

Detangling PPI Networks to Uncover Functionally Meaningful Clusters

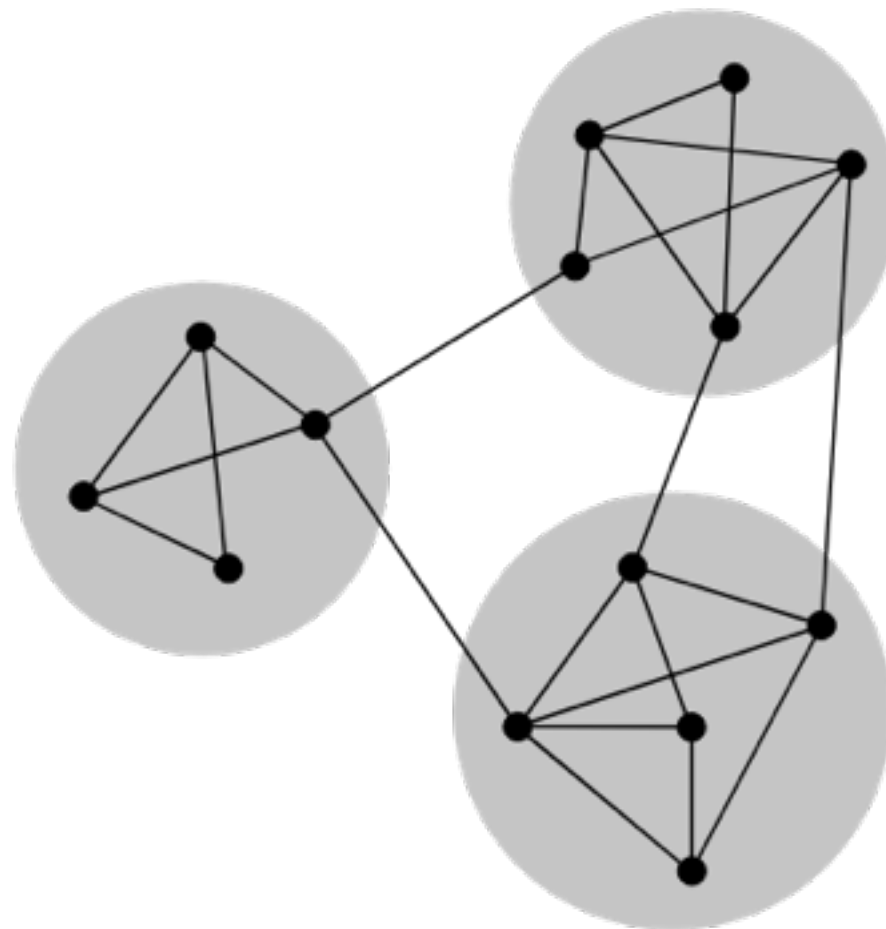
Sarah Hall-Swan, **Jake Crawford**, Rebecca Newman, Lenore J. Cowen

Department of Computer Science, Tufts University



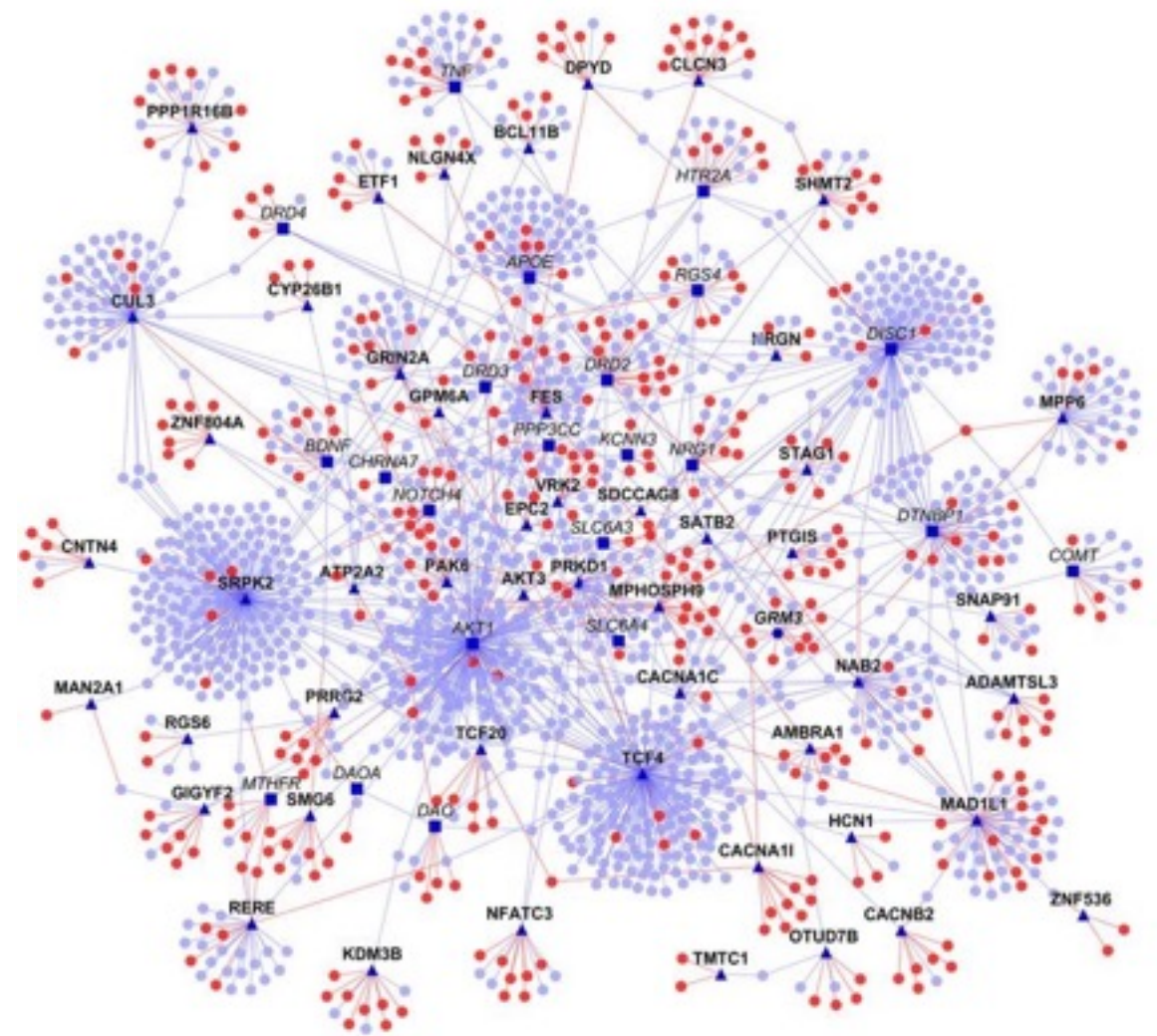
Community Detection

- Using only graph-theoretic measures of closeness and density, can we partition a network into meaningful “communities”?



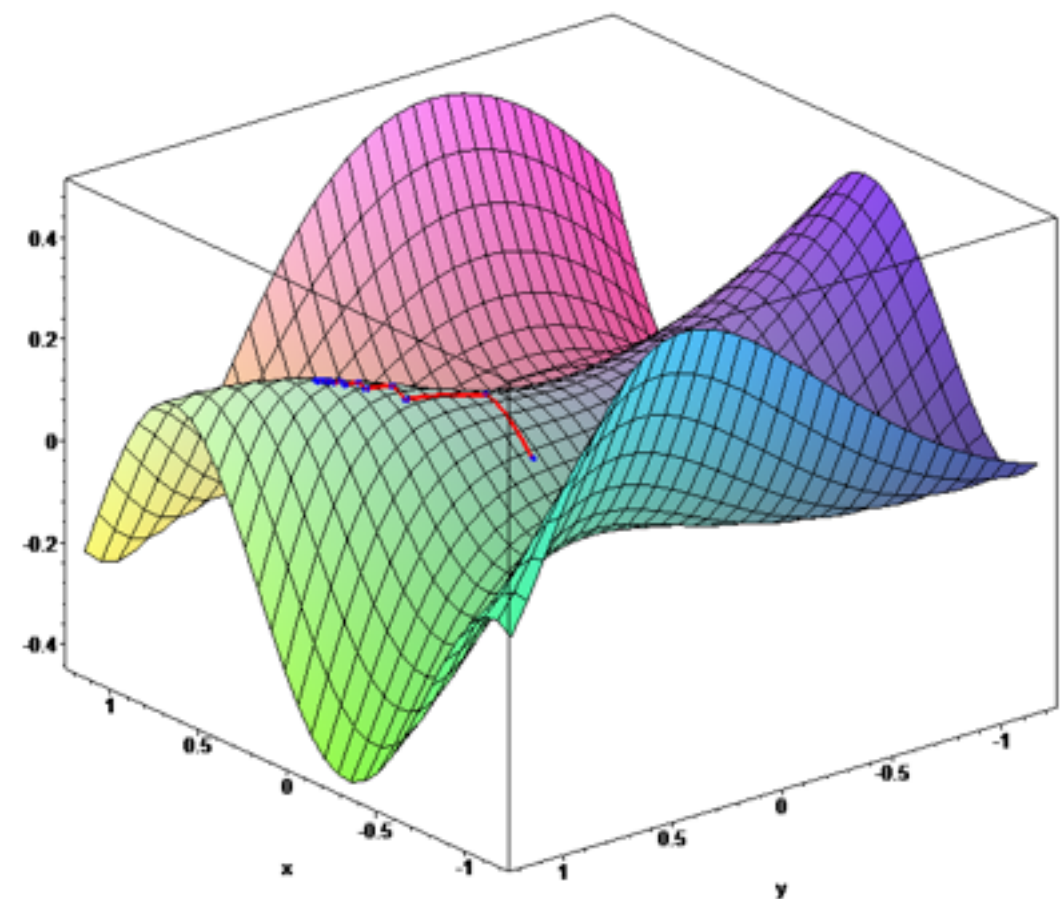
Why is this important biologically?

