + \ell_v - 1 \leq b_{(u,v)}$$

$$r_v + \ell_u - 1 \leq b_{(u,v)}$$

$$r_u + r_v - 1 \leq s_{(u,v)}$$

$$\ell_u + \ell_v - 1 \leq s_{(u,v)}$$

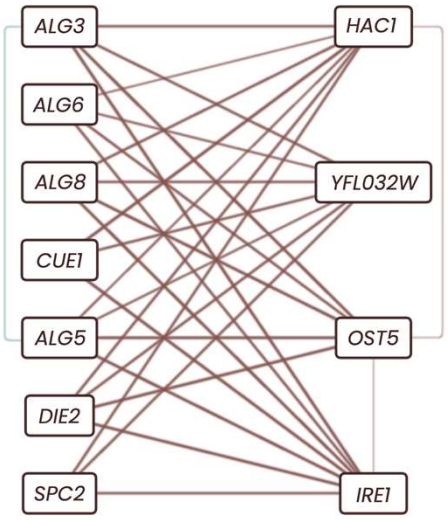
$$w_{(u,v)} + b_{(u,v)} \leq 1 - o_u$$

$$w_{(u,v)} + b_{(u,v)} \leq 1 - o_v$$

Spotlights in the Response Surface

A small set of BPMs in the search space have **huge** interaction weights

The maximum interaction weight BPM containing *IRC20* (which only has weak interactions) **disregards** its interactions

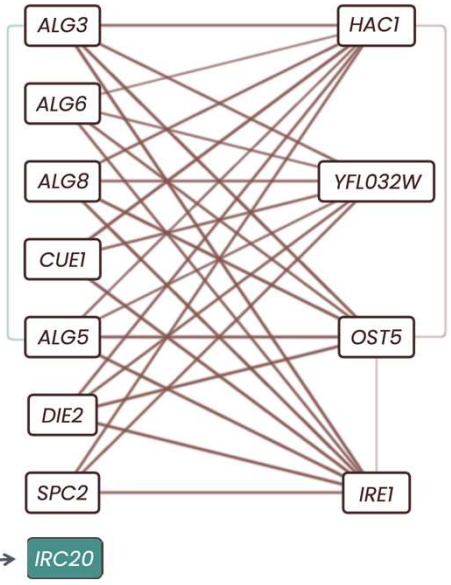


Spotlights in the Response Surface

A small set of BPMs in the search space have **huge** interaction weights

The maximum interaction weight BPM containing *IRC20* (which only has weak interactions) **disregards** its interactions

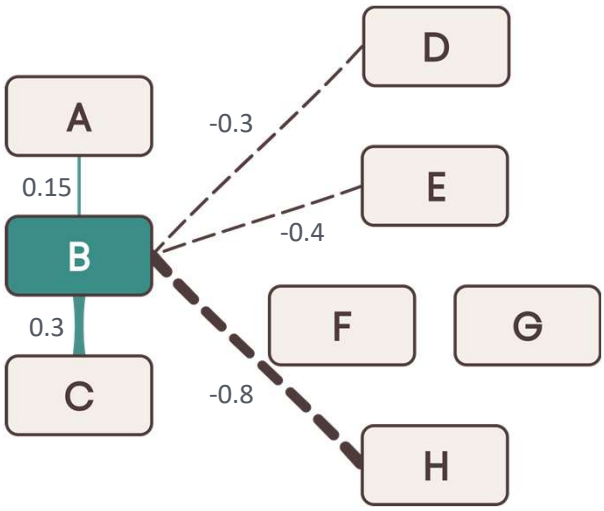
The ILP appends these genes to strong BPMs



Enforcing Local Search: *Gene* Interaction Weight

For each gene, compute the interaction weight **only using its adjacent edges**

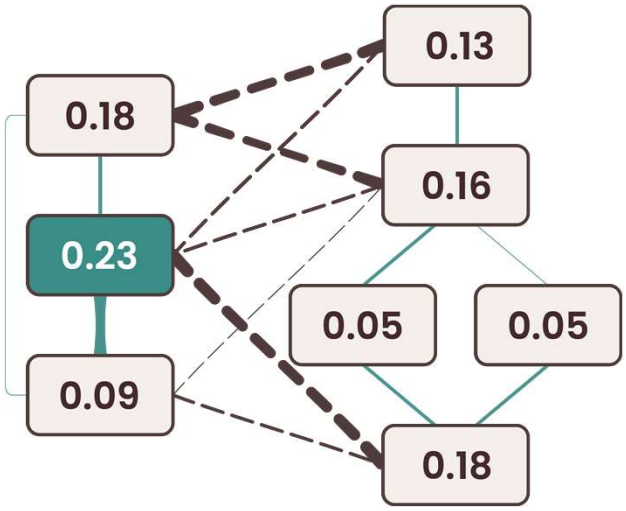
$$iw_B = \frac{(0.3 + 0.15) - (-0.8 + -0.3 + -0.4)}{8} = 0.23$$



Gene-Centered Constraints

Mathematically, *some* BPM gene must have the largest "gene interaction weight"

Constraints can ensure that **the gene used to build the BPM is that primary contributor** to the interaction weight

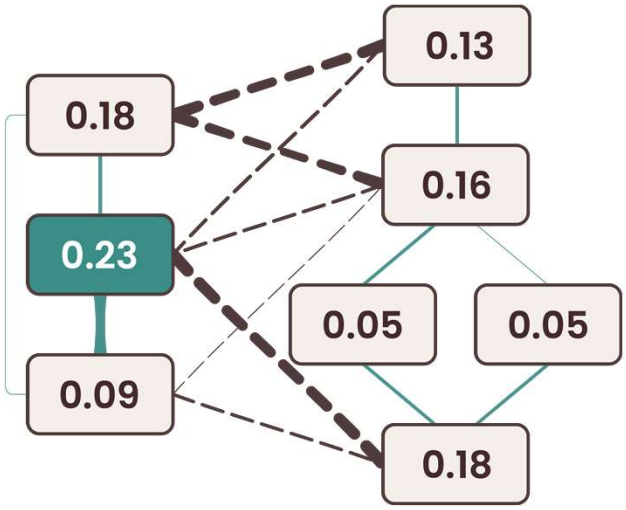


Gene-Centered Constraints

Mathematically, *some* BPM gene must have the largest "gene interaction weight"

Constraints can ensure that **the gene used to build the BPM is that primary contributor** to the interaction weight

...and this doesn't remove any BPMs from the search space!



ILP Solvers are NOT All-Powerful

Computational Limits

- Use the **No Relaxation Heuristic** (powerful for combinatorial ILPs), which pieces together sub-ILPs
- Return best feasible solution **within a computational limit** (~400 s)

GUROBI OPTIMIZATION

ILP Solvers are NOT All-Powerful

Computational Limits

- Use the **No Relaxation Heuristic** (powerful for combinatorial ILPs), which pieces together sub-ILPs
- Return best feasible solution **within a computational limit** (~400 s)

