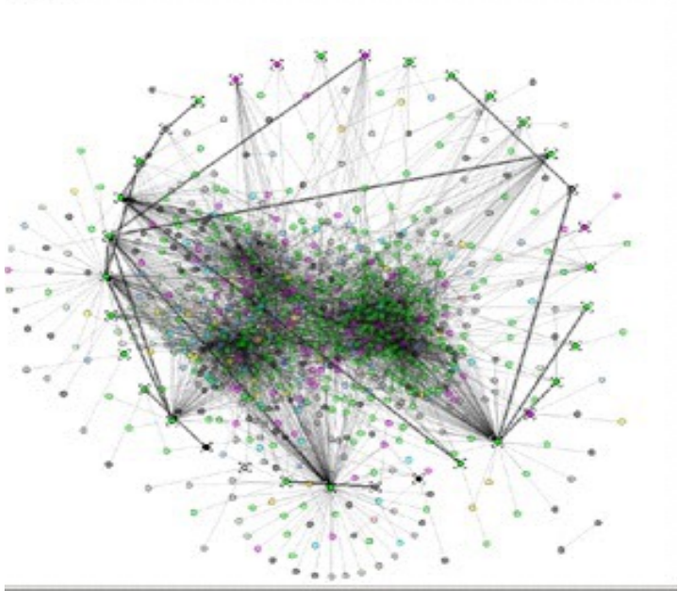
A Double Spectral Approach to Disease Module Identification

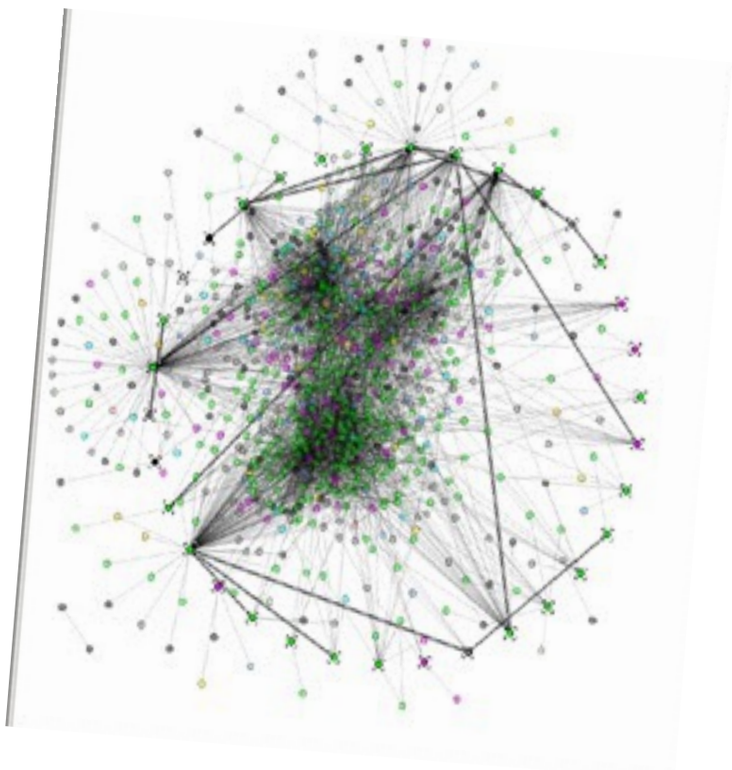
Jake Crawford, Junyuan Lin, Xiaozhe Hu, Benjamin Hescott,
Donna Slonim, and **Lenore Cowen**

Tufts University

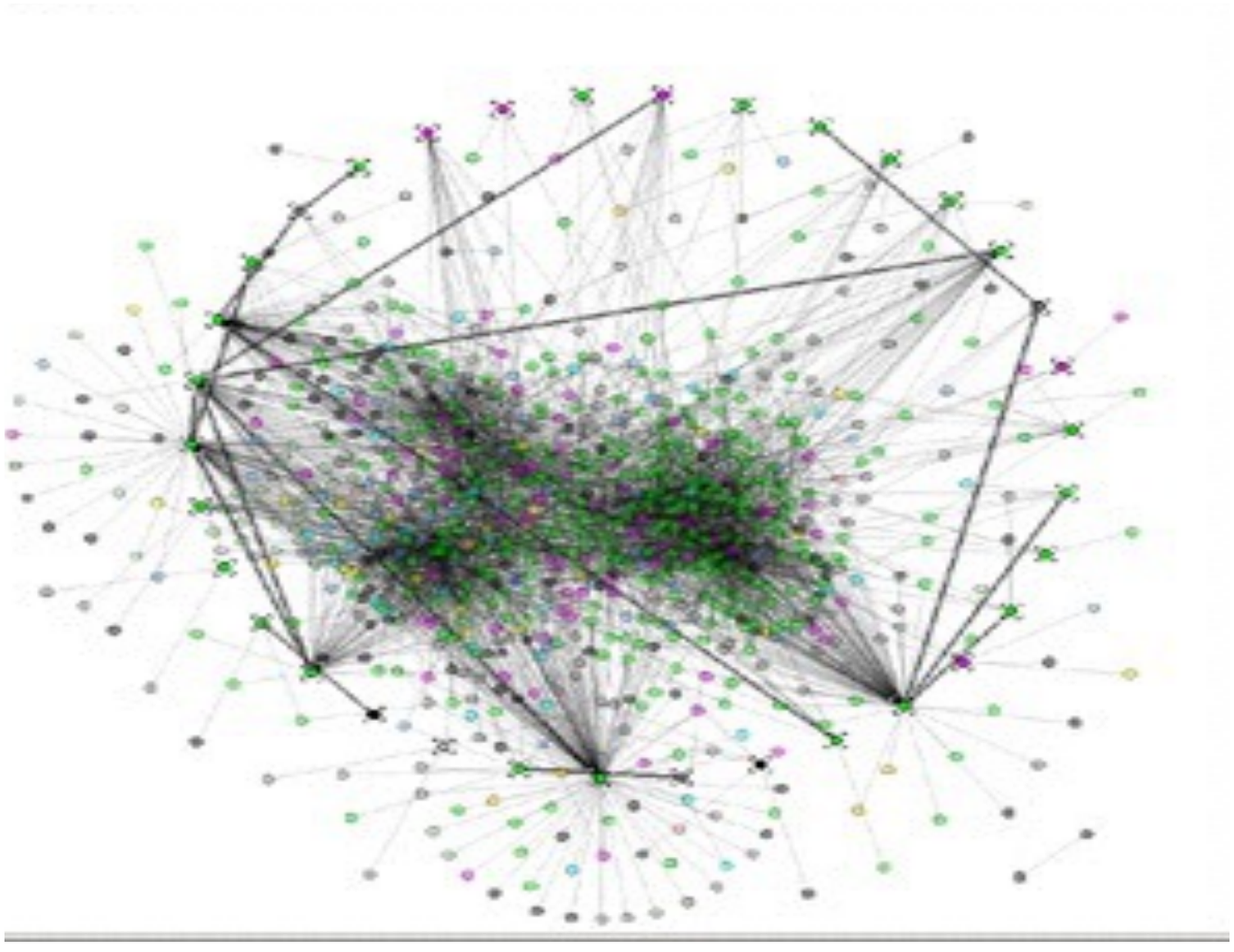
Our Goal

- Identify module structure in biological networks, enriched for a collection of disease gene sets.