- Closely connected nodes often share a biological function
- May also be involved in similar pathways, diseases, etc.



Approaches

- Not one right answer to what makes a “best” partition
- Many existing methods optimize different graph-theoretic measures of cluster quality (modularity, conductance)
- Even if criteria is agreed on, exact optimization is often NP-hard



Some popular methods

- Louvain (iterative modularity optimization using local cluster modifications)¹
- Walktrap (agglomerative clustering via random walks)²
- Spectral clustering (dimension reduction, then clustering based on distance/similarity)³
- ...and many others

¹Blondel et al. *Journal of statistical mechanics* (2008)

²Pons and Latapy. *ISCIS* (2005)

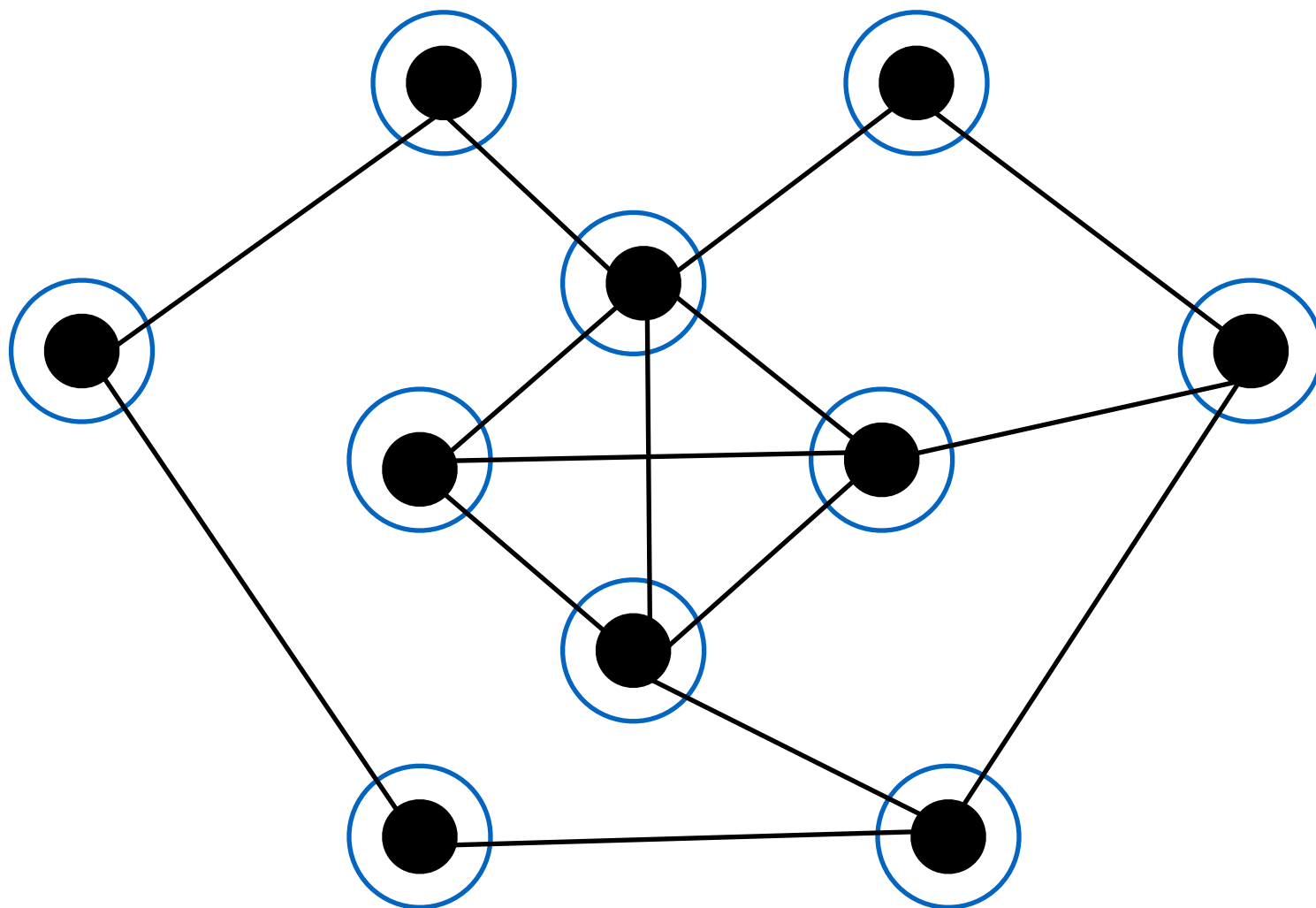
³Ng et al. *NIPS* (2001)

We look initially at the Louvain clustering algorithm.

(In the full paper, we explore how our results generalize to other algorithms)