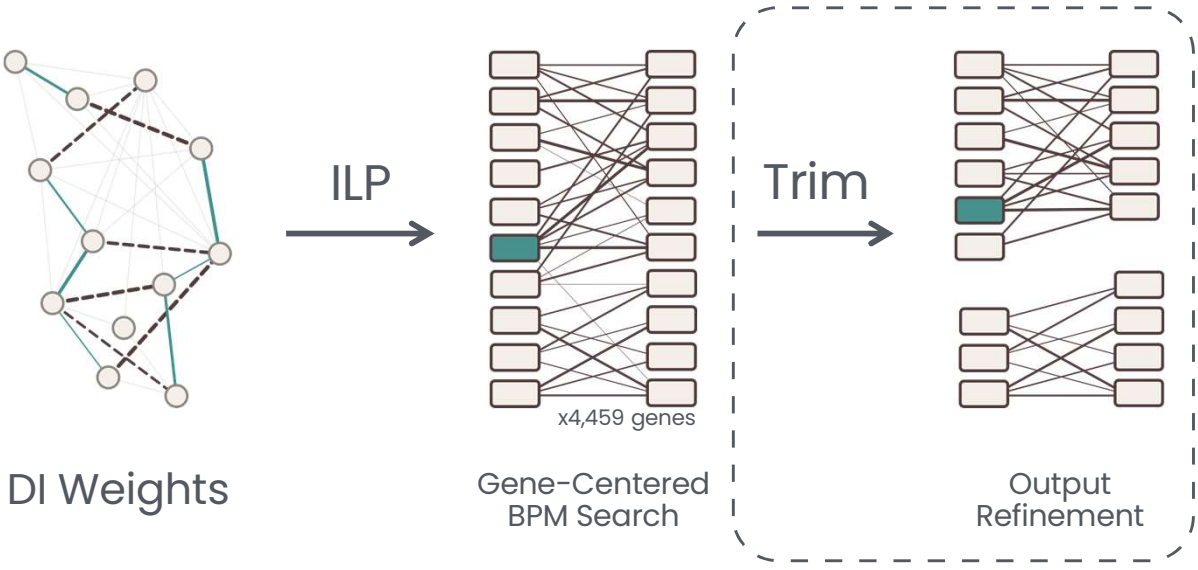
GUROBI OPTIMIZATION

ILP Formulation Matters

- Liany et. al. maximized **only the negative weight across** the BPM
- After a BPM is found, they remove all its edges from the network and rerun the ILP **(fully sequential)**


An Algorithm to Mine Therapeutic Motifs for Cancer From Networks of Genetic Interactions
Publisher: IEEE Cite This PDF
Herty Liany ; Yu Lin ; Anand Jeyasekharan ; Vaibhav Rajan All Authors

GIDEON Method Outline

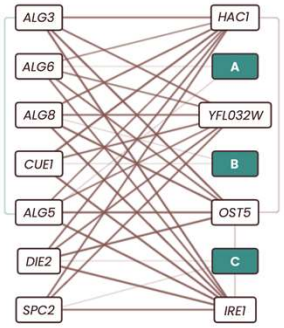


Interaction Weight is a Nonlinear Objective

maximize **interaction weight** $\neq$ *maximize* $\sum_{(u,v)\in E} w_{(u,v)} \cdot (s_{(u,v)} - b_{(u,v)})$

$$\frac{\sum \textit{Interactions within sets} - \sum \textit{Interactions between sets}}{\textit{Number of Genes in BPM}}$$

Larger graphs have **larger** edge weight sums, which LocalCut accounted for using the **BPM size divisor**



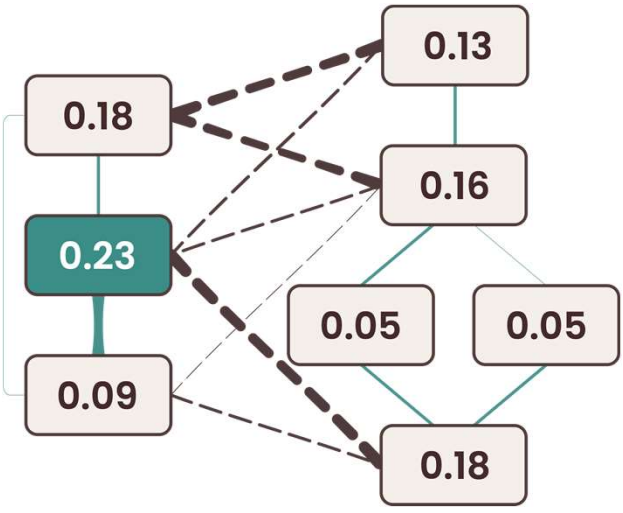
Without a divisor, BPMs are always the **maximum allowed size** because at least some of the 4,459 genes **slightly improve** the objective

Solution: 25 genes per module instead of 3 to 25

Initial Trimming Algorithm

Iteratively remove weakly included genes until **all genes** in the BPM satisfy these requirements:

1. Gene interaction weight of at least 0.015
2. At least **two** negative interactions across modules



Initial Trimming Algorithm

Iteratively remove weakly included genes until **all genes** in the BPM satisfy these requirements:

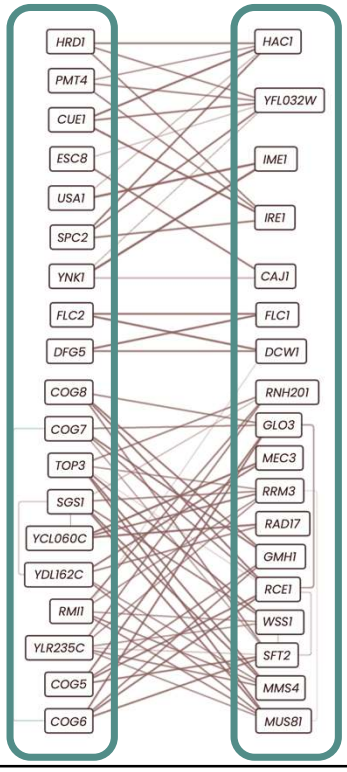
1. Gene interaction weight of at least 0.015
2. At least **two** negative interactions across modules

Poor BPMs centered on genes with weak interactions **lose all genes** and are naturally filtered out

Final Trim: Splitting Disconnected BPMs

Module 1

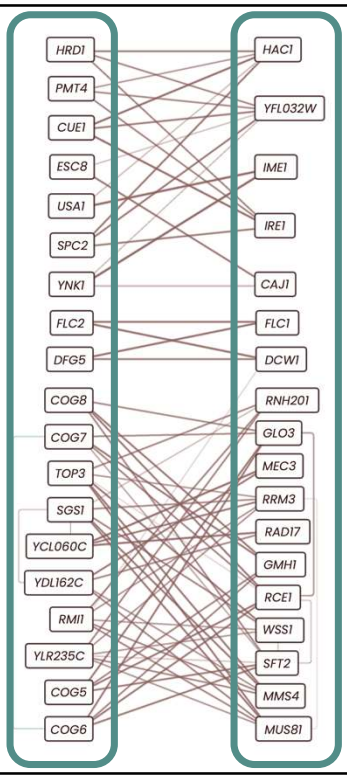
Module 2



Merged BPMs perform well under the objective function and survive the initial trimming process

Final Trim: Splitting Disconnected BPMs

Module 1

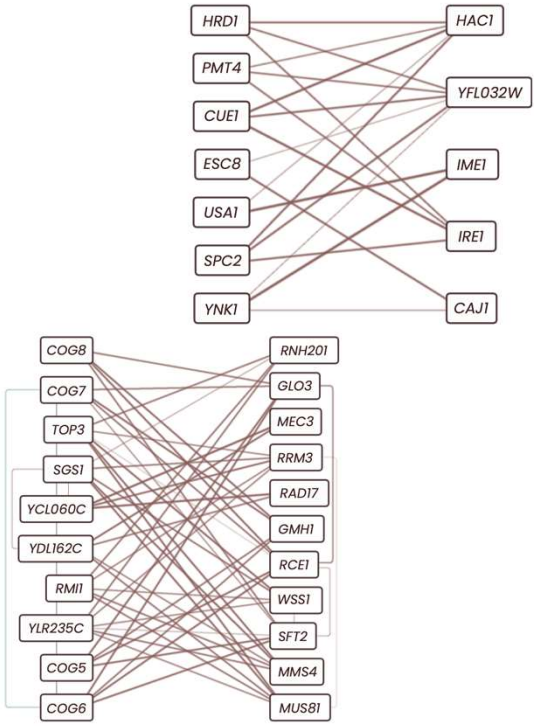


Module 2

Merged BPMs perform well under the objective function and survive the initial trimming process

Final Trim: After removing weak edges ($|\varepsilon| < 0.2$), connected components with at least 3 genes in each module become their own BPM

Final Trim: Splitting Disconnected BPMs



Merged BPMs perform well under the objective function and survive the initial trimming process

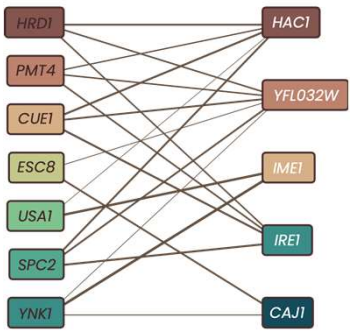
Final Trim: After removing weak edges ($|\varepsilon| < 0.2$), connected components with at least 3 genes in each module become their own BPM

LocalCut Pruning for BPM Diversity

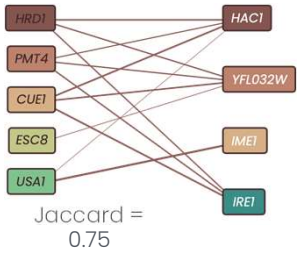
GIDEON uses the LocalCut approach to ensure the final BPM set is **diverse**

Method: In order of interaction weight, add a BPM ***A*** to the result set if **different enough** from all previously added BPMs ***B***:

$$Jaccard\ Index: \frac{A \cap B}{A \cup B} < 0.66$$

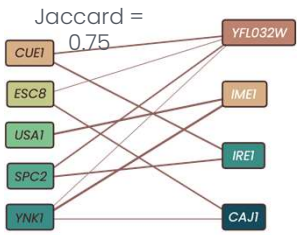


If we already have this BPM...