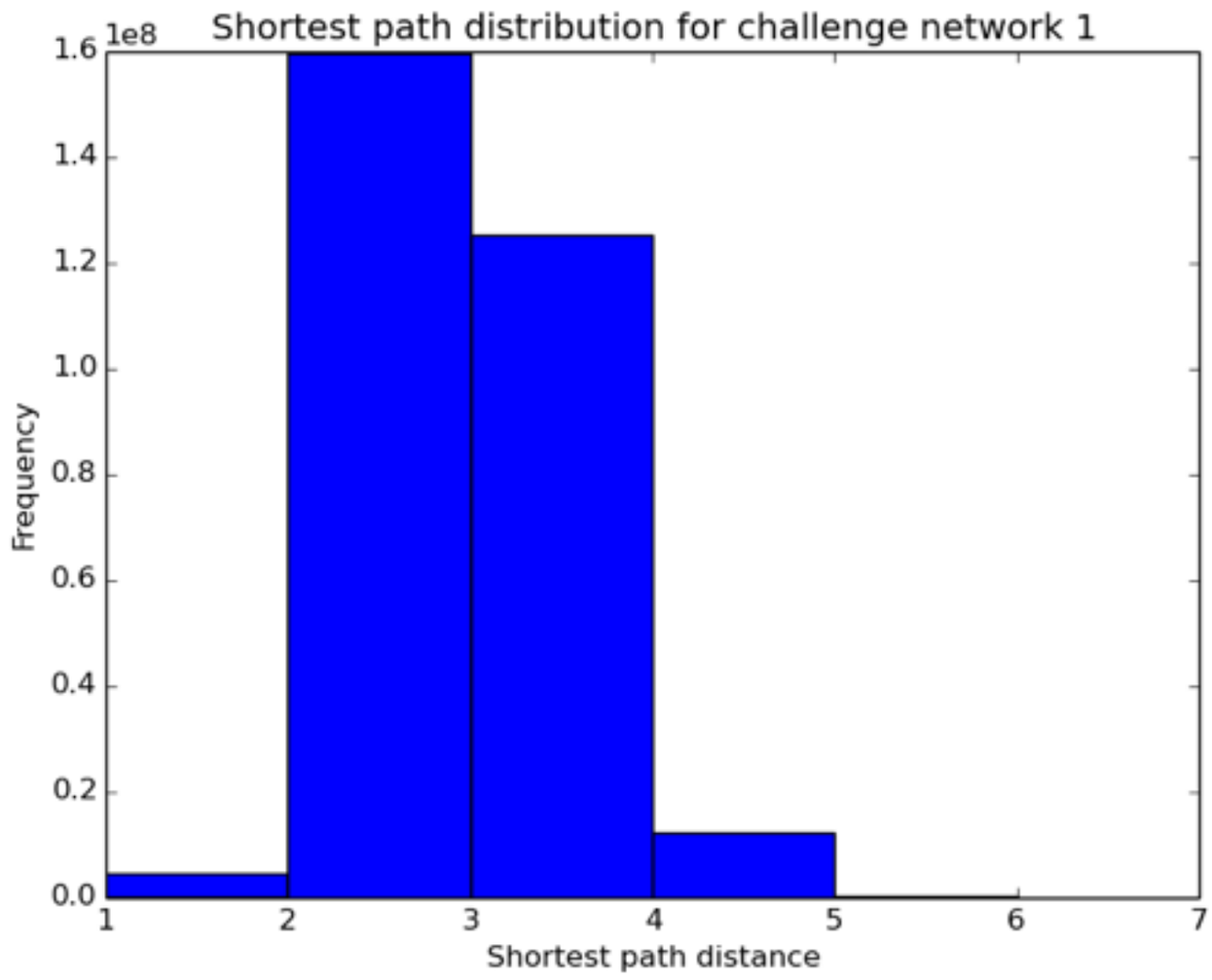




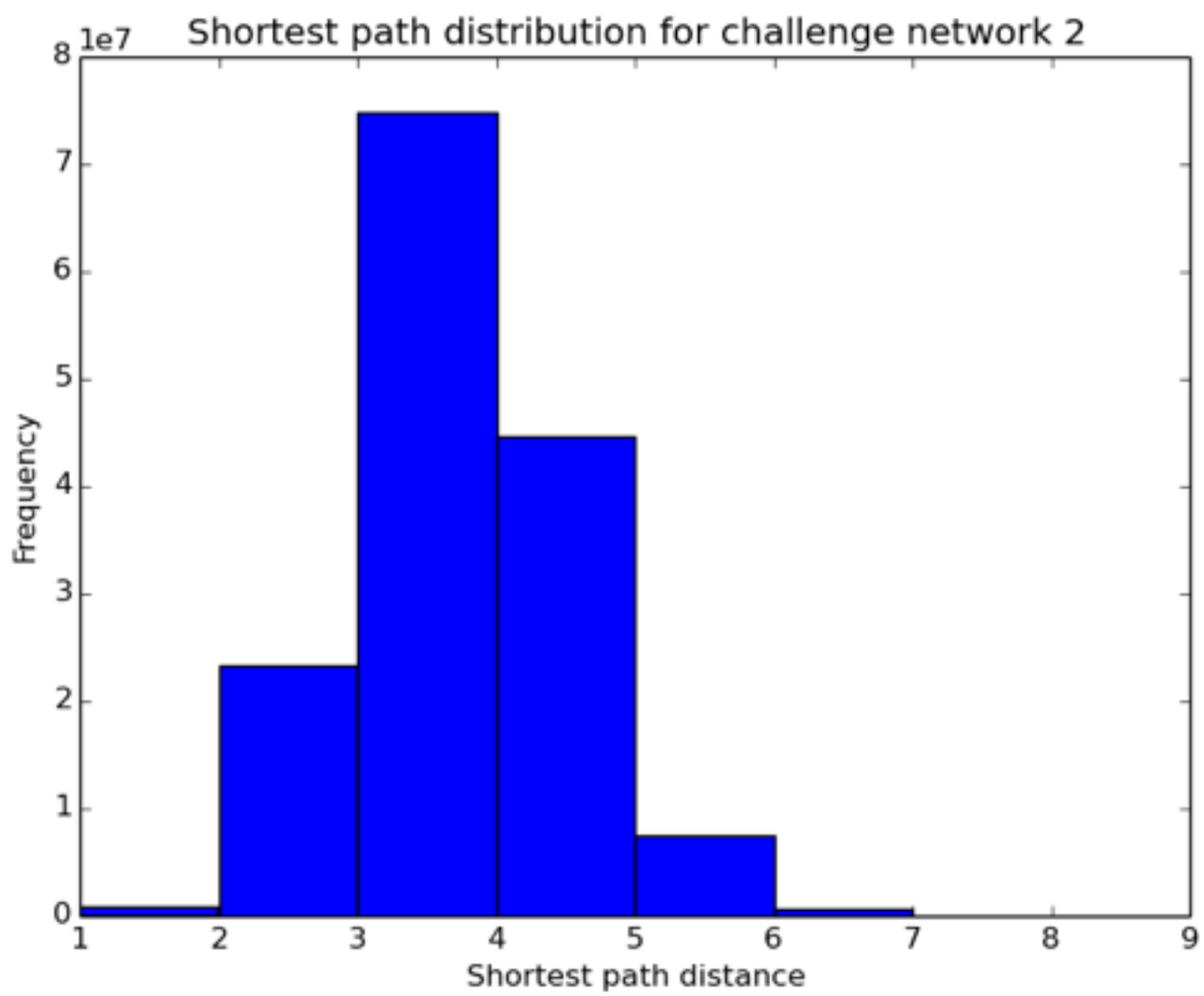
PPI network: issue with defining local neighborhoods: “small-world” network where everything is nearby!