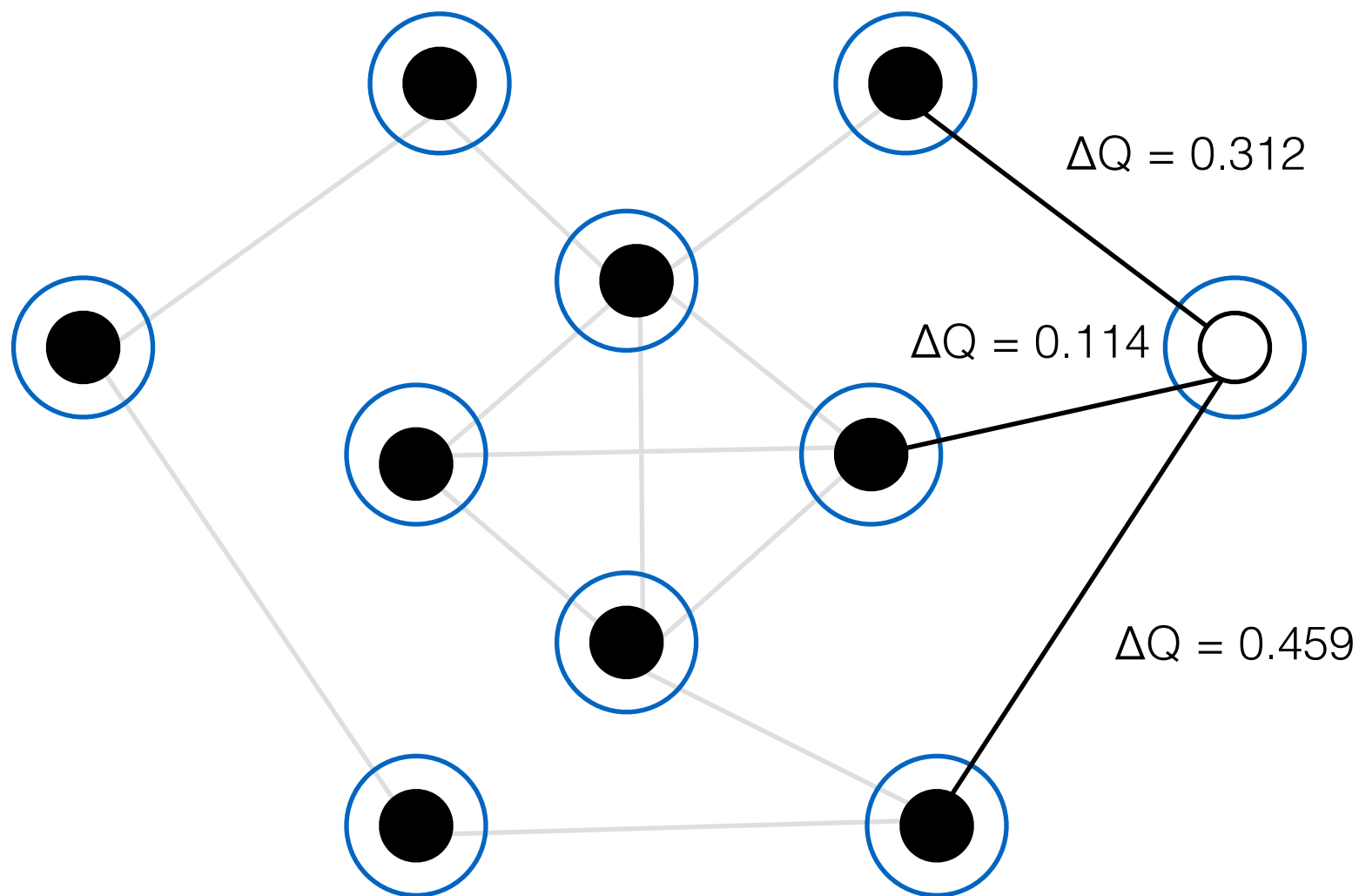
Louvain Clustering: a crash course

- Start with each node assigned to its own community



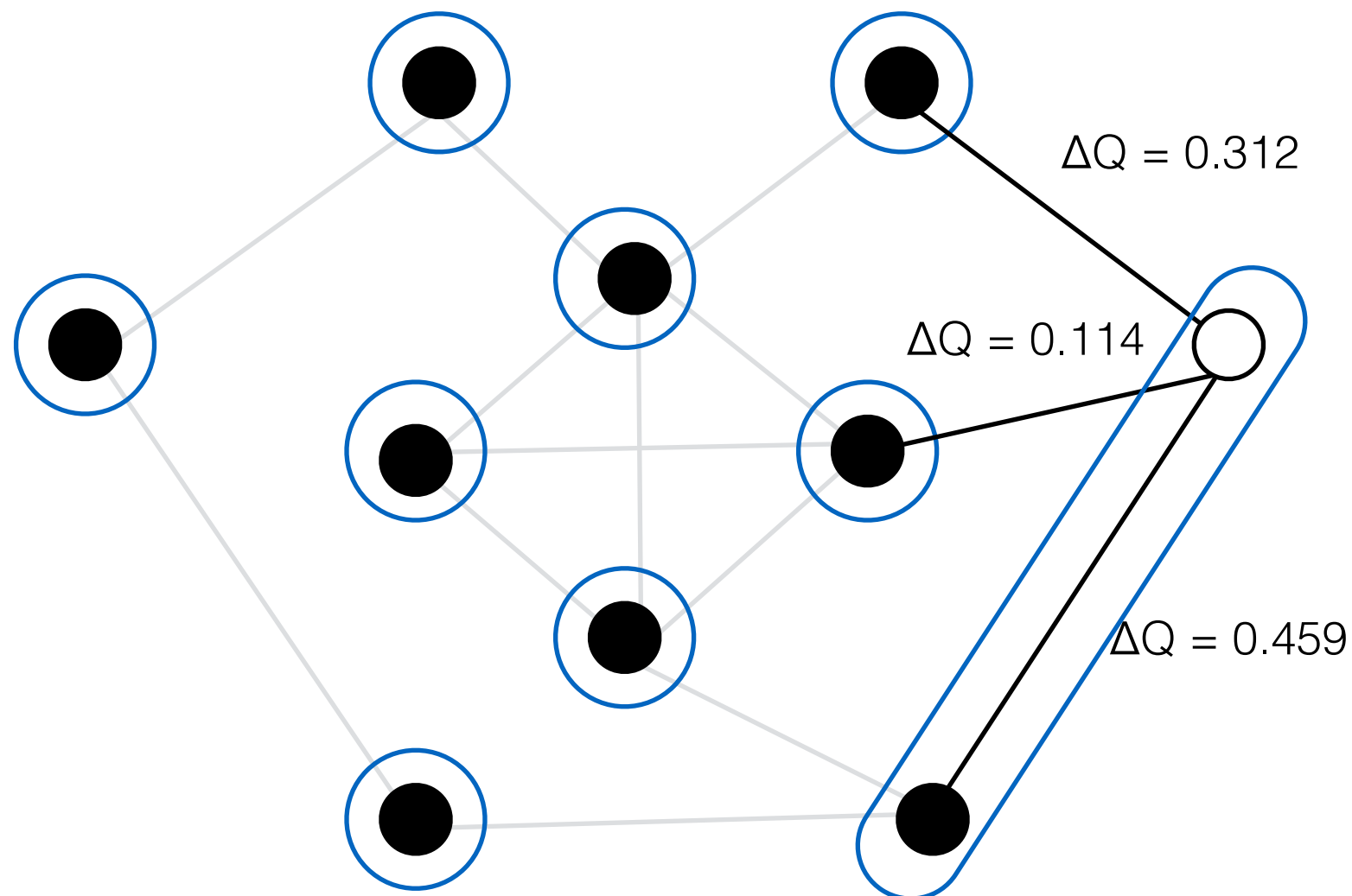
Louvain Clustering: a crash course

- Consider change in modularity if each node is grouped with its neighbors (this can be computed efficiently)



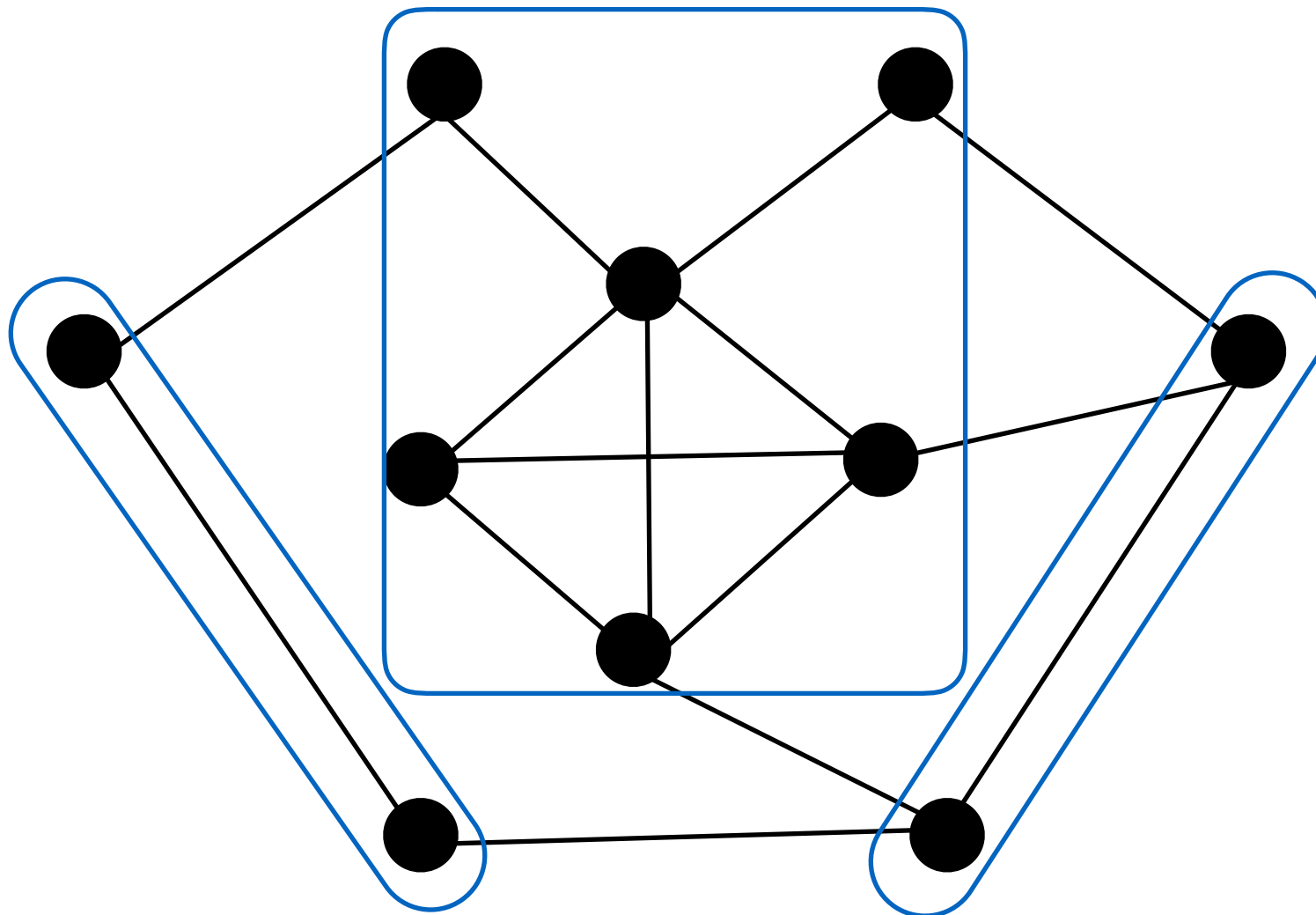
Louvain Clustering: a crash course

- Assign node to the community that gives the best (most positive) change in modularity