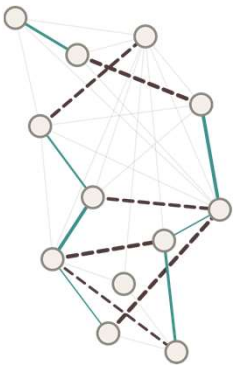


Jaccard = 0.75

...then don't add either of these

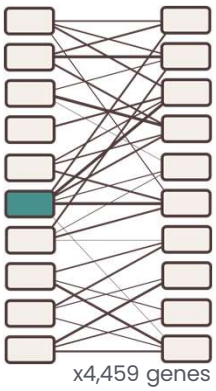


GIDEON Method Outline



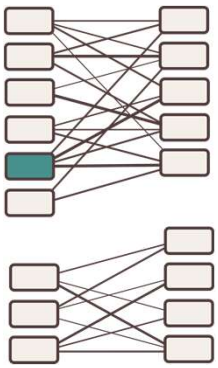
DI Weights

ILP



Gene-Centered BPM Search

Trim

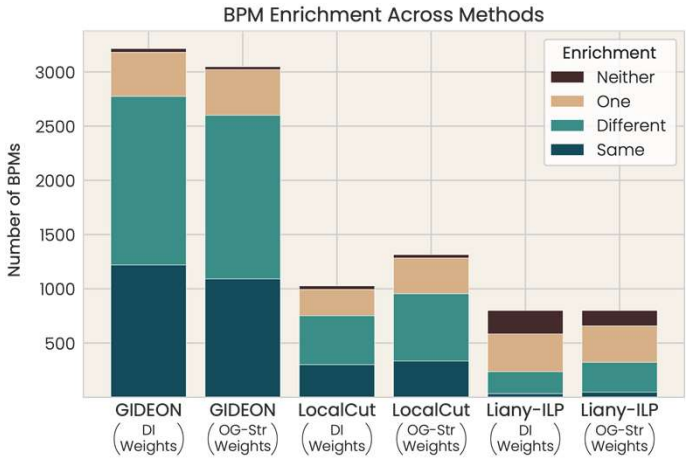


Output Refinement

GIDEON outperforms all previous methods

Nearly **triples** the number of BPMs of LocalCut, the closest competitor, with **stronger enrichment**

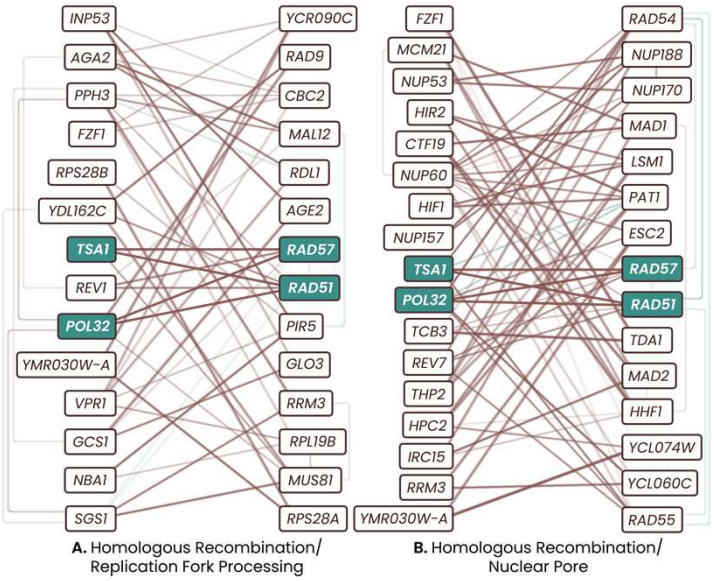
GIDEON with DI weights outperform OG-Str **without additional experimental data**



Unique BPMs from GIDEON

GIDEON identifies both BPMs on the right, highlighting the role of the overlapping genes in **shared yet distinct** processes

By removing edges from the network before searching again, Liany-ILP *cannot find BPMs with overlapping interactions*

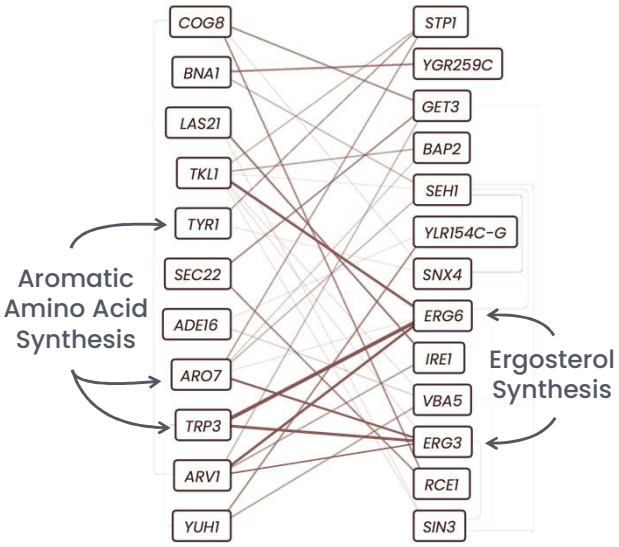


Insight Into Antifungal Drug Design

Ergosterol is a critical fungal cell membrane steroid for stress adaptation

Aromatic amino acids are phenylalanine, tyrosine, tryptophan, and histidine

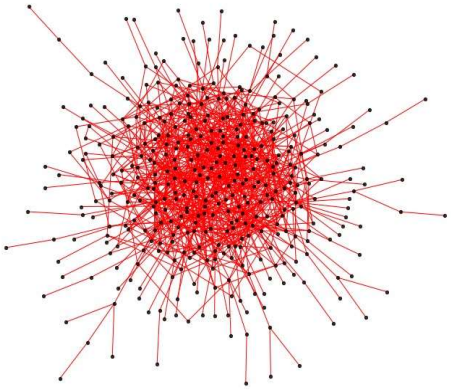
Previous work has suggested inhibitors of aromatic amino acid biosynthesis as **antifungal drugs** (via the shikimate pathway)

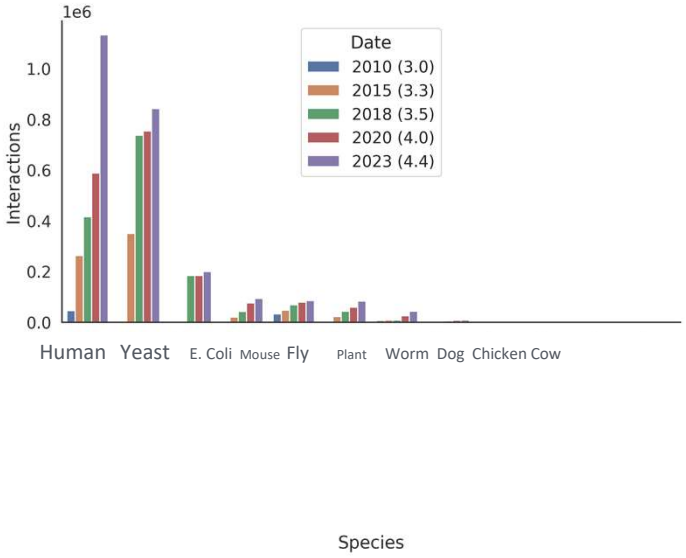


Memory-Efficient Blocked GPU

Parallel D-SCRIPT

Back to classical PPI networks: What data is available?