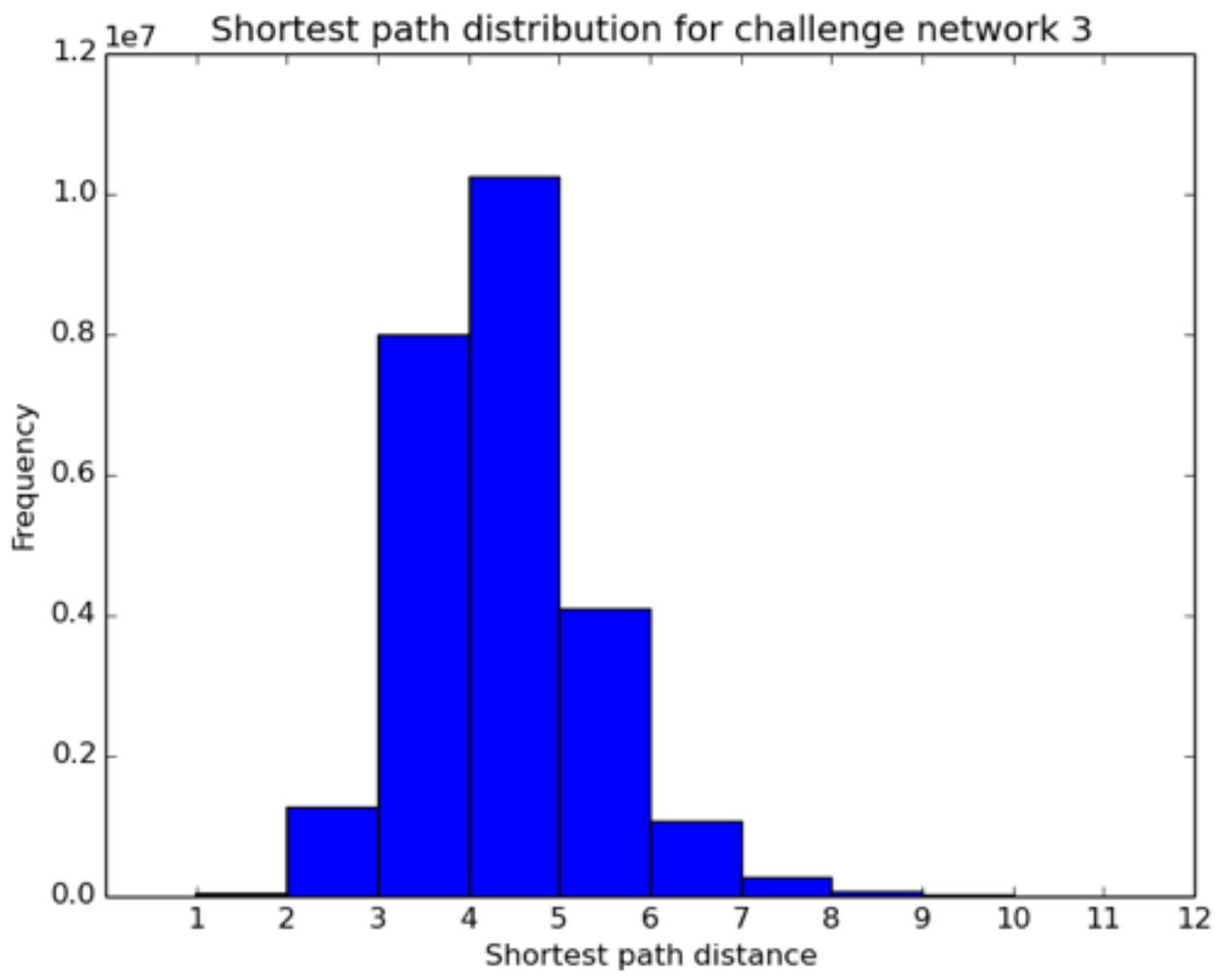
Dream Network 1



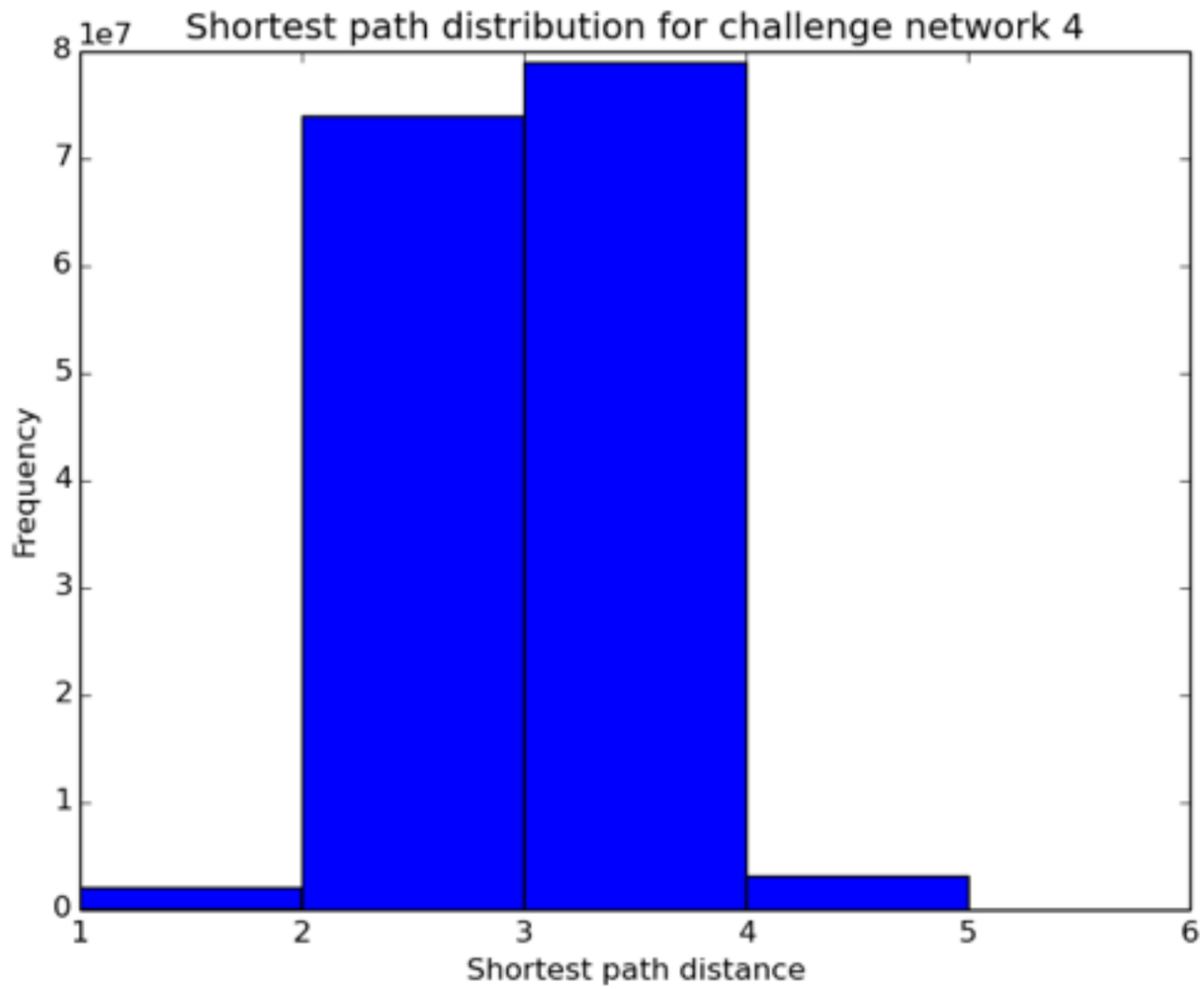
Dream Network 2



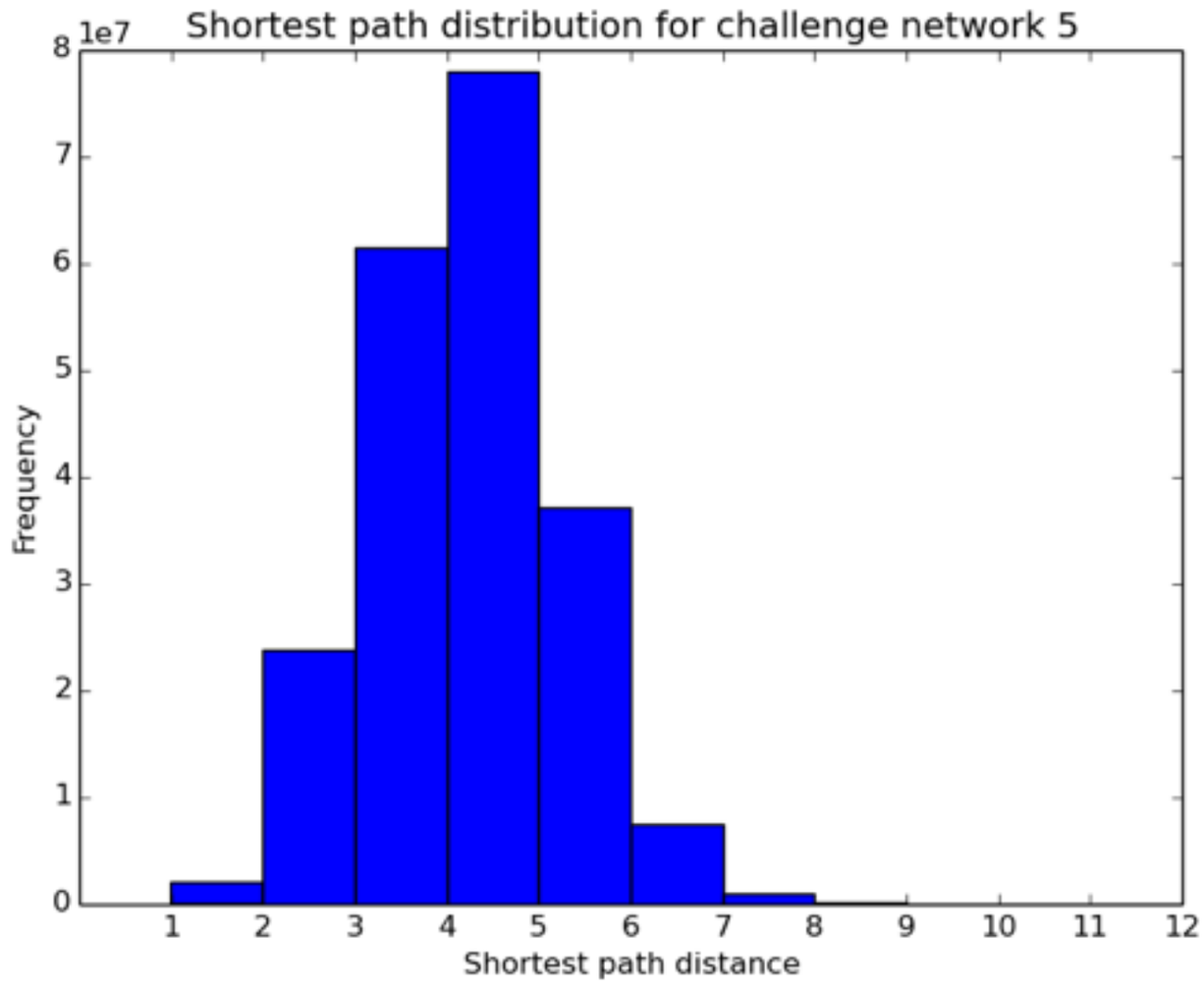
Dream Network 3



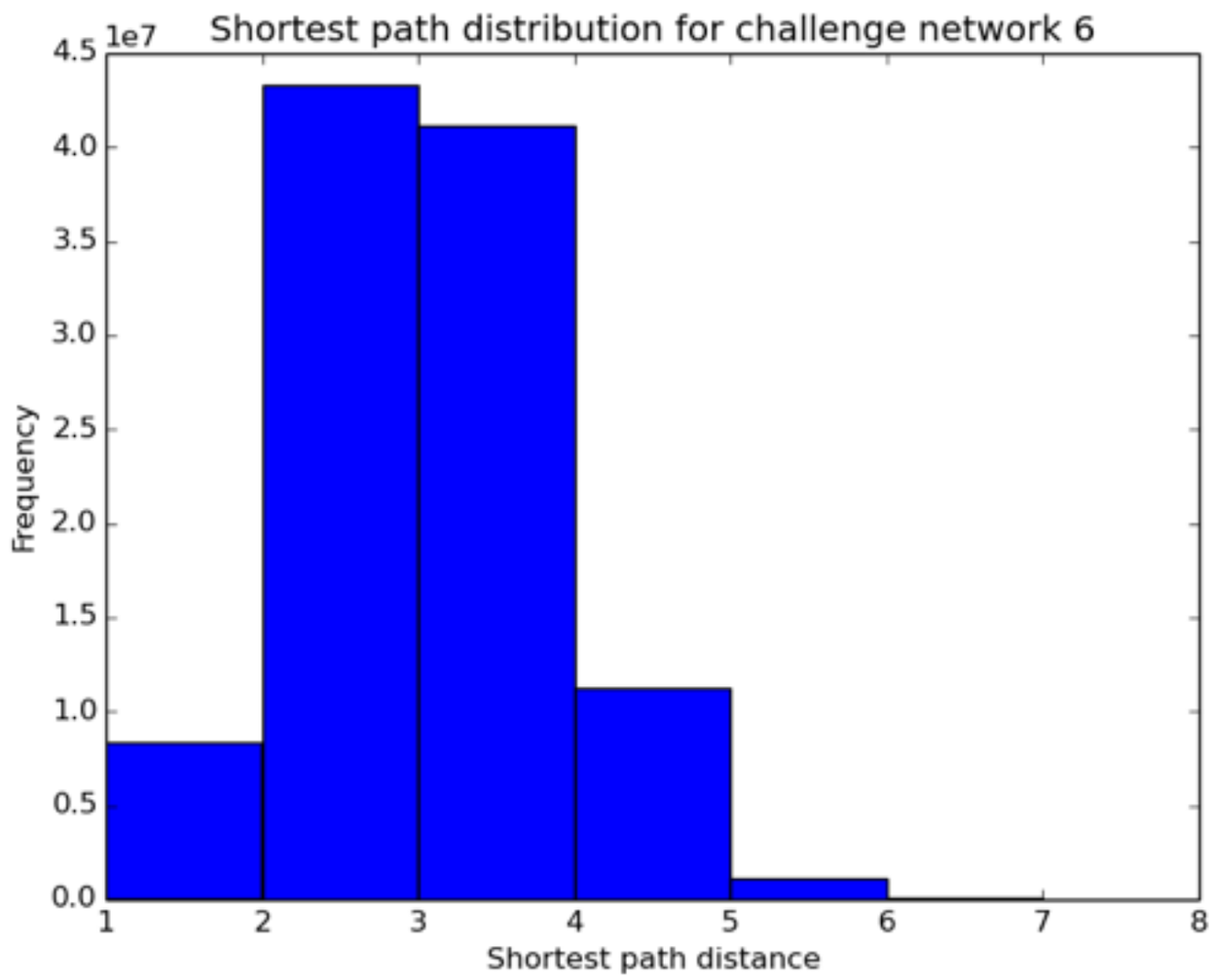
Dream Network 4



Dream Network 5



Dream Network 6



Our Approach

DSD

Diffusion State Distance

Cluster the resulting distance matrix

Correct cluster sizes

Look for bipartite subgraphs

Return valid clusters

Our Approach

DSD

Diffusion State Distance

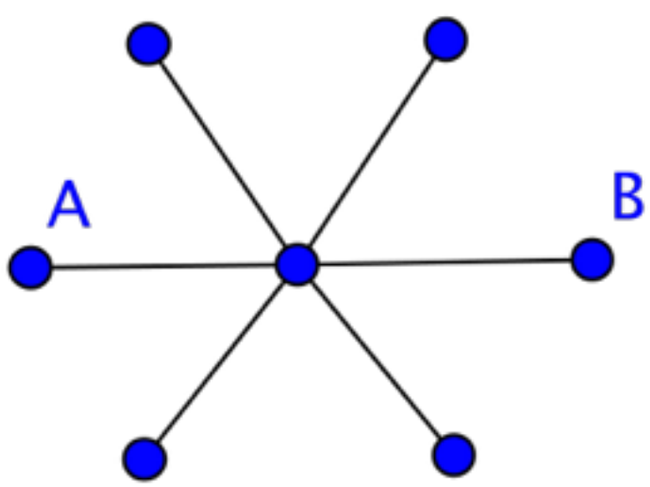
Cluster the resulting distance matrix

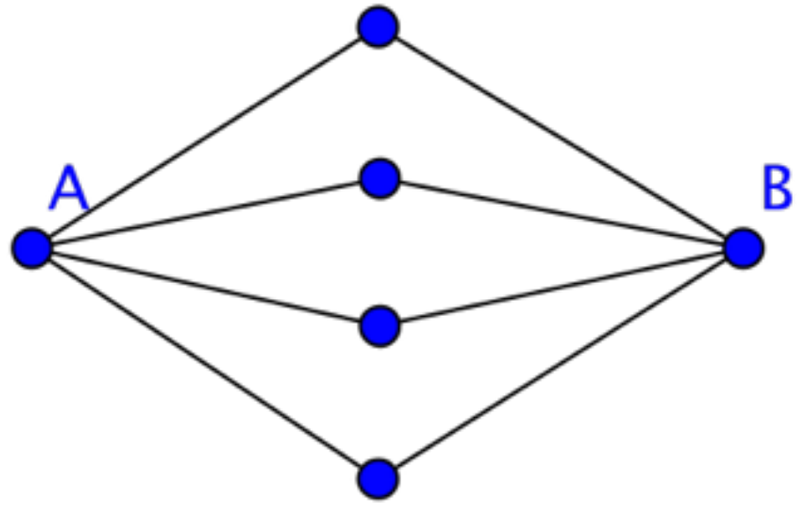
Correct cluster sizes

Look for bipartite subgraphs

Return valid clusters

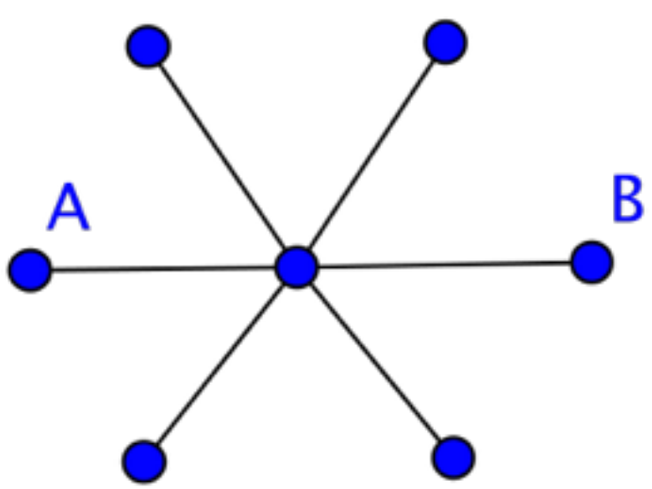
Our insight: not all short paths give equal indications of similarity