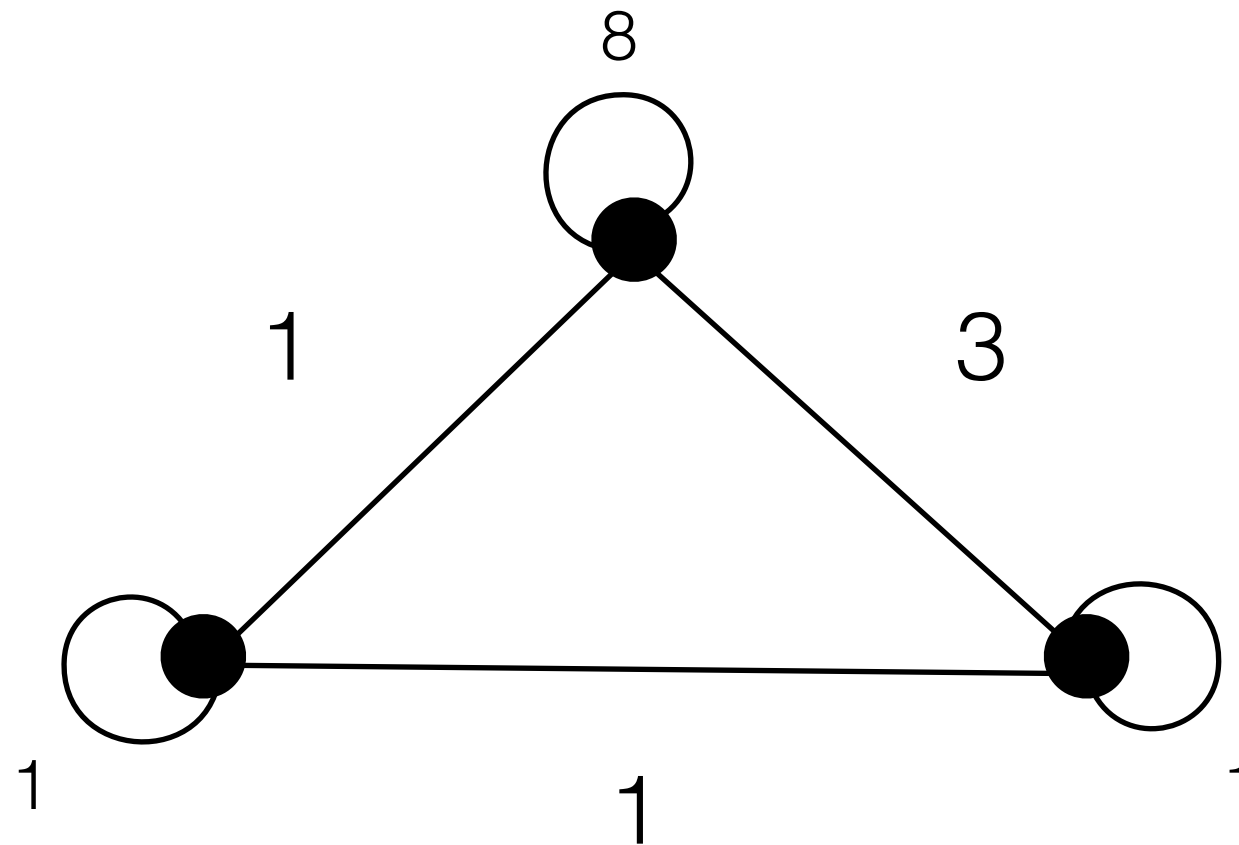
Louvain Clustering: a crash course

- Iterate through all nodes in the network, combining to give the best change in modularity at each step



Louvain Clustering: a crash course

- Iterate by grouping communities into individual nodes, and repeat until modularity no longer improves



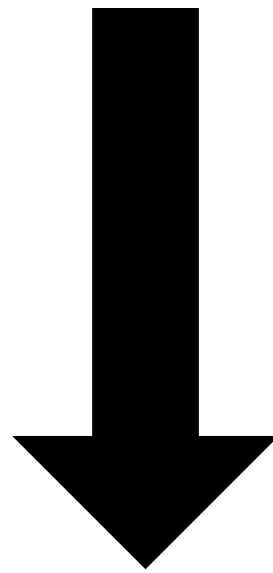
Louvain Clustering

- Note: this process is highly sensitive to the order in which nodes are grouped/compared with their neighbors
- So, the statistics we report (later) are from the median over 10 independently randomized runs
- We also consider a version of Louvain that does not merge clusters if they are bigger than 100 nodes

Importantly, Louvain requires some definition of what it means for two nodes in a network to be “neighbors”.

Our question: **Can we redefine distance to better identify similar genes as neighbors?**

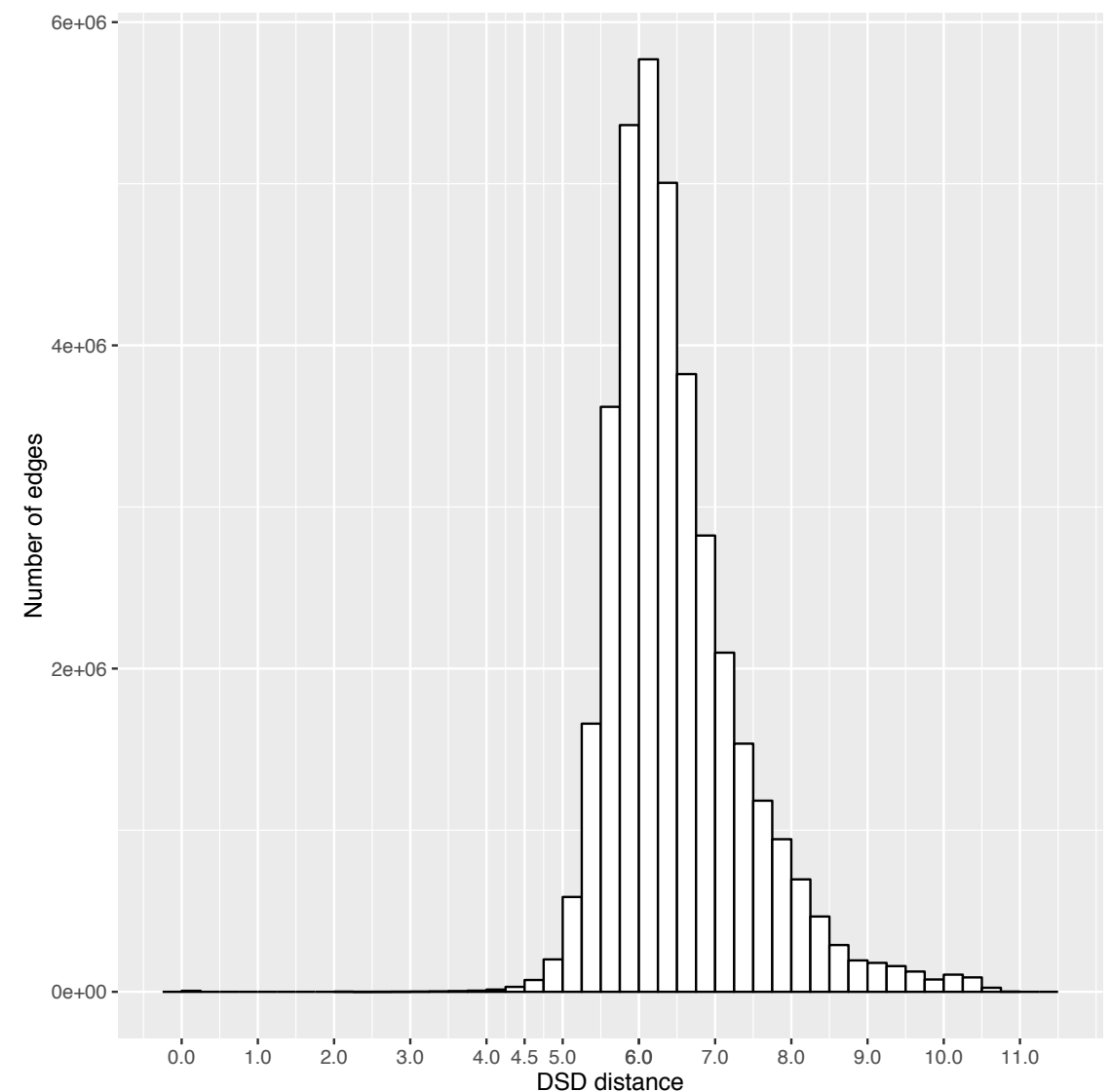
Can we redefine distance to better identify similar genes as neighbors?