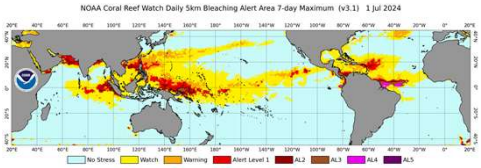


But: what about non-model organisms?

Coral Reefs Are Bleaching due to Climate Change/Heat Stress

Photo credit: Putnam Lab

Photo credit: Oregon State University



NOAA: Daily Global 5km Satellite Coral Bleaching Heat Stress Alert Area

Coral **Animal** and Symbiont Genomes are now becoming available

Family	Species	Clade	Genome Size (Mb)	Scaffold #	Scaffold N50 kb	Gene #	BUSCO Completness (genome assembly)	Reference(s)	Link
Acroporidae	Acropora acuminata	Complex	395	3,293	1,005	21,904	C:91.5 [S:89.8, D:1.7], F:1.4, M:7.1	Shinzato et al. (2021)	https://academic.oup.com/mbe/article/38/1/16/5900672
Acroporidae	Acropora awi	Complex	429	2,821	1,089	22,491	C:89.2 [S:88.2, D:1.0], F:1.8, M:9.0	Shinzato et al. (2021)	https://academic.oup.com/mbe/article/38/1/16/5900672
Acroporidae	Acropora cytherea	Complex	426	4,046	1,084	22,584	C:90.4 [S:86.5, D:3.9], F:1.9, M:7.7	Shinzato et al. (2021)	https://academic.oup.com/mbe/article/38/1/16/5900672
Acroporidae	Acropora digitifera	Complex	416	955	1,856	22,221	C:90.2 [S:88.7, D:1.5], F:1.6, M:8.2	Shinzato et al. (2021)	https://academic.oup.com/mbe/article/38/1/16/5900672
Acroporidae	Acropora echinata	Complex	401	2,002	1,917	21,554	C:87.7 [S:86.4, D:1.3], F:2.5, M:9.8	Shinzato et al. (2021)	https://academic.oup.com/mbe/article/38/1/16/5900672
Acroporidae	Acropora florida	Complex	442	6,979	751	23,237	C:88.6 [S:86.2, D:2.4], F:2.8, M:8.6	Shinzato et al. (2021)	https://academic.oup.com/mbe/article/38/1/16/5900672
Acroporidae	Acropora gemmifera	Complex	401	2,274	1,141	21,983	C:88.6 [S:87.3, D:1.3], F:3.0, M:8.4	Shinzato et al. (2021)	https://academic.oup.com/mbe/article/38/1/16/5900672
Acroporidae	Acropora hyacinthus	Complex	447	2,758	1,584	22,464	C:90.7 [S:87.8, D:2.9], F:1.5, M:7.8	Shinzato et al. (2021)	https://academic.oup.com/mbe/article/38/1/16/5900672
Acroporidae	Acropora intermedia	Complex	417	6,224	577	22,835	C:90.6 [S:87.9, D:2.7], F:1.4, M:8.0	Shinzato et al. (2021)	https://academic.oup.com/mbe/article/38/1/16/5900672
Acroporidae	Acropora microphthalma	Complex	384	4,878	1,061	22,016	C:88.6 [S:86.9, D:1.7], F:2.6, M:8.8	Shinzato et al. (2021)	https://academic.oup.com/mbe/article/38/1/16/5900672
Acroporidae	Acropora millepora	Complex	475	854	19,640	28,188	C:86 [S:79.7,D:6.3],F:5.8,M:8.2	Fuller et al. (2020)	https://www.science.org/doi/10.1126/science.abo4674
Acroporidae	Acropora millepora	Complex	387	3,869	494	23,710	C:91.3 [S:89.5, D:1.8], F:1.5, M:7.2	Ying et al. (2019)	https://academic.oup.com/gbe/article/11/5/1374/5480306
Acroporidae	Acropora muricata	Complex	421	6,861	575	23,103	C:88.3 [S:85.8, D:2.5], F:3.2, M:8.5	Shinzato et al. (2021)	https://academic.oup.com/mbe/article/38/1/16/5900672
Acroporidae	Acropora nasta	Complex	416	4,717	1,051	22,545	C:90.5 [S:86.4, D:4.1], F:1.7, M:7.8	Shinzato et al. (2021)	https://academic.oup.com/mbe/article/38/1/16/5900672
Acroporidae	Acropora selago	Complex	393	5,816	657	22,616	C:87.6 [S:85.5, D:2.1], F:3.5, M:8.9	Shinzato et al. (2021)	https://academic.oup.com/mbe/article/38/1/16/5900672
Acroporidae	Acropora tenuis	Complex	403	1,538	1,166	22,802	C:90.5 [S:89.4, D:1.1], F:1.8, M:7.7	Shinzato et al. (2021)	https://academic.oup.com/mbe/article/38/1/16/5900672
Acroporidae	Acropora yongei	Complex	438	1,010	3,033	23,044	C:88.6 [S:86.2, D:2.4], F:2.8, M:8.6	Shinzato et al. (2021)	https://academic.oup.com/mbe/article/38/1/16/5900672
Acroporidae	Astreopora myriophthalma	Complex	373	1,149	1,634	28,712	C:89.3 [S:87.7, D:1.6], F:2.6, M:8.1	Shinzato et al. (2021)	https://academic.oup.com/mbe/article/38/1/16/5900672
Acroporidae	Montipora cactus	Complex	653	4,925	899	21,983	C:88.5 [S:86.7, D:1.8], F:2.7, M:8.8	Shinzato et al. (2021)	https://academic.oup.com/mbe/article/38/1/16/5900672
Acroporidae	Montipora capitata	Complex	885	3043	540	63,229	C: 90.6, F: 1.2	Shumaker et al. (2019)	https://doi.org/10.1038/s41598-019-39274-3
Acroporidae	Montipora capitata	Complex	614	27,865	185	36,691	C:82.1 [S:80.9, D:1.2], F:6.6, M:11.3	Helmkampf et al. (2019)	https://academic.oup.com/gbe/article/11/7/2045/5523851?login=true
Acroporidae	Montipora effloresc ans	Complex	643	5,162	1,112	21,370	C:86.4 [S:84.8, D:1.6], F:2.7, M:10.9	Shinzato et al. (2021)	https://academic.oup.com/mbe/article/38/1/16/5900672
Poritidae	Porites lutea	Complex	552	2975	660.7	31,126	98% CEGMA	Robbins et al. (2019)	https://doi.org/10.1038/s41564-019-0532-4
Poritidae	Porites rus	Complex	470	14982	137	39,453	C:72.8 %[S:70.7%,D:2.1%],F:9.7%,M:17.5%	Cellis et al. (2018)	https://www.ncbi.nlm.nih.gov/genome/?term=txid2625931[orgn]
Agariciidae	Pachyseris speciosa	Robust	984	2368	766.6	39,160	C: 90.5 F: 5.1%	Bongaerts et al (2021)	https://www.sciencedirect.com/science/article/pii/S0960982221003675
Euphyllidae	Galaxea fascicularis	Robust	334	11269	87	22,418	C: 68 F:8 M:23	Ying et al. (2018)	https://doi.org/10.1186/s13059-018-1552-8
Fungiidae	Fungia spp	Robust	606	7424	323	38,209	C: 65, F: 12, M:22	Ying et al. (2018)	https://doi.org/10.1186/s13059-018-1552-8
Merulinidae	Orbicella annularis	Robust	NA	NA	NA	NA	NA	Prada et al. (2016)	https://www.sciencedirect.com/science/article/pii/S0960982216311233?via%3Dihub
Merulinidae	Orbicella faveolata	Robust	486	1,932	1,162	25,916	C:85.8 [S:83.2, D:2.6], F:4.5, M:9.7	Prada et al. (2016)	https://www.sciencedirect.com/science/article/pii/S0960982216311233?via%3Dihub
Merulinidae	Orbicella franksi	Robust	NA	NA	NA	NA	NA	Prada et al. (2016)	https://www.sciencedirect.com/science/article/pii/S0960982216311233?via%3Dihub
Merulinidae	Goniastrea aspera	Robust	764	5396	518	35,901	C: 68 F:8 M:23	Ying et al. (2018)	https://doi.org/10.1186/s13059-018-1552-8
Montastraeidae	Montastrea cavernosa	Robust	448	5161	343	43,577	C: 81.0, F: 9.8	Rippe et al (2021)	https://onlinelibrary.wiley.com/doi/10.1111/mec.15931
Pocilloporidae	Pocillopora acuta	Robust	404	550	1,096	36,704	C:95.6% [S:80.3%, D:15.3%], F:1.6%, M:2.8%	Stephens et al (2021)	https://www.biorxiv.org/content/10.1101/2021.11.21.469467v2
Pocilloporidae	Pocillopora damicornis	Robust	234	4,392	326	19,935	C:89.2 [S:88.7, D:0.5], F:2.4, M:8.4	Cunning et al. (2018)	https://doi.org/10.1038/s41598-018-34459-8
Pocilloporidae	Pocillopora verrucosa	Robust	380	18268	333	27,439	C:92.22 %[S:89.05%,D:3.17%],F:1.84%,M:5.93%	Buitrago-López et al 2020	https://doi.org/10.1093/gbe/evaa184
Pocilloporidae	Stylophora pistillata	Robust	400	5,687	457	24,833	C:88.3 [S:86.8, D:1.5], F:3.3, M:8.4	Voolstra et al. (2017)	https://doi.org/10.1038/s41598-017-17484-x