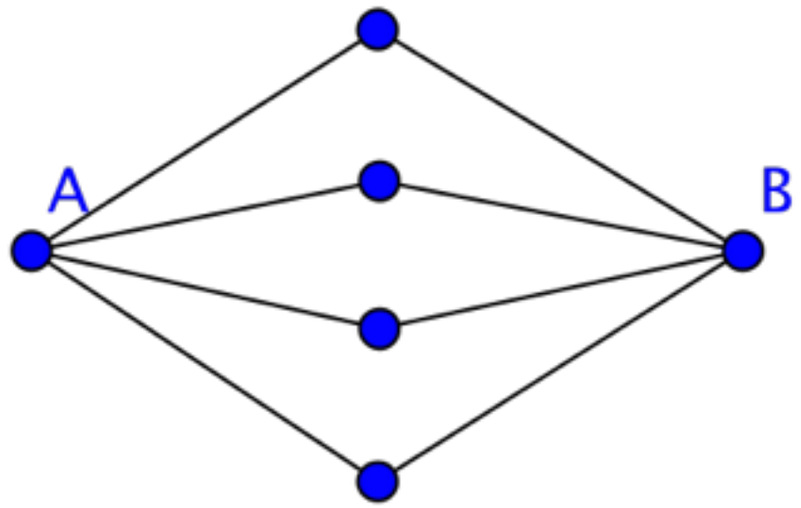




Define He (A,B) to be the expected number of times a k-step random walk starting at A reaches B (including 0-length walk)

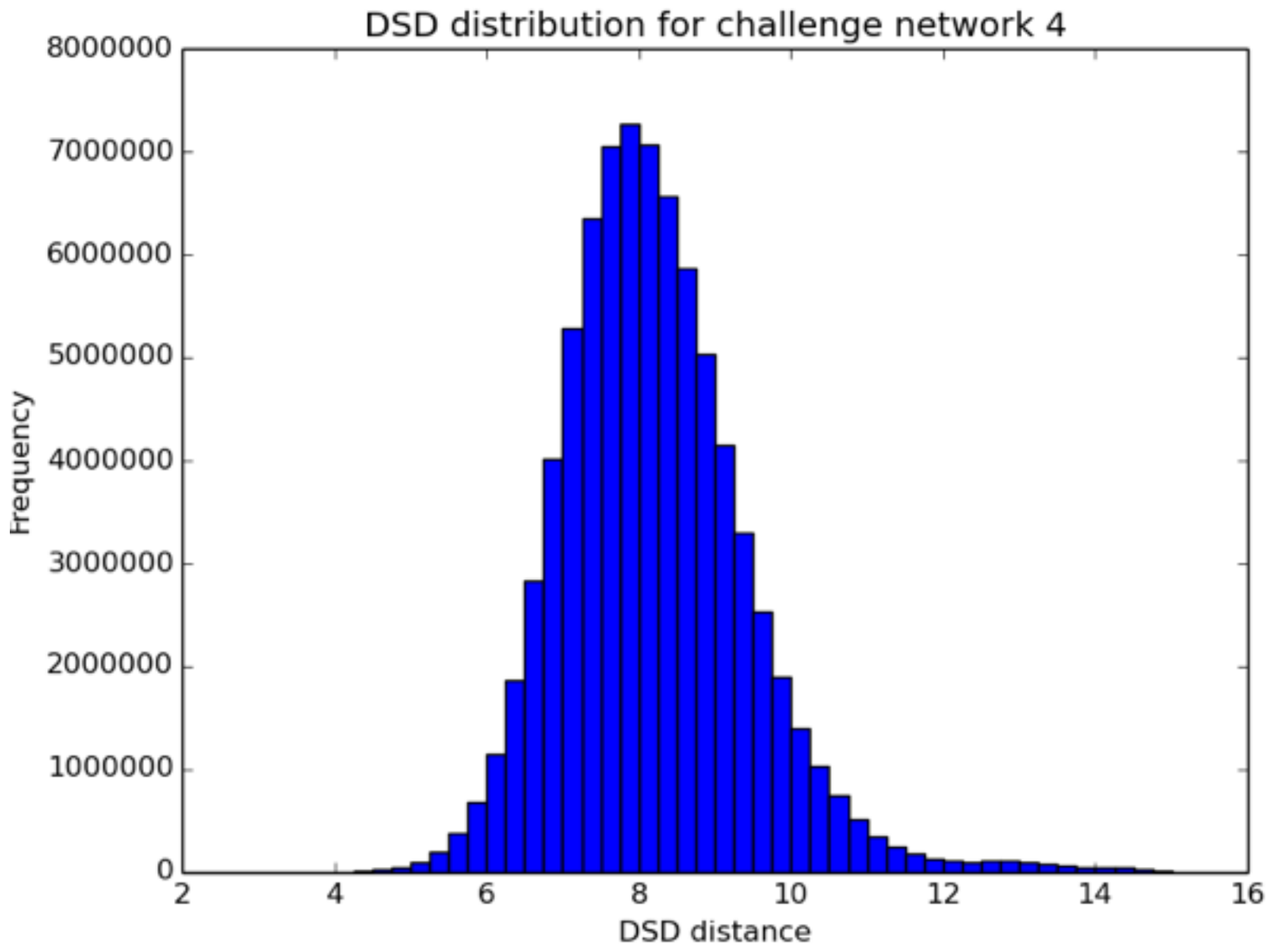
DSD: A Spectral Distance Metric





- $He^k(A) = (He^k(A, v_1), He^k(A, v_2), \ldots, He^k(A, v_n))$
- $DSD^k(A, B) = | He^k(A) - He^k(B) |_1$
- We prove DSD is a metric, and converges as $k \to \infty$: call the converged version DSD (A, B) .

DSD spreads out distances



Our Approach

DSD

Diffusion State Distance

Cluster the resulting distance matrix

Correct cluster sizes

Look for bipartite subgraphs

Return valid clusters

Speeding Up DSD

For an ergodic network, the DSD distance between two nodes u and v can also be written as

$$\mathrm{DSD}(u,v) = ||(\boldsymbol{b}_u^T - \boldsymbol{b}_v^T)(\boldsymbol{I} - \boldsymbol{P} + \boldsymbol{W})^{-1}||_1$$

where $\mathbf{b_i}$ is the i^{th} basis vector, **I** is the identity matrix, **P** is the transition matrix of the network, and **W** is a matrix where each row is a copy of $\boldsymbol{\pi}^{\mathbf{T}}$ = the steady state distribution of the network.

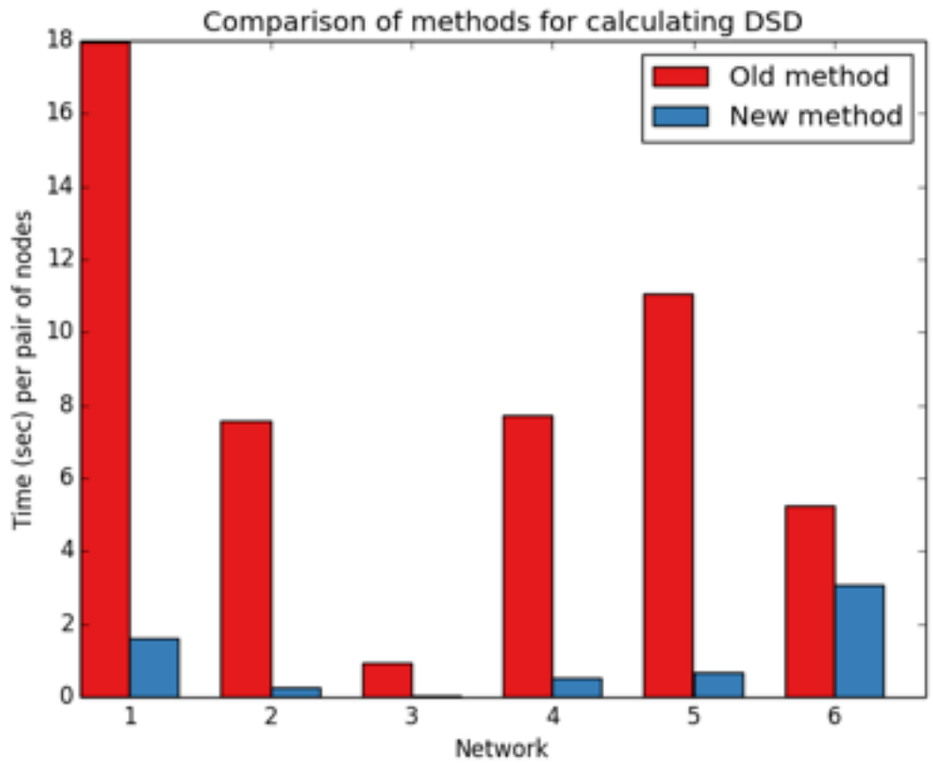
(proof of this in Cao et al. 2013)

Speeding Up DSD

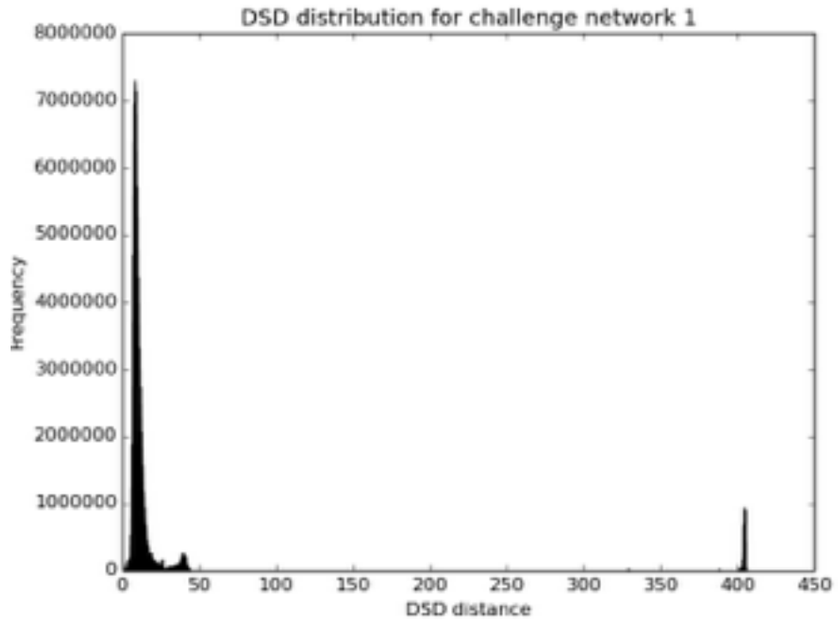
$$\mathrm{DSD}(u, v) = ||(\boldsymbol{b}_u^T - \boldsymbol{b}_v^T)(\boldsymbol{I} - \boldsymbol{P} + \boldsymbol{W})^{-1}||_1$$

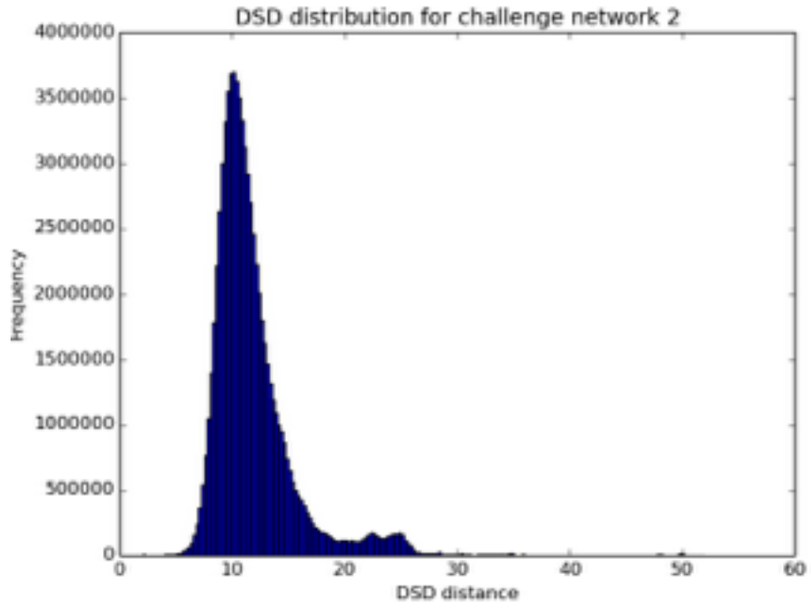
- Computing the matrix inverse $(\mathbf{I - P + W})^{-1}$ is the most computationally expensive part