We used our favorite distance metric, Diffusion State Distance (DSD)

DSD: A Spectral Distance Metric

- A random walk-based, fine-grained distance measure for biological networks
- We proved DSD is a metric, and converges as the number of walk steps goes to infinity

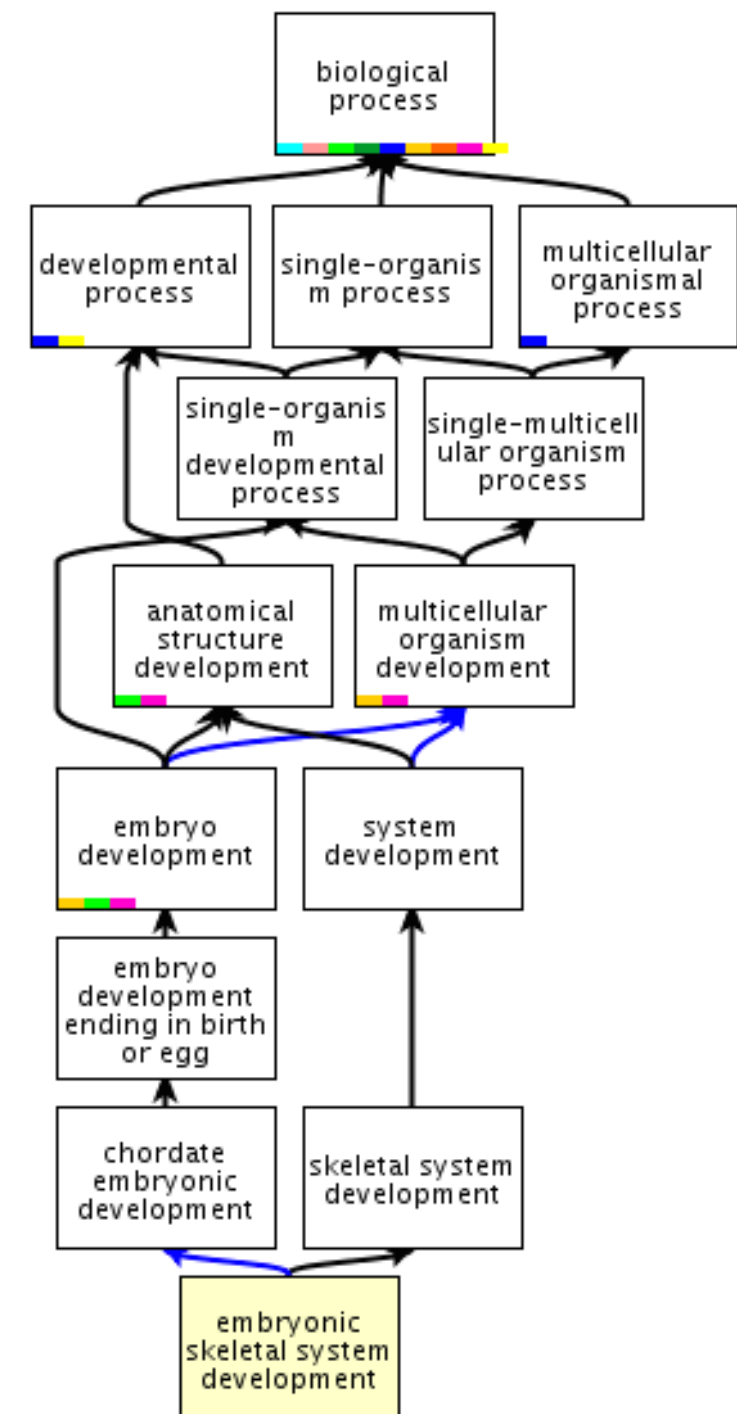


We claim: detangling the network
using DSD distance makes
Louvain produce a “better” set of
clusters in the network.

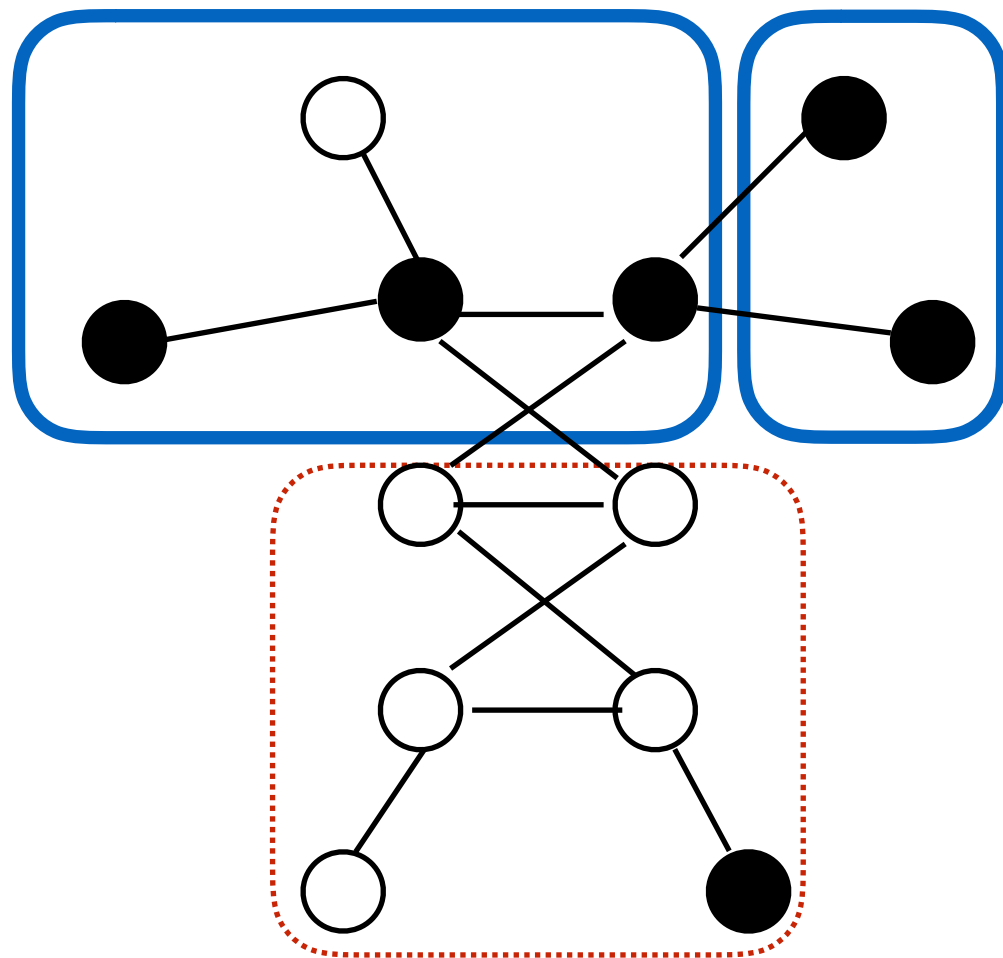
But, what is a “better” set of clusters?