From Cowen and Putnam, Annual Reviews of Biomedical Science, 2022

Species	Clade	Genome size (Mb)	Gene #	URL
Pocillopora damnicornis	Robust	234	19,935	https://doi.org/10.1038/s41598-018-34459-8

Coral Animal and **Symbiont** Genomes are now becoming available

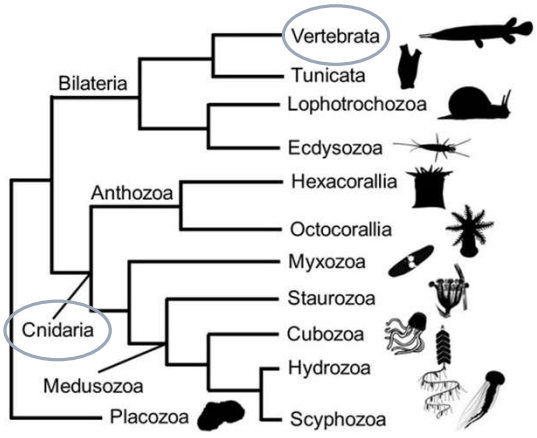
Family	Species	Genome Size (Mb)	Gene #	Reference(s)
Symbiodiniaceae	Breviolum minutum	616	41,925	Shoguchi et al. (2013)
Symbiodiniaceae	Cladocopium sp. C92	705	65,832	Shoguchi et al. (2018)
Symbiodiniaceae	Cladocopium goreaui	1028	35,913	Liu et al. (2018)
Symbiodiniaceae	Durusdinium trenchii	670	30,054	Shoguchi et al (2021)
Symbiodiniaceae	Fugacium kawagutii (V3)	937	45,192	Aranda et al. (2016)
Symbiodiniaceae	Symbiodinium fitti	602	37,621	Reich et al (2021)
Symbiodiniaceae	Symbiodinium microadriaticum	808	49,109	Aranda et al. (2016)
Symbiodiniaceae	S. microadriaticum CassKB8	1120	42,652	Gonzalez-Pech et al. (2021)
Symbiodiniaceae	S. microadriaticum 04-503SCI.03	1053	38,462	Gonzalez-Pech et al. (2021)
Symbiodiniaceae	S. tridacnidorum CCMP2592*	1287	45,474	Gonzalez-Pech et al. (2021)
Symbiodiniaceae	S. linucheae CCMP2456	915	32,053	Gonzalez-Pech et al. (2021)
Symbiodiniaceae	S. necroappetens CCMP2469	1007	35,672	Gonzalez-Pech et al. (2021)
Symbiodiniaceae	S. natans CCMP2548*	740	35,270	Gonzalez-Pech et al. (2021)
Symbiodiniaceae	S. pilosum CCMP2461	1994	23,437	Gonzalez-Pech et al. (2021)
Symbiodiniaceae	Symbiodinium tridacnidorum	767	69,018	Shoguchi et al. (2018)

From Cowen and Putnam, Annual Reviews of Biomedical Science, 2022

Family	Species	Genome size (Mb)	Gene #	Reference
Symbiodiniaceae	Cladocopium goreaui	1028	35,913	Liu et al (2018(

Corals very evolutionarily distant so protein sequences very different

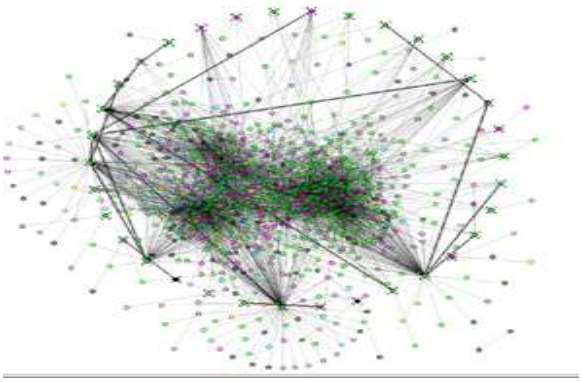
For many coral proteins, you won't be able to find functionally annotated proteins that are good sequence matches..

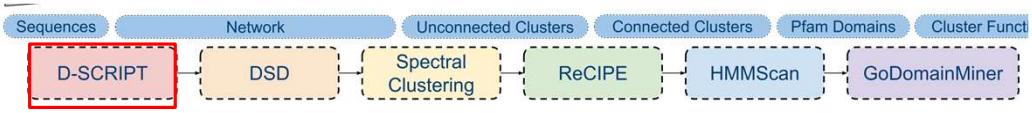


From: Vandepas et al, Biomolecules **2023**, *13*(5), 777;

Genome wide prediction of protein-protein interactions is the first step in the Philharmonic Pipeline