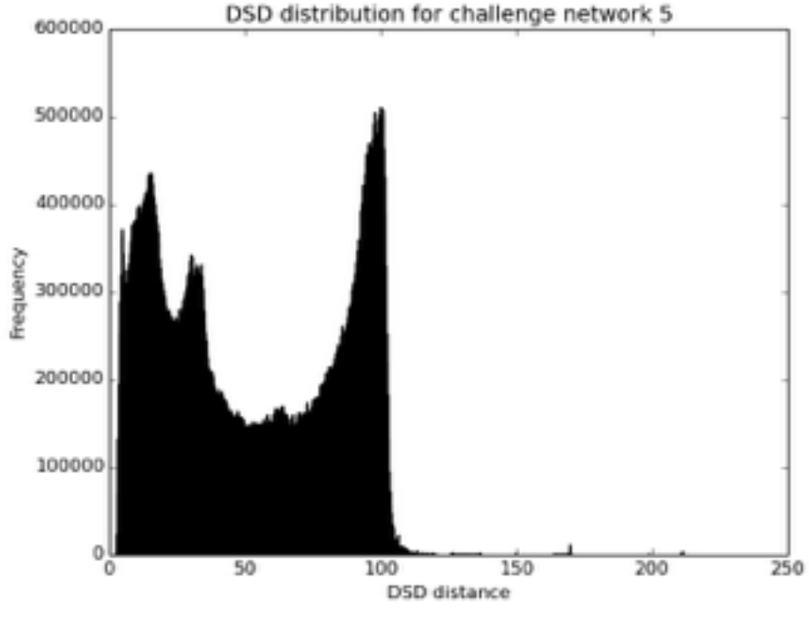
- We developed an efficient algorithm to reduce the amortized time for sparse networks, using algebraic multigrid methods

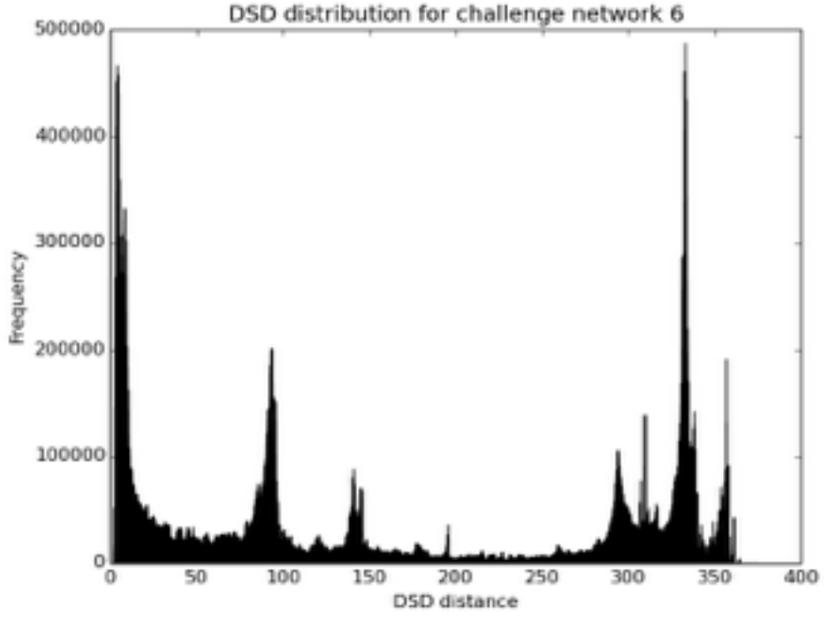


The method only computes approximate DSD; some artifacts, good enough









Our Approach

DSD

Diffusion State Distance

Cluster the resulting distance matrix

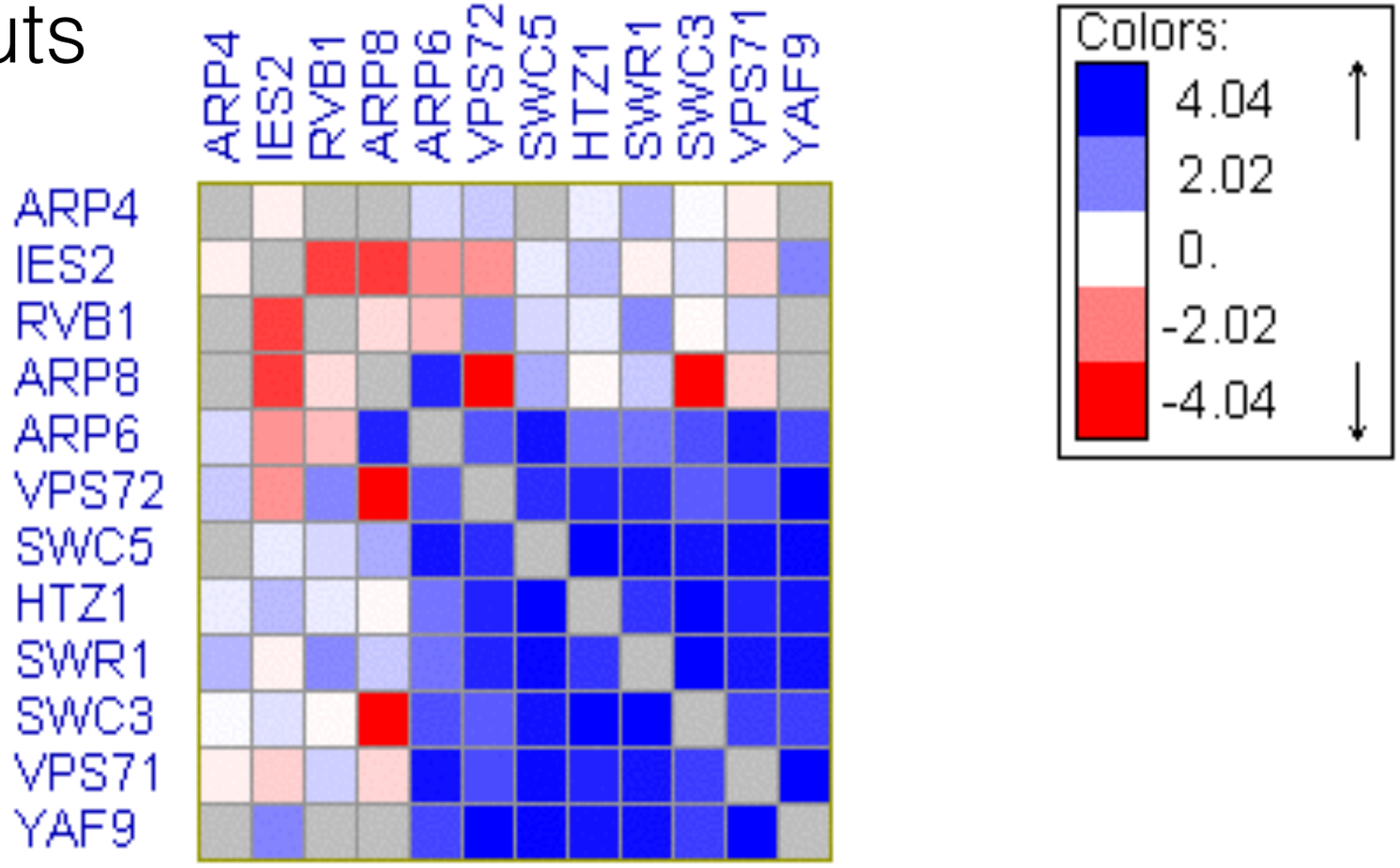
Correct cluster sizes

Look for bipartite subgraphs

Return valid clusters

Genetic interactions (epistasis)

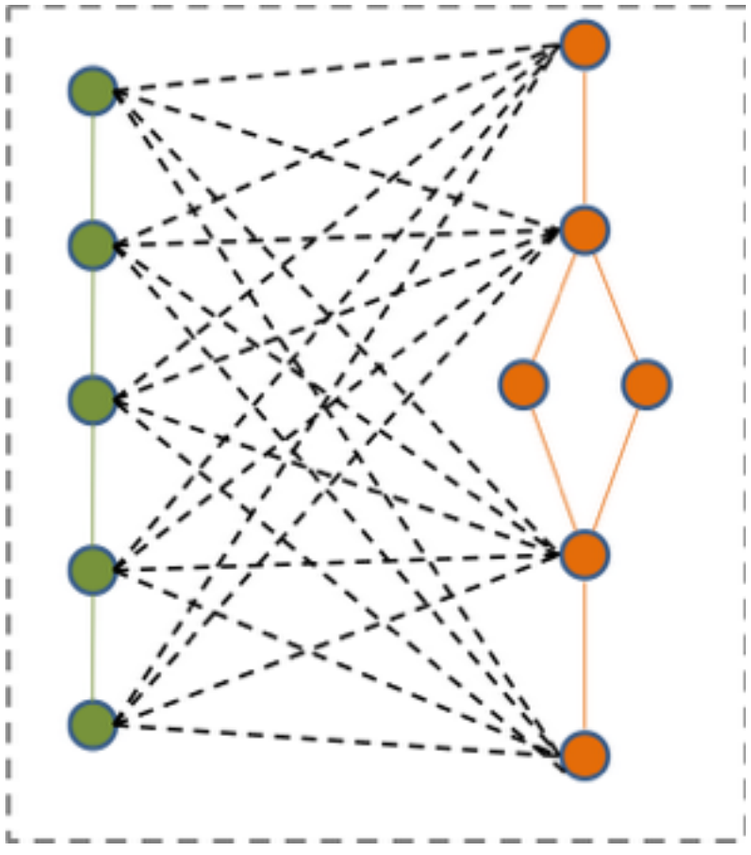
- For non-essential genes, we can compare the growth of the double knockout to its component single knockouts



Picture: Ulitsky

Finding Bipartite Subgraphs

- “Between Pathway Model” (Kelley and Ideker, 2005)
- Regions having many negative genetic interactions between them (and physical interactions within regions)
- May indicate compensatory biological pathways
- Use Brady et. al 2009 definition; based only on negative genetic interactions (ignore physical interactions)