- Define a single cluster as “good” using the well-studied notion of functional enrichment; i.e., it has many nodes annotated with the same function
- Even if we agree on what makes a good cluster, it’s not yet obvious what makes a good *set* of clusters

To identify the functions themselves, we use Gene Ontology annotations



To identify enriched clusters, we statistically compare cluster size to number of annotations (using the FuncAssociate tool¹)



● = “annotated with function f ”

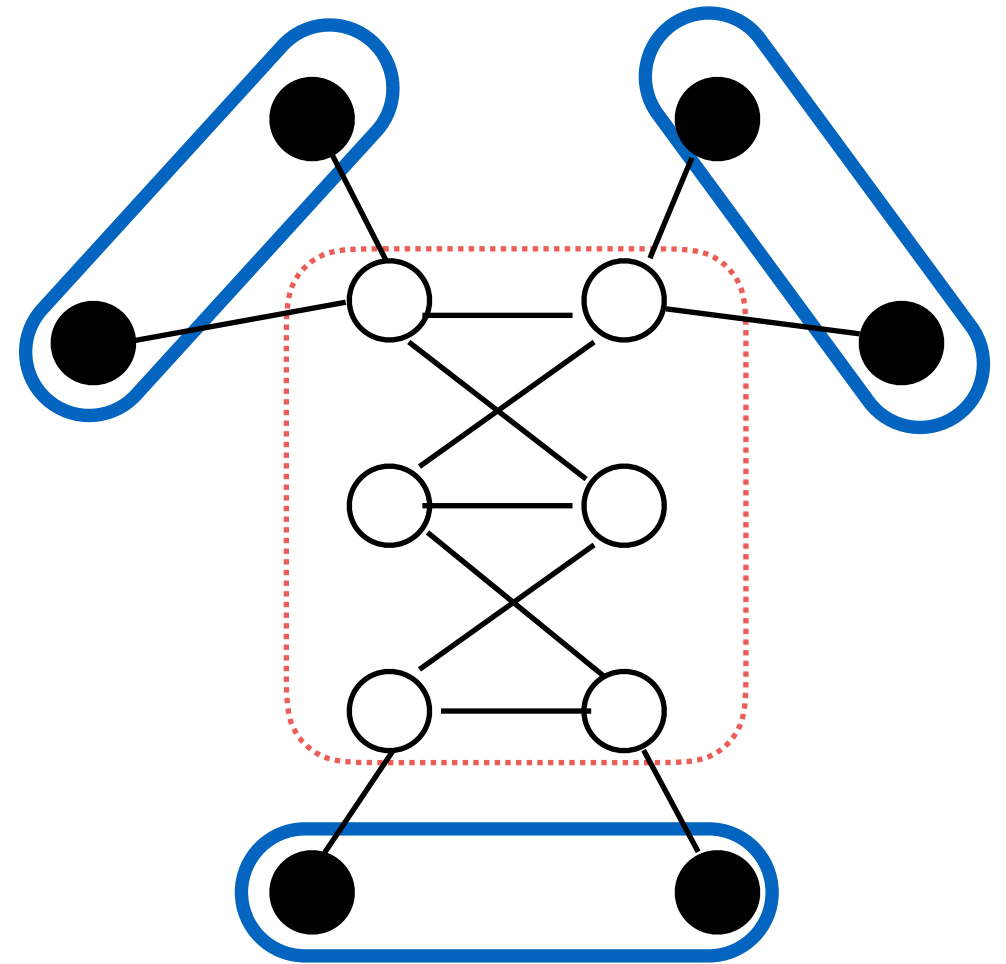
¹Berriz et al. *Bioinformatics* (2009)

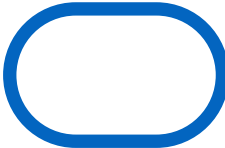
Now we know what to look for in each individual cluster.

But, how can we score a partition (i.e., a group of clusters)? How can we judge which of two partitions is “better”?