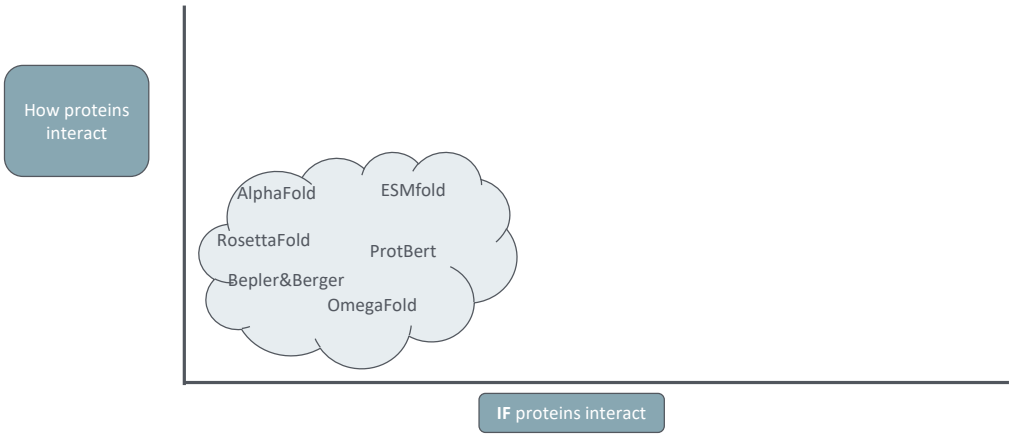
Organism Complete Proteome



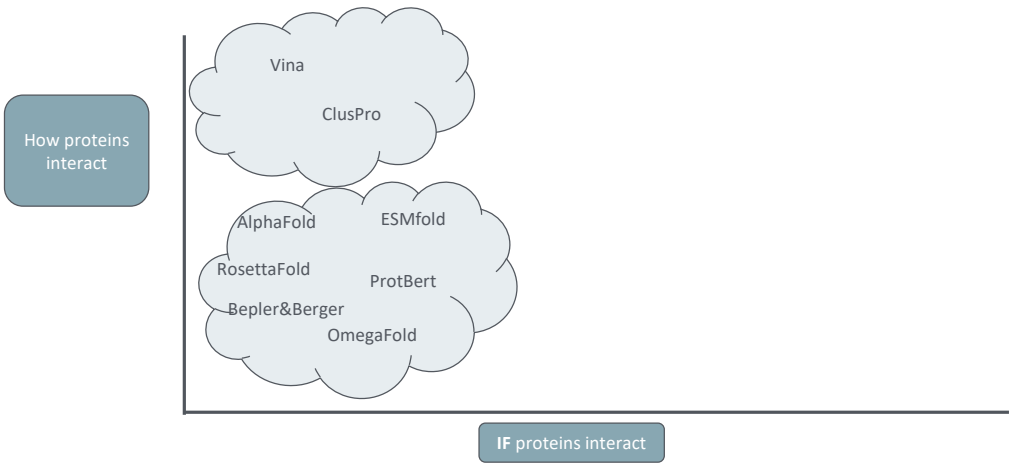


Leveraging language models for **lightweight** and **computationally efficient** PPI predictions

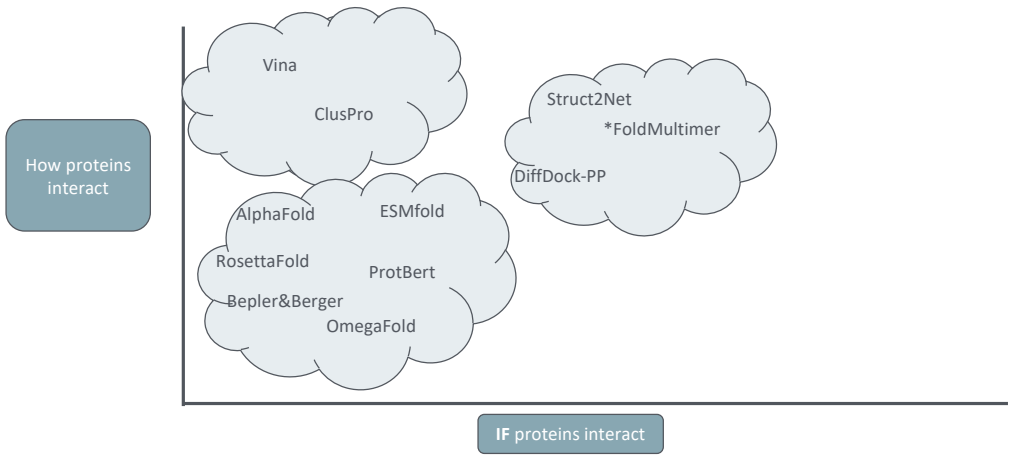
Leveraging language models for **lightweight** and **computationally efficient** PPI predictions



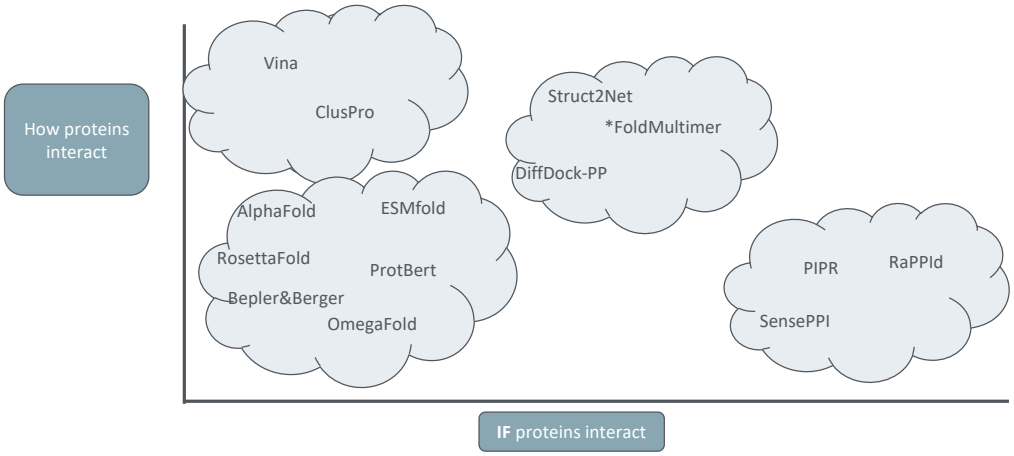
Leveraging language models for **lightweight** and **computationally efficient** PPI predictions



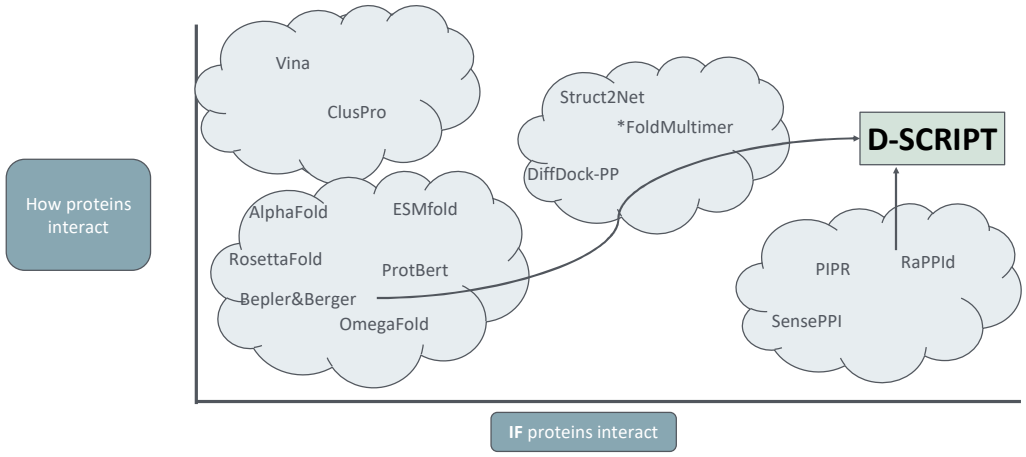
Leveraging language models for **lightweight** and **computationally efficient** PPI predictions



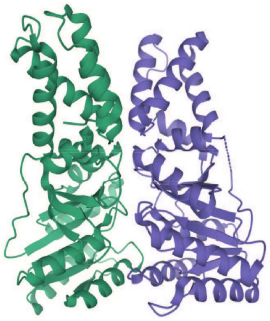
Leveraging language models for **lightweight** and **computationally efficient** PPI predictions

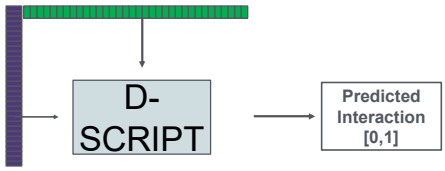


Leveraging language models for **lightweight** and **computationally efficient** PPI predictions



D-Script Is Trained on the Human Interactome, and Learns to Predict PPIs in A Different Species