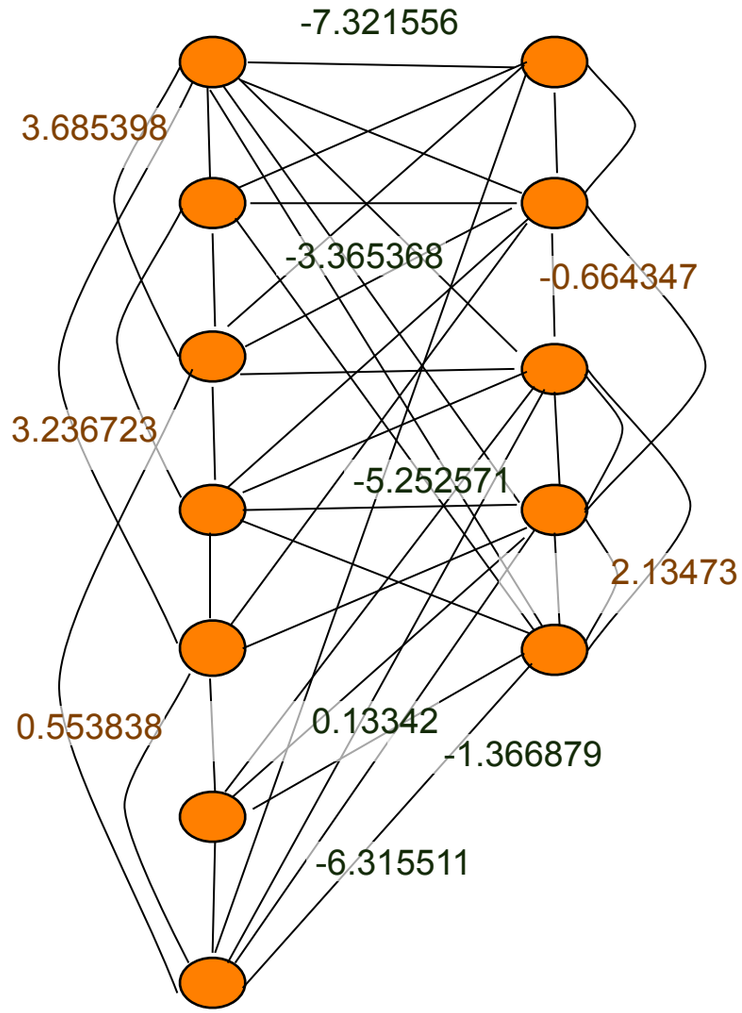
What is the Quality of a BPM?

Once we obtain a candidate BPM we can score it using interaction data.

Sum interactions within

Sum interactions between

Take the difference and normalize to create an interaction score



Look for a collection of BPMs in half the networks

Our Approach

DSD

Diffusion State Distance

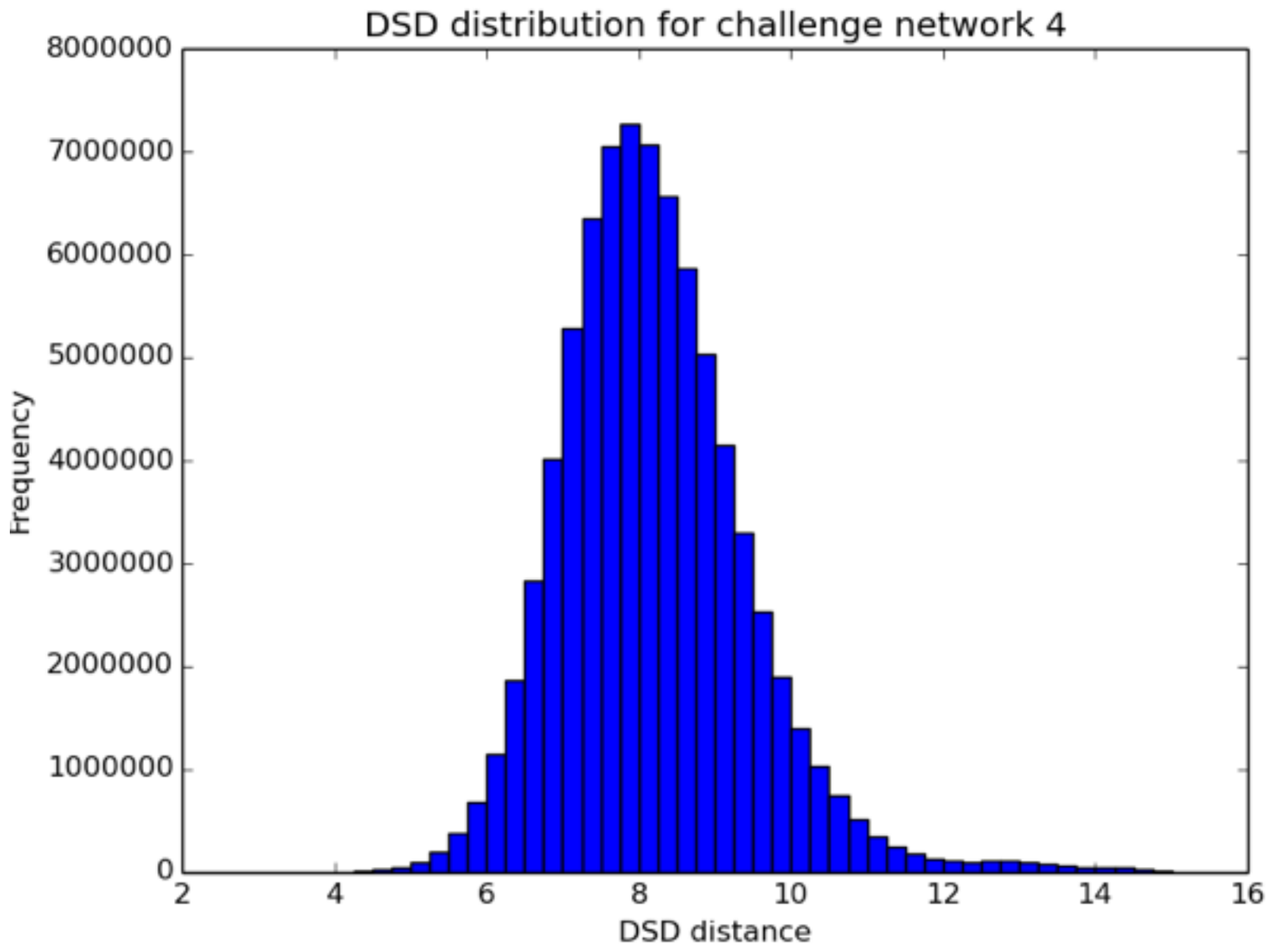
Cluster the resulting distance matrix

Correct cluster sizes

Look for bipartite subgraphs

Return valid clusters

Subchallenge 1, Step 1: Get the DSD matrix for each network

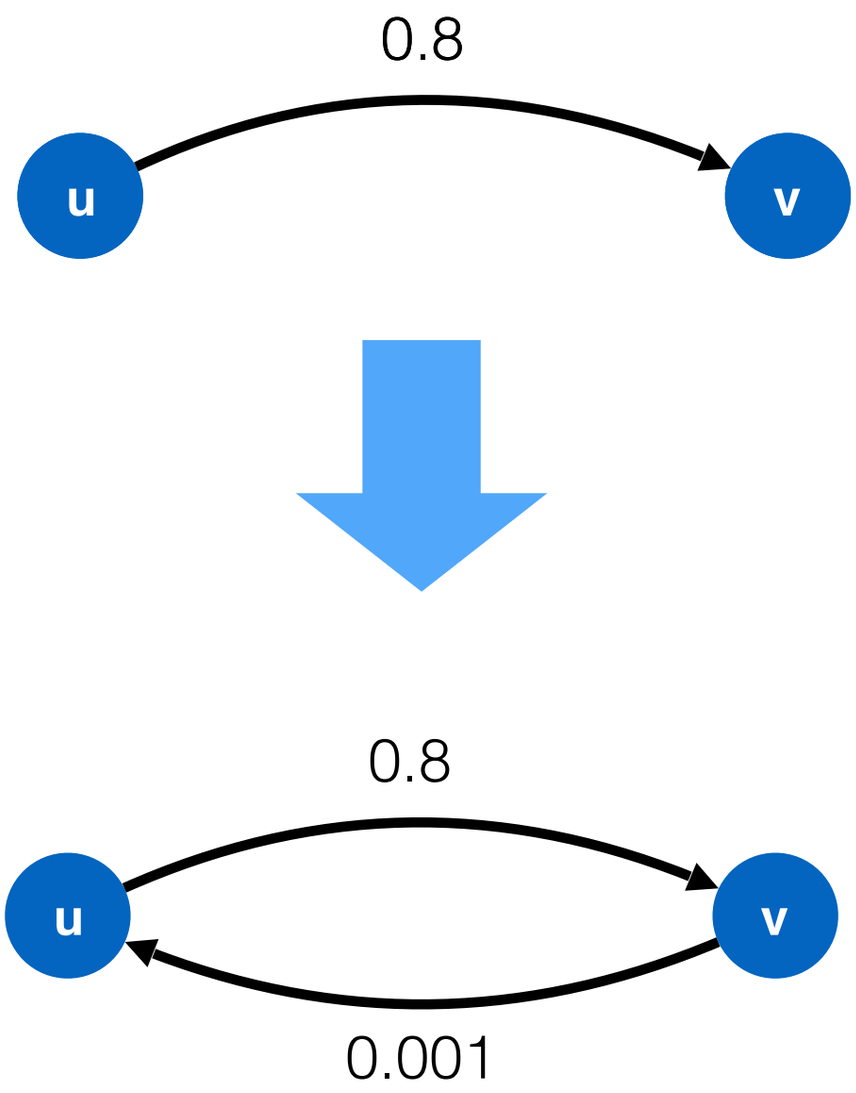


What about the directed network?

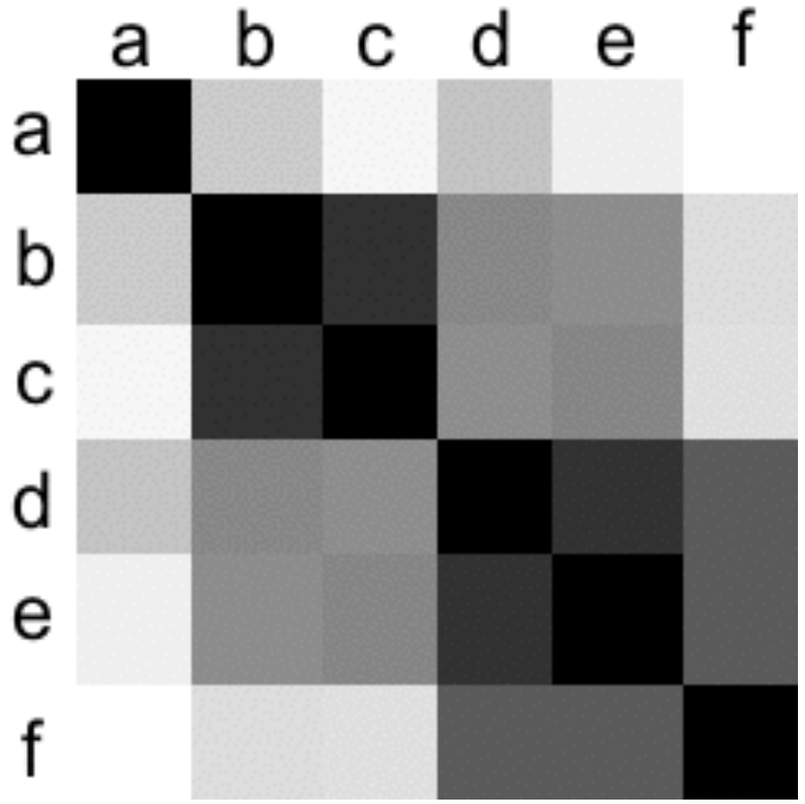
- We wanted to preserve edge direction, but keep the network (strongly) connected to run DSD in a sensible way

- Add low-weight back edges if none exists already

- Gave better results than treating all edges as undirected



We have a distance matrix. Now what?