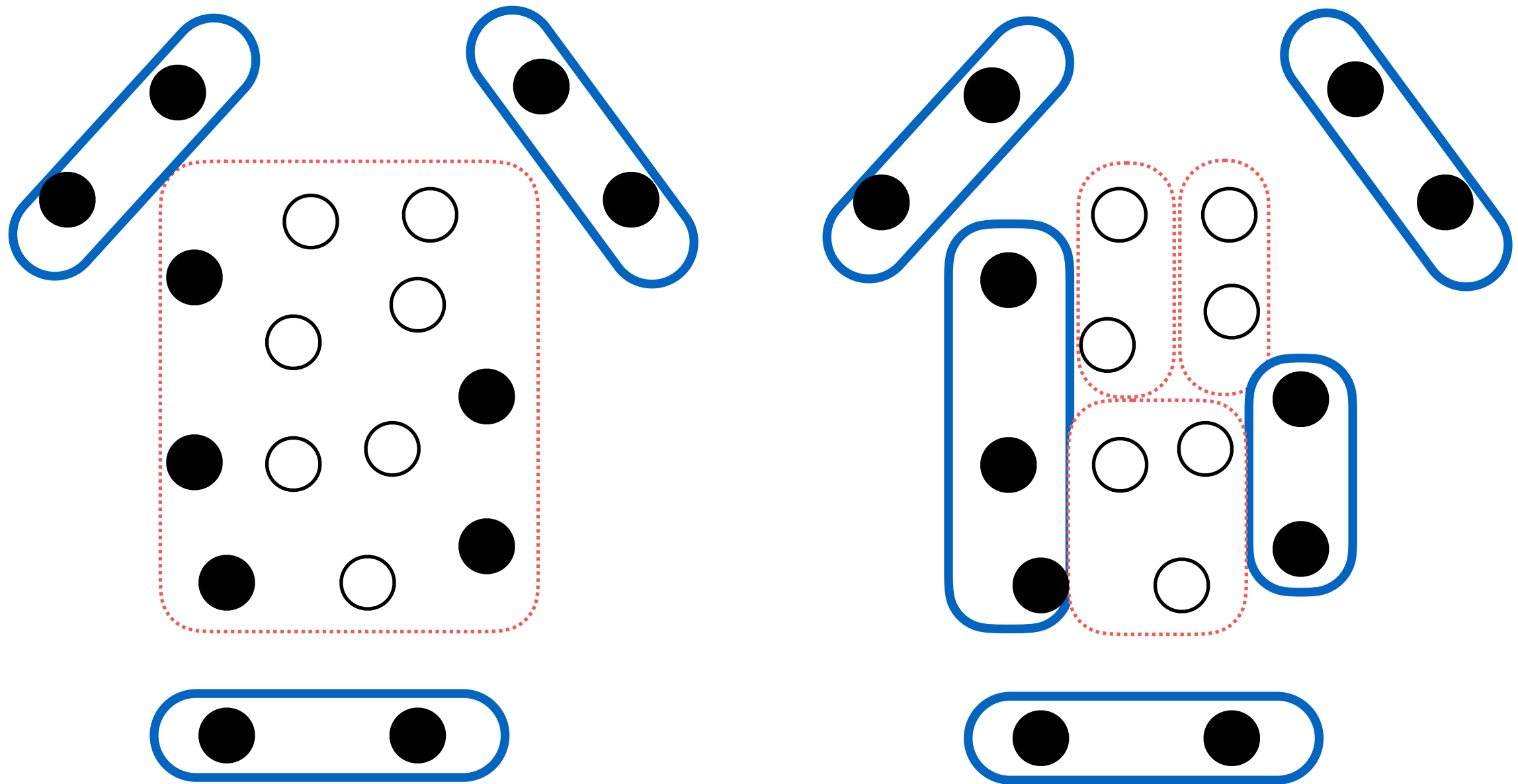
Idea 1: % of clusters enriched

- Gives a way to compare a set of communities in a global sense



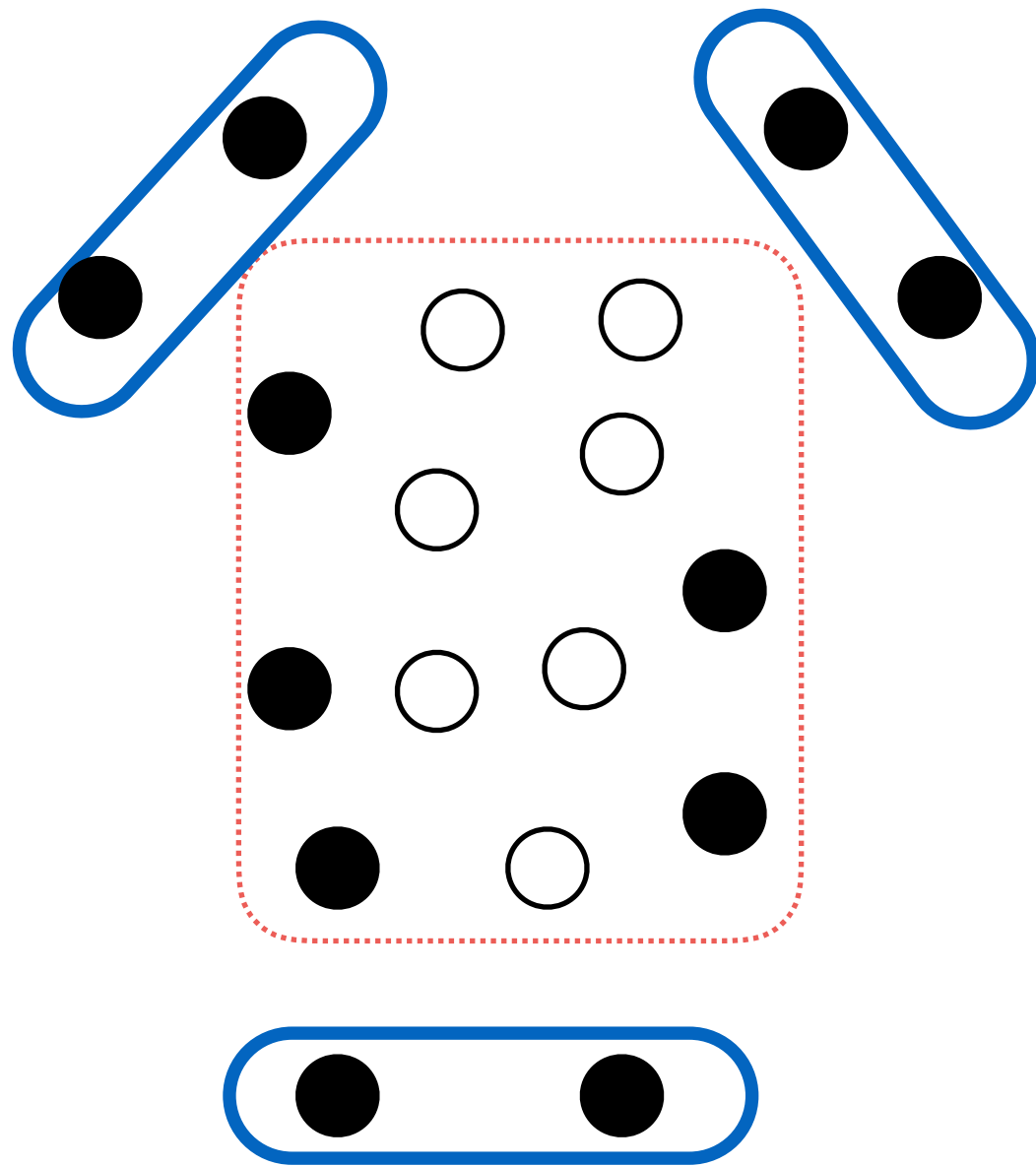
 = enriched
 = not enriched

Idea 1: % of clusters enriched

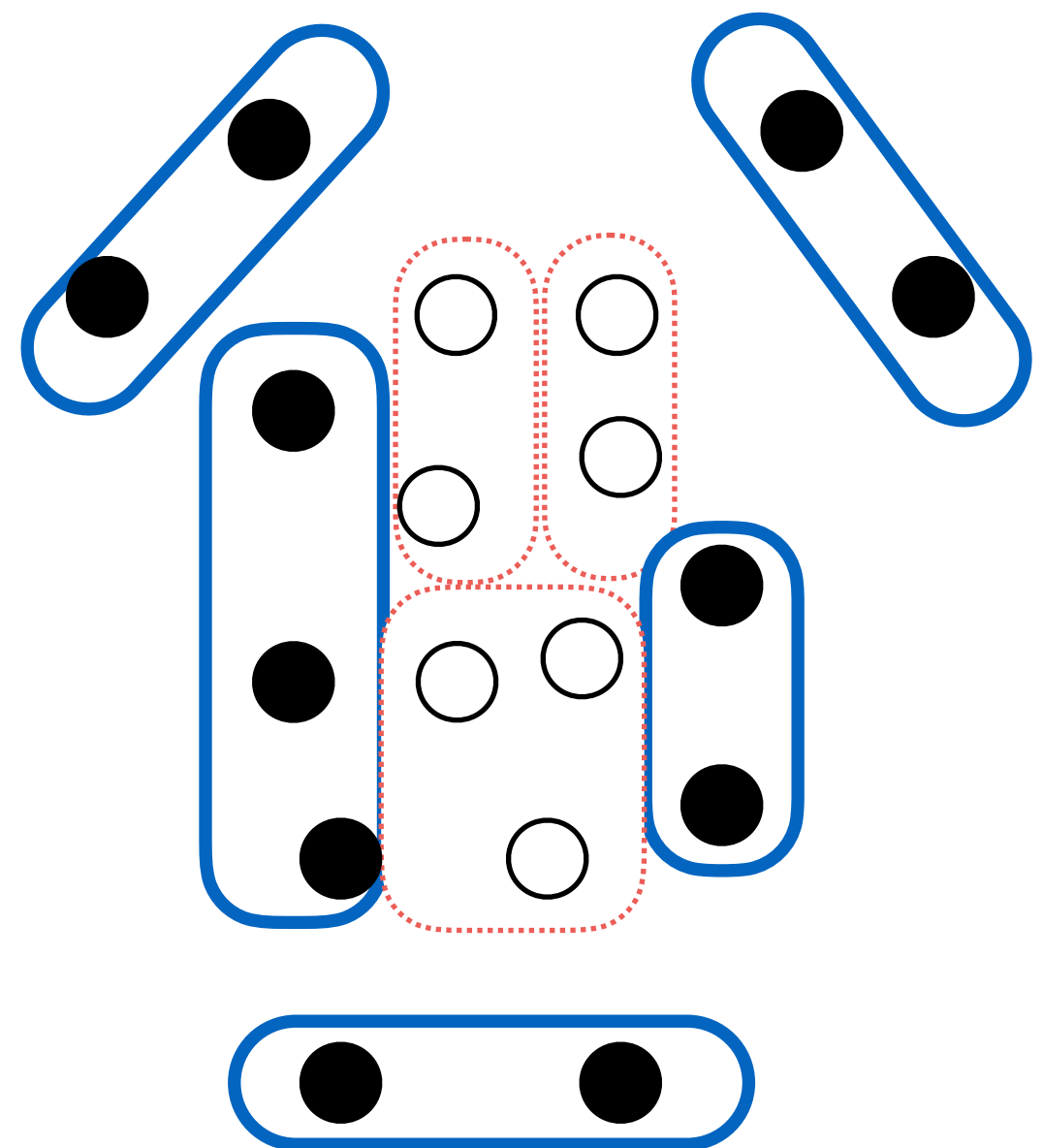


● = “annotated with function f ”

Idea 1: % of clusters enriched



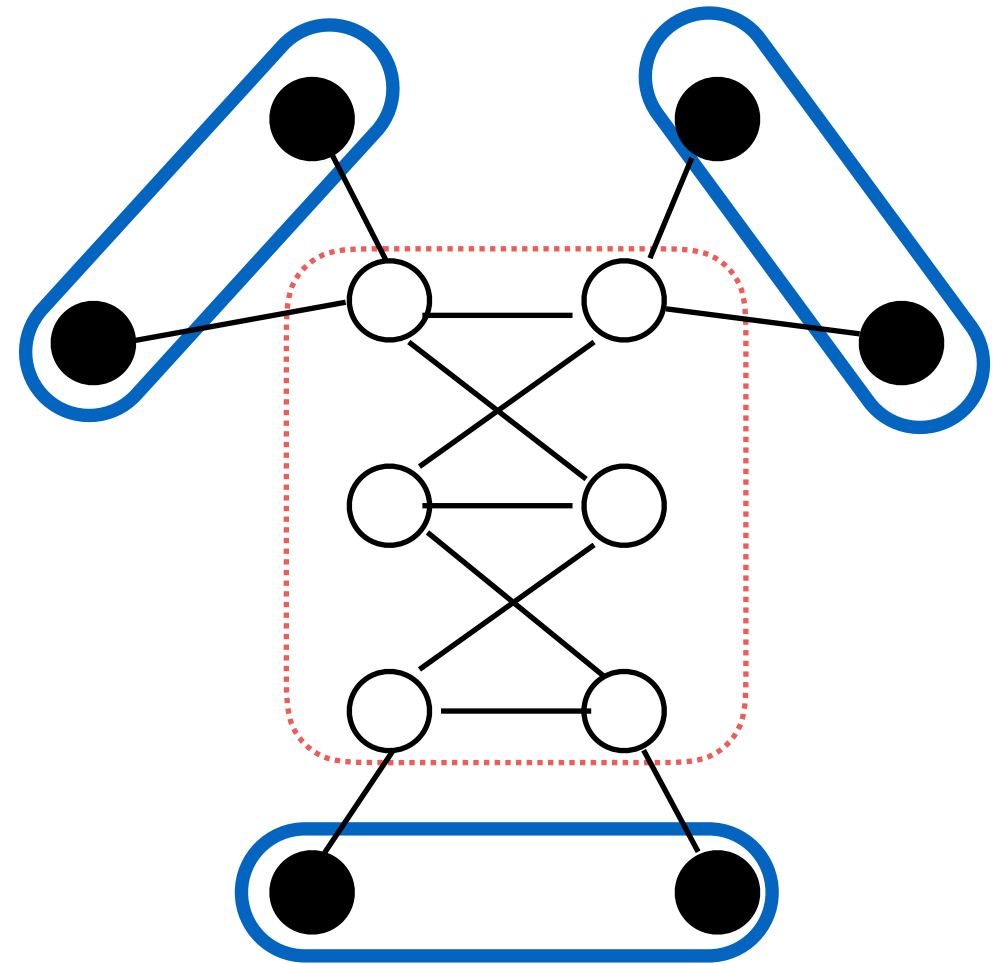
3 of 4 clusters enriched (75%)