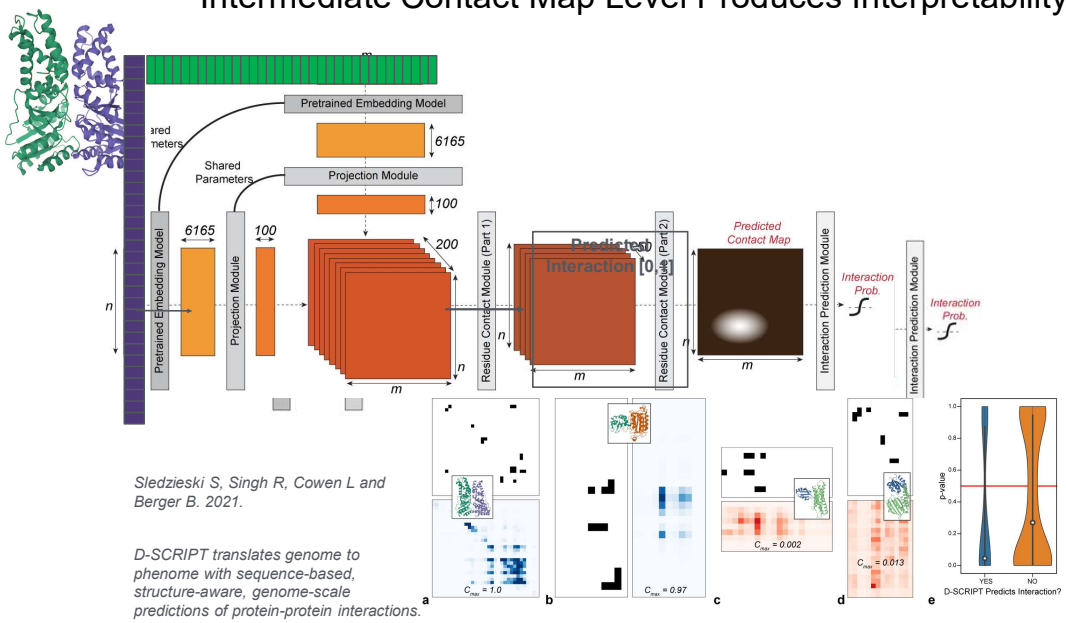
Sledzieski et al., 2021

Important to use a method that is lightweight

(only needs to predict the binary)

D-Script Can Predict PPIs Across Species:

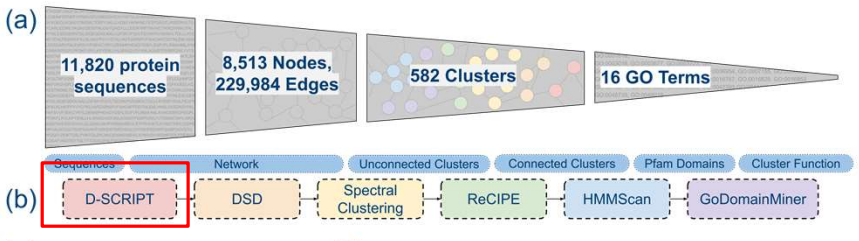
Intermediate Contact Map Level Produces Interpretability



Sledzieski S, Singh R, Cowen L and Berger B. 2021.

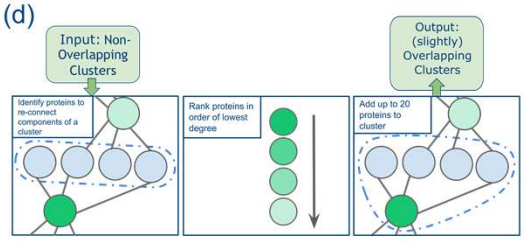
D-SCRIPT translates genome to phenome with sequence-based, structure-aware, genome-scale predictions of protein-protein interactions.

D-SCRIPT is the first step in the PHILHARMONIC Pipeline

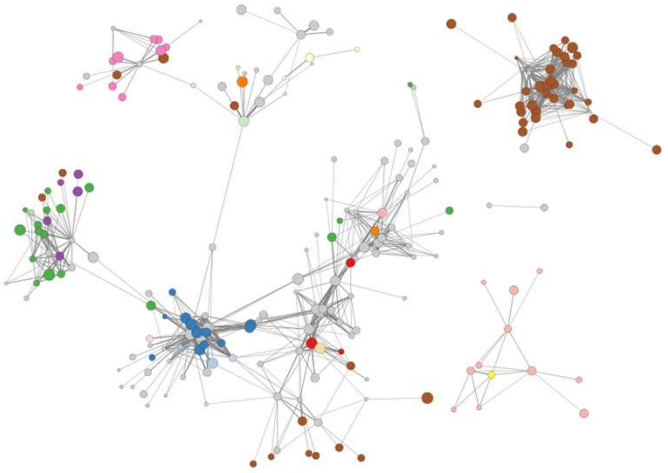


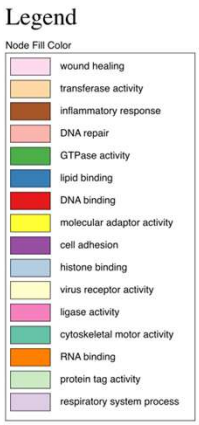
(c)

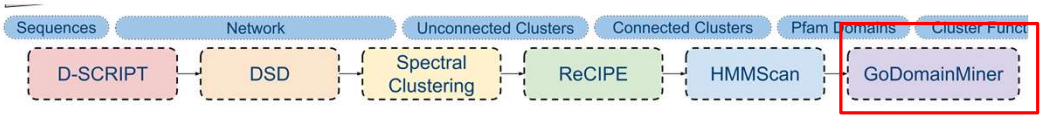
	Pocillopora damicornis	*Symbiodinium goreaui*
Proteins	8,513	7,928
Edges	229,984	394,137
Med. Degree	22	34
Avg. Degree	54.031	99.429
Sparsity	0.00317	0.00627



Pfam to GO mapping is used to assign functional labels to proteins (and clusters)







Leveraging language models for **lightweight** and **computationally efficient** PPI predictions

- Language models trained on a large corpus of proteins learn **general structural features** of proteins that can be used to predict interaction
- Maintain **speed** and ease of use of sequence-based deep learning methods

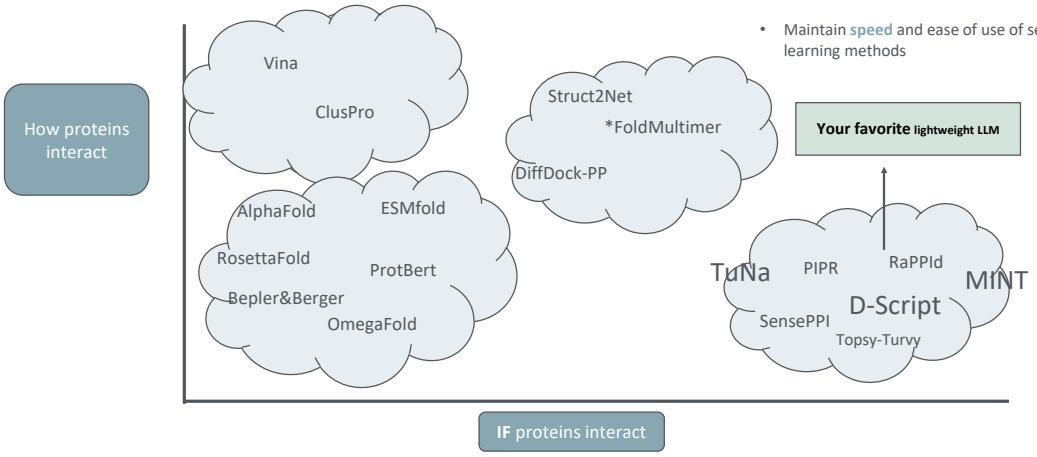


Figure 1. (A) The original serial version of D-SCRIPT iterates through each pair of candidate proteins. (B) This ...

Running D-SCRIPT genome-wide:

If we load all proteins embeddings into memory:

Very expensive!!

On the other hand, if we fetch on demand very slow!!