- Many pre-existing methods to cluster a pairwise distance (or similarity) matrix
- Spectral clustering performed the best in the leaderboard rounds

Spectral Clustering

- Convert distance matrix to similarity matrix using radial basis function (RBF) kernel (high distance -> low similarity and vice-versa)
- Dimension reduction + k-means clustering (used default scikit-learn implementation)
- Tested different values of k in the training rounds

Pedregosa et al, JMLR 12:2825-2830 (2011)

Network 1 - Choosing k

Submission	NS (train)	NS for different FDR (1%, 2.5%, 10%)	NS (test)
k = 1000	14	(8, 10, 22)	16
k = 1200	14	(3, 7, 19)	N/A
k = 500, altered weighting scheme	14	(5, 9, 23)	N/A
k = 500	12	(3, 7, 20)	N/A

Network 5 - Choosing k

Submission	NS (train)	NS for different FDR (1%, 2.5%, 10%)	NS (test)
GC and k=100 overlap	6	(1, 2, 8)	N/A
GC and k=200 overlap	5	(3, 3, 7)	5
k=100	5	(1, 3, 6)	N/A

So far we have:

DSD

Diffusion State Distance

Cluster the resulting distance matrix

Correct cluster sizes

Look for bipartite subgraphs

Return valid clusters

Correcting Cluster Sizes

- Small clusters (size < 3): throw out
- Large clusters (size > 100): partition recursively (again using spectral clustering on the DSD matrix), until all clusters are of size < 100
- In practice, splitting large clusters improved most of the results we tested, and only hurt one result (out of ~15 comparisons)

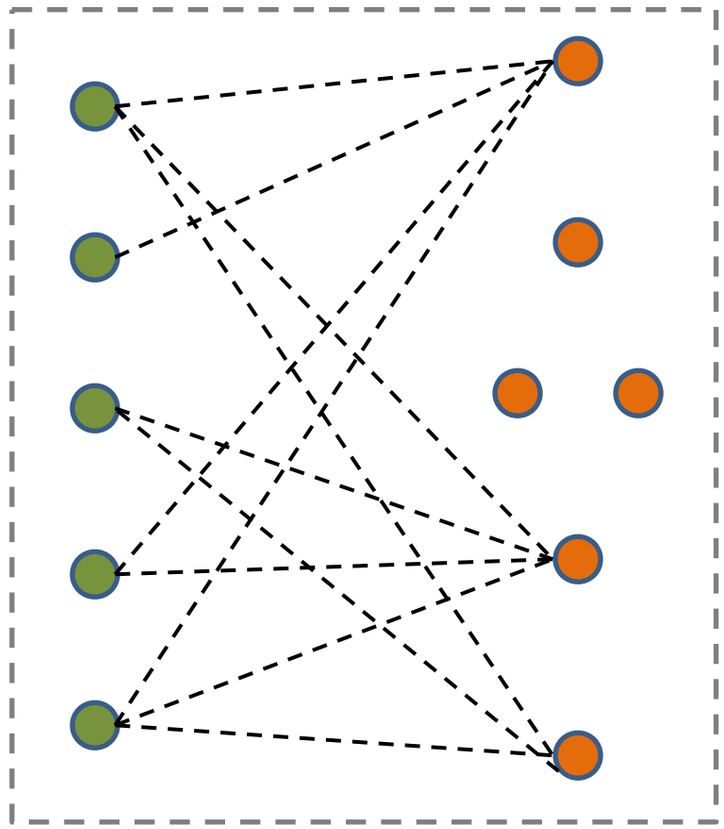
- For networks 1, 4, and 6, we're done.
- For networks 2, 3, and 5, we tried one more trick… looking for BPMs!

Finding Bipartite Subgraphs

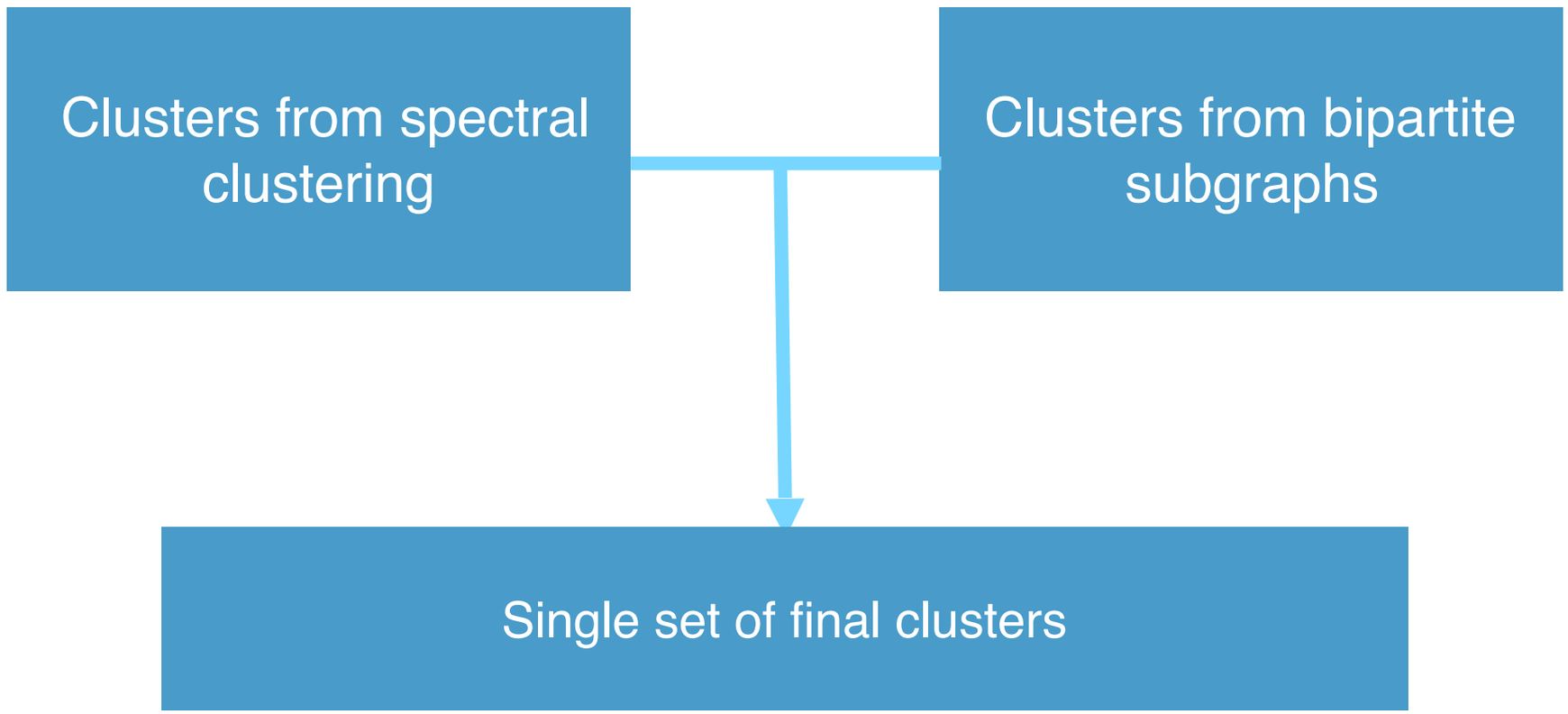
- We can identify many candidates by looking only at negative genetic interactions (Brady et al., 2009)