5 of 8 clusters enriched (63%)

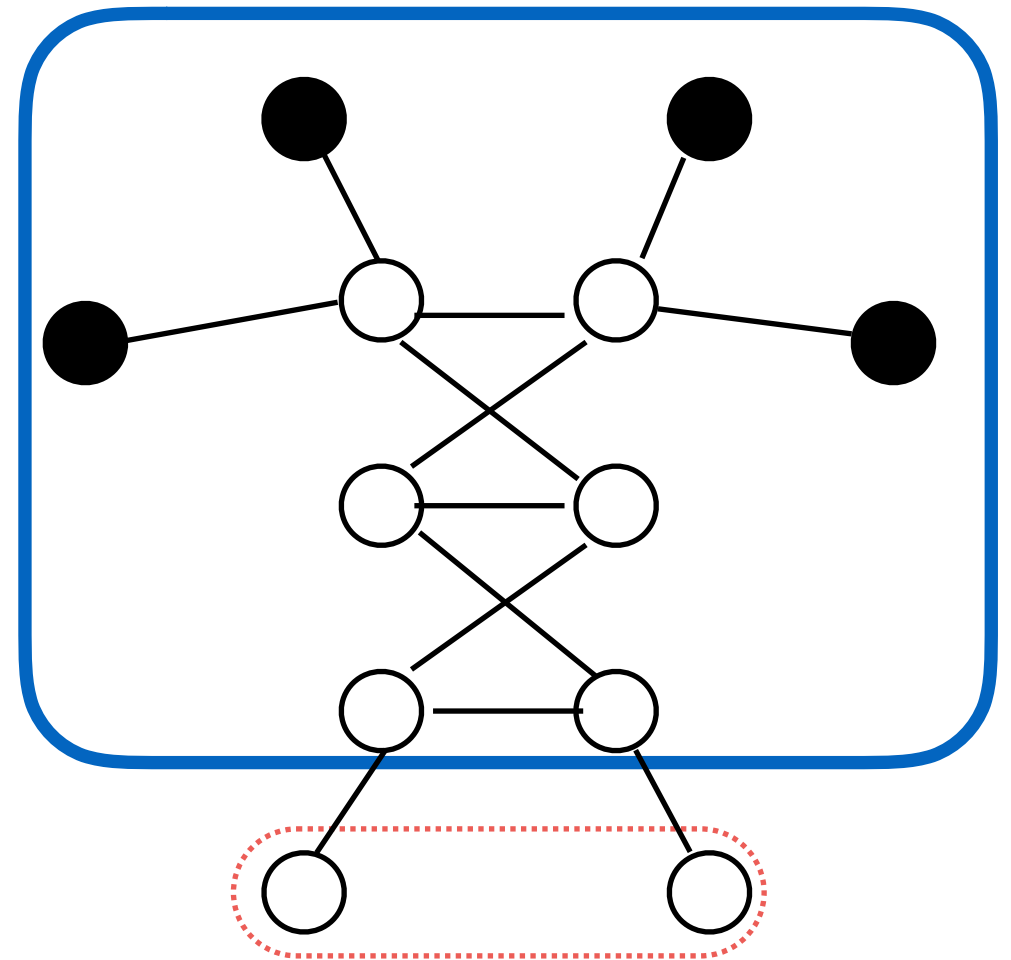
Idea 1: % of clusters enriched

- Gives a way to compare a set of communities in a global sense
- But: favors many small, peripheral clusters (or few total clusters)

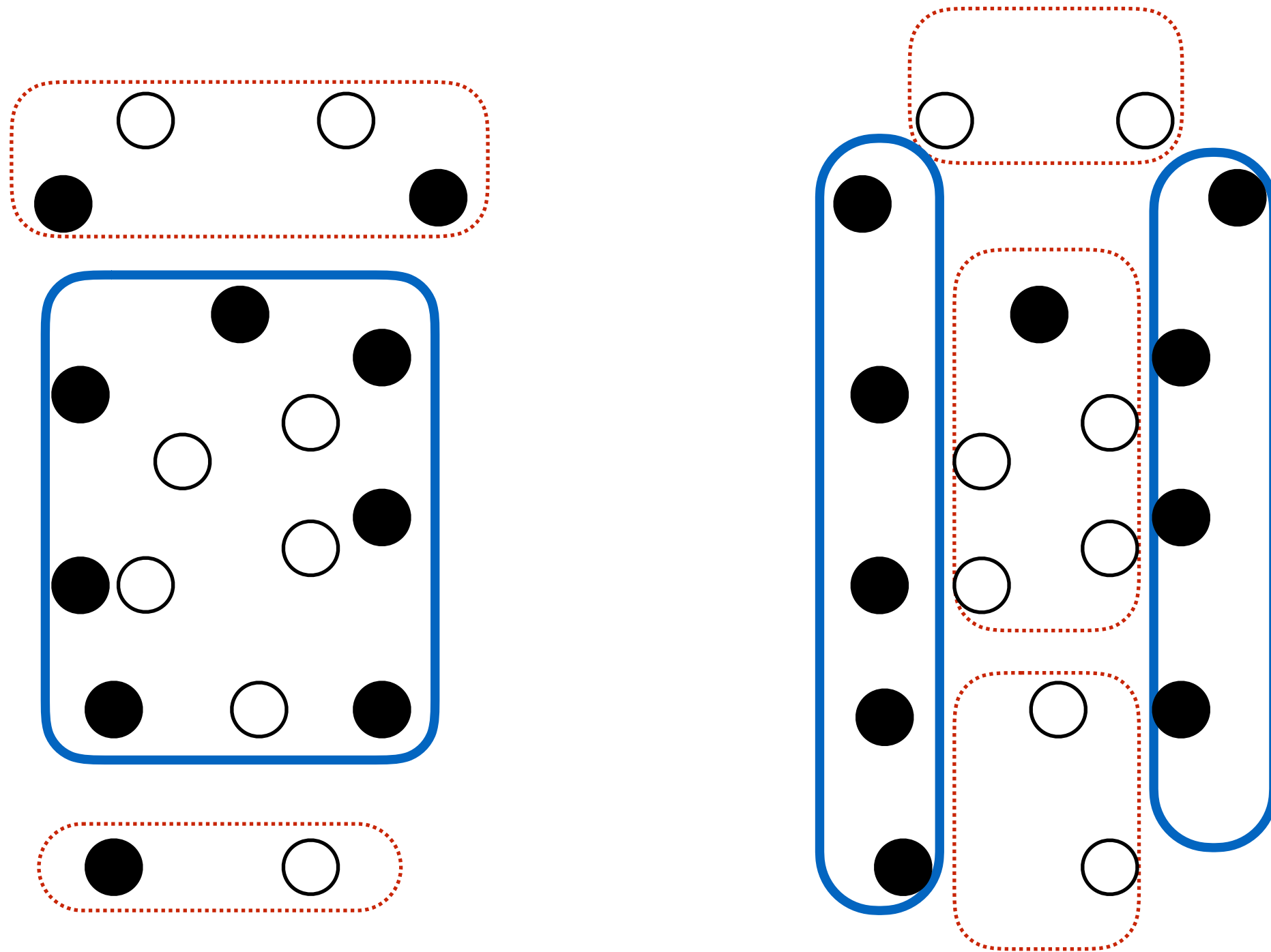


Idea 2: % nodes in enriched clusters

- Addresses the issue with many small clusters