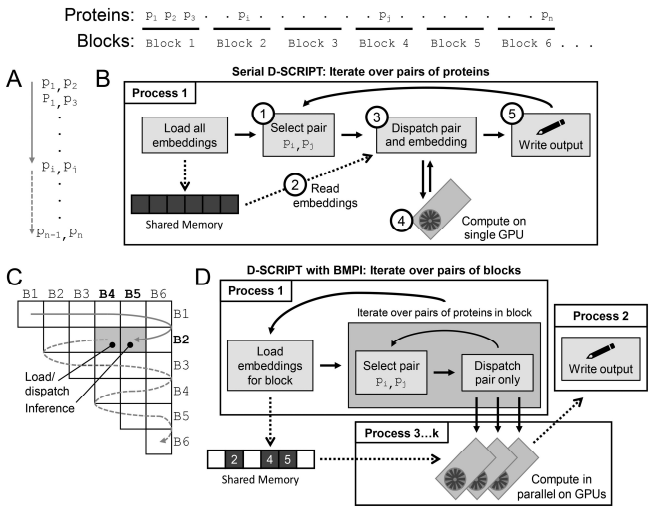


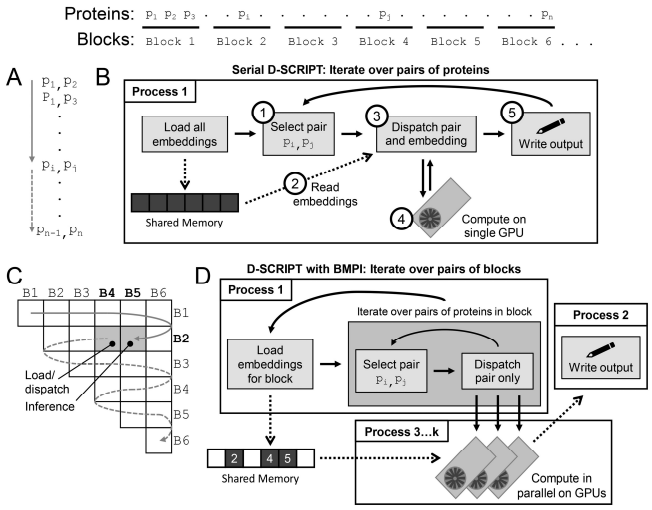
Figure 1. (A) The original serial version of D-SCRIPT iterates through each pair of candidate proteins. (B) This ...

Running D-SCRIPT genome-wide:

If we load all proteins embeddings into memory:

Very expensive!!

On the other hand, if we fetch on demand very slow!!



Solution: divide proteins into blocks;

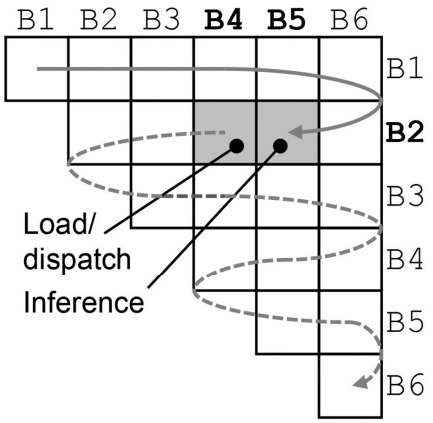
Solve combinatorial problem of

Choosing pairs so that blocks do not

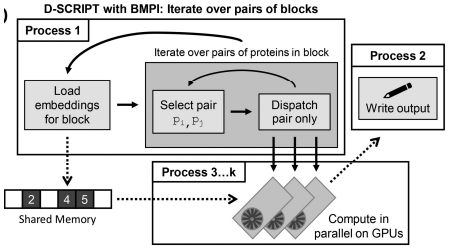
Have to be frequently loaded/unloaded

From memory

Example with 6 Blocks




```
if i = 0 and blocks > 1 then Embeds3 ← LOAD(1) end if
for j ∈ [i + 1, blocks) do                                  ▷ Incrementing inner iteration
    if j = i + 1 then
        Embeds2 ← Embeds3                                   ▷ Reuse previously-loaded data
        if j ≠ blocks − 2 then Embeds3 = ∅ end if
    else
        Embeds2 ← LOAD(j, flag : (cp − 1))
    end if
    cp ← cp + 1
    SUBMIT_PAIR(i, j, Embeds1, Embeds2, flag : cp)
end for
else
    if i = blocks − 1 then  ▷ Special case to reuse loaded embeddings in the final iteration
        Embeds1 ← Embeds2
    else if i = blocks − 2 then
        Embeds1 ← Embeds3
    else
        Embeds1 ← LOAD(i, flag : (cp − 1))
    end if
    for j ∈ [blocks − 1, i + 2) do                          ▷ Decrementing inner iteration
        cp ← cp + 1
        SUBMIT_PAIR(i, j, Embeds1, Embeds2, flag : cp)
        Embeds2 ← LOAD(j − 1, flag : (cp − 1))
    end for
    if i < blocks − 2 then                                  ▷ Special cases to re-use loaded embeddings
        SUBMIT_PAIR(i, i + 2, Embeds1, Embeds2)
        cp ← cp + 1
        Embeds3 ← Embeds2
    end if
    SUBMIT_SELF(i, Embeds1)                                 ▷ Self-pair submission (i = j) using loaded embeddings
```



Benchmarking Results

Dataset: DMel50: Take complete Fly proteome; first isoform from each protein; remove proteins

With more than 1000 residues; randomly sample half. Gives: 6282 proteins (19 728 621 pairs)

Server config: 2 AMD EPYC 7502 processors, 2TB of RAM, and 8 NVIDIA A6000 GPUs

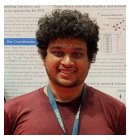
Benchmarking Results

Dataset: DMel50: Take complete Fly proteome; first isoform from each protein; remove proteins

With more than 1000 residues; randomly sample half. Gives: 6282 proteins (19 728 621 pairs)

Server config: 2 AMD EPYC 7502 processors, 2TB of RAM, and 8 NVIDIA A6000 GPUs

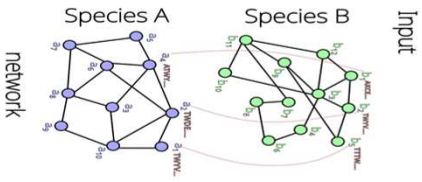
Method	GPUs	Blocks	Time (Hours)	Est. Memory (GB)
Original D-SCRIPT	1	N/A	52.06	61.10
Blocked D-SCRIPT	8	1	6.42	87.97
Blocked D-SCRIPT	1	64	50.91	8.39
Blocked D-SCRIPT	4	64	12.82	19.48
Blocked D-SCRIPT	8	64	6.62	34.55


Kapil Devkota


Anselm Blumer


Xiaozhe Hu

Devkota, K., Blumer, A., Hu, X., & Cowen, L. (2024). Approximate IsoRank for Scalable and Functionally Meaningful Cross-Species Alignments of Protein Interaction Networks. *Journal of Computational Biology*, *31*(10), 990-1007.



Tufts UNIVERSITY

NIH NSF

Garcia, J.J., Kevin, M.Y., Freudenreich, C.H. and Cowen, L.J., 2026. A Novel ILP Framework to Identify Compensatory Pathways in Genetic Interaction Networks with GIDEON. *bioRxiv*.

NIH-R01GM163241


Jocelyn Garcia


Catherine Freudenreich

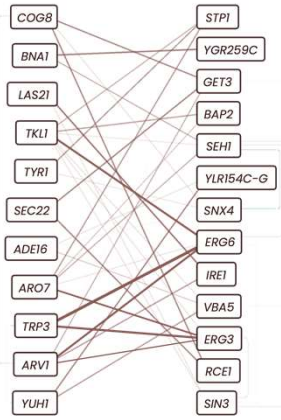

Kevin Yu

NSF-2149871

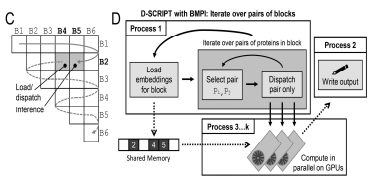
Tufts UNIVERSITY

NIH R01GM163241

NSF



Schäffer, D. E., Sledzieski, S., Cowen, L., & Berger, B. (2025). Memory-efficient, accelerated protein interaction inference with blocked, multi-GPU D-SCRIPT. *Bioinformatics*, *41*(10), btaf564.



NSF-2141064

NIH 1R35GM14186

Daniel Schäffer Sam Sledzieski Bonnie Berger

Tufts UNIVERSITY

NIH NSF

NIH-R01GM163241

Kapil Devkota Anselm Blumer Xiaozhe Hu

Jocelyn Garcia Catherine Freudenreich Kevin Yu

NIH-R01GM163241

NSF-2149871

Daniel Schäffer Sam Sledzieski Bonnie Berger

Tufts UNIVERSITY

NIH NSF

NIH-R01GM163241

THANK YOU!!!