- We suspected there might be bipartite subgraphs in networks 2, 3, and 5

- We used the Genecentric software package to identify bipartite substructure

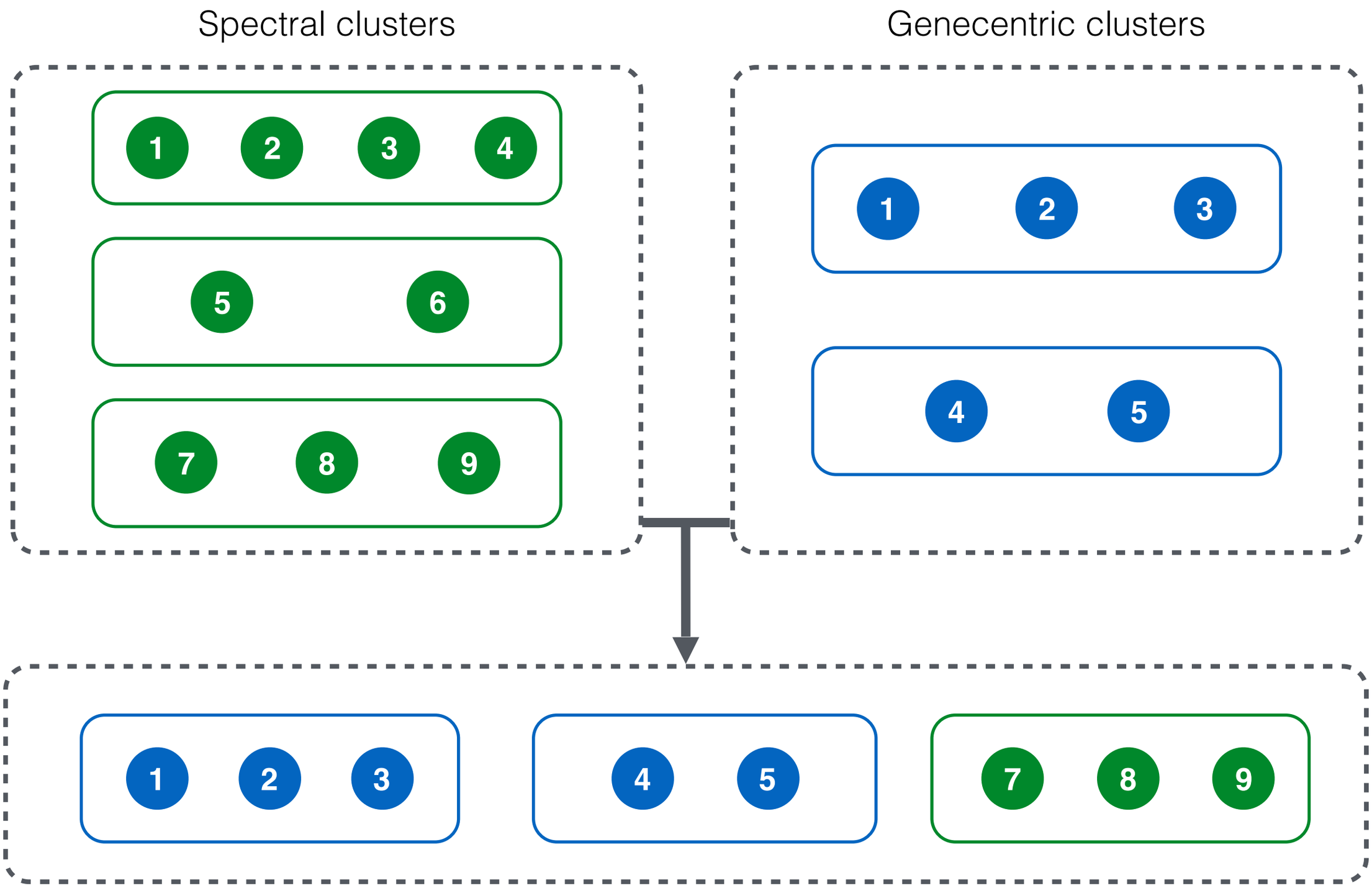


But, we're not quite finished

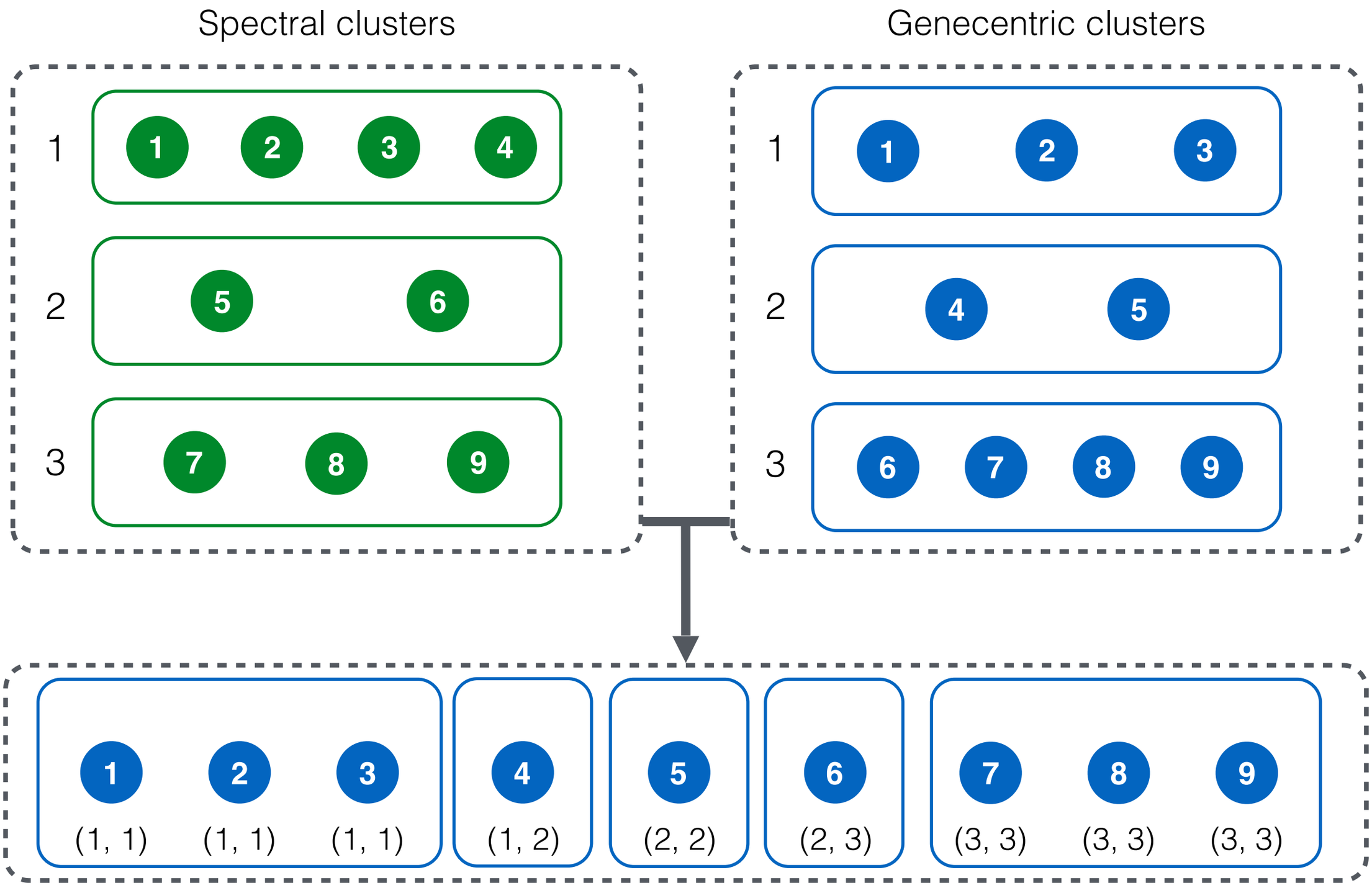


(for networks 2, 3, and 5)

Merging Clusters: Approach 1 (greedy)



Merging Clusters: Approach 2 (overlap)



Looking back:

DSD

Diffusion State Distance

Cluster the resulting distance matrix

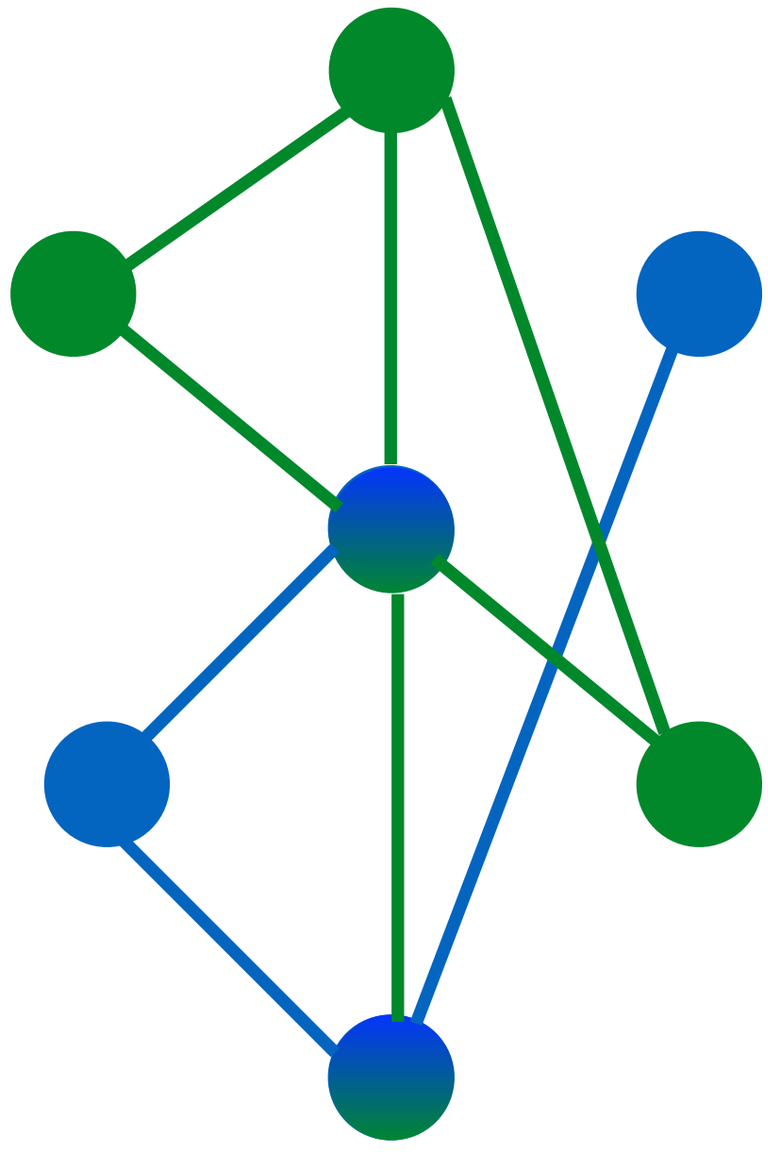
Correct cluster sizes

Look for bipartite subgraphs

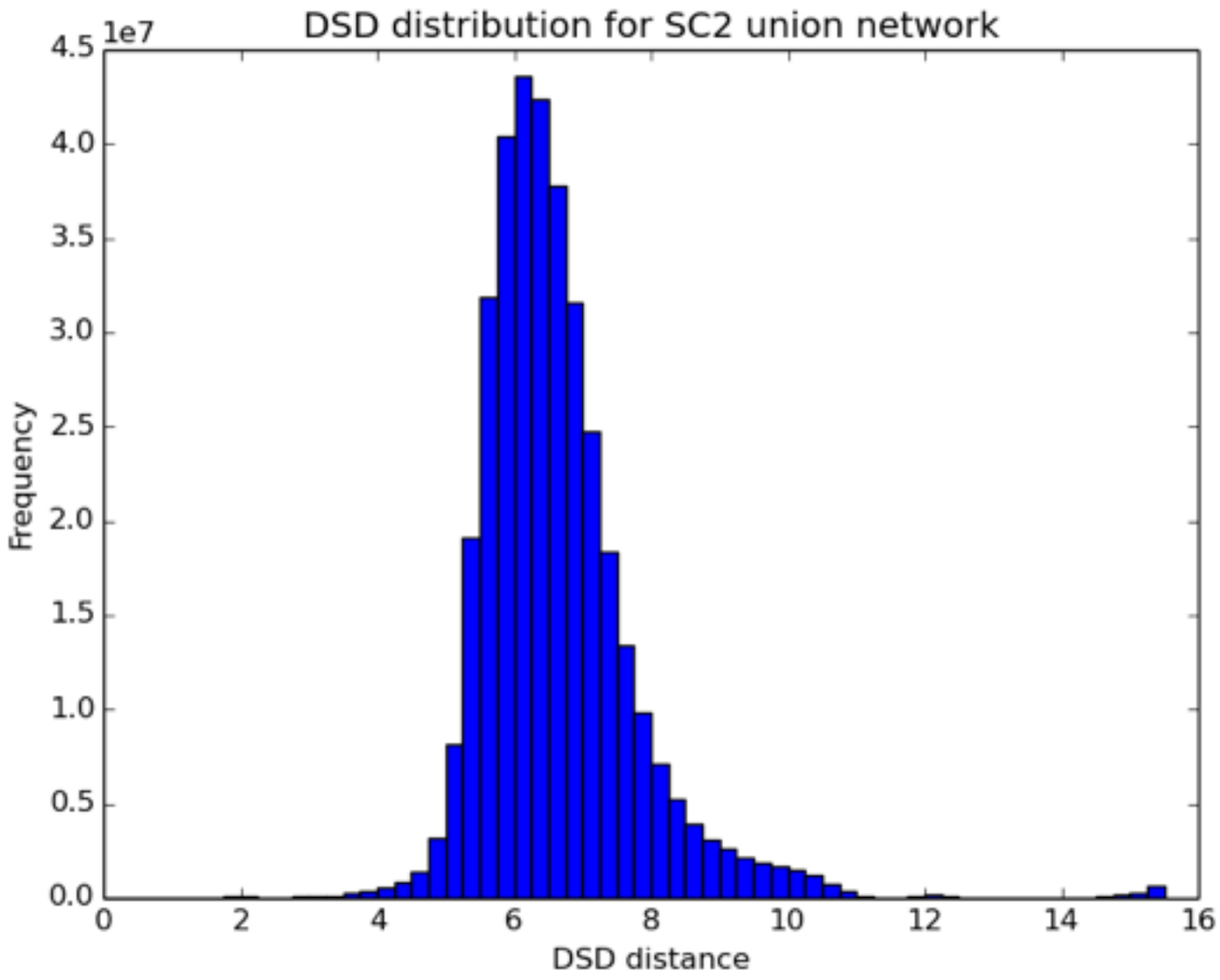
Return valid clusters

Subchallenge 2

- Took union of networks
- For overlapping edges with weights w_1, w_2, create edge with weight **max(w_1, w_2) + 0.05**
- Same DSD + spectral clustering approach as before
- Union of networks **1, 2, and 4** performed the best across FDR levels



Subchallenge 2:
Step 0: Superimpose networks 1,2 and 4
Step 1: Get the DSD matrix for the combined network



Future Directions

- Test modifications to edge weights
- Modify other clustering algorithms to run on top of DSD, there are many possible alternatives
- We could also test additional methods of combining edges (or adding networks) for Subchallenge 2

Acknowledgements

Thanks to the Tufts Bioinformatics and Computational Biology group for helpful ideas and critiques

BCBG

Bioinformatics and Computational Biology Group

Looking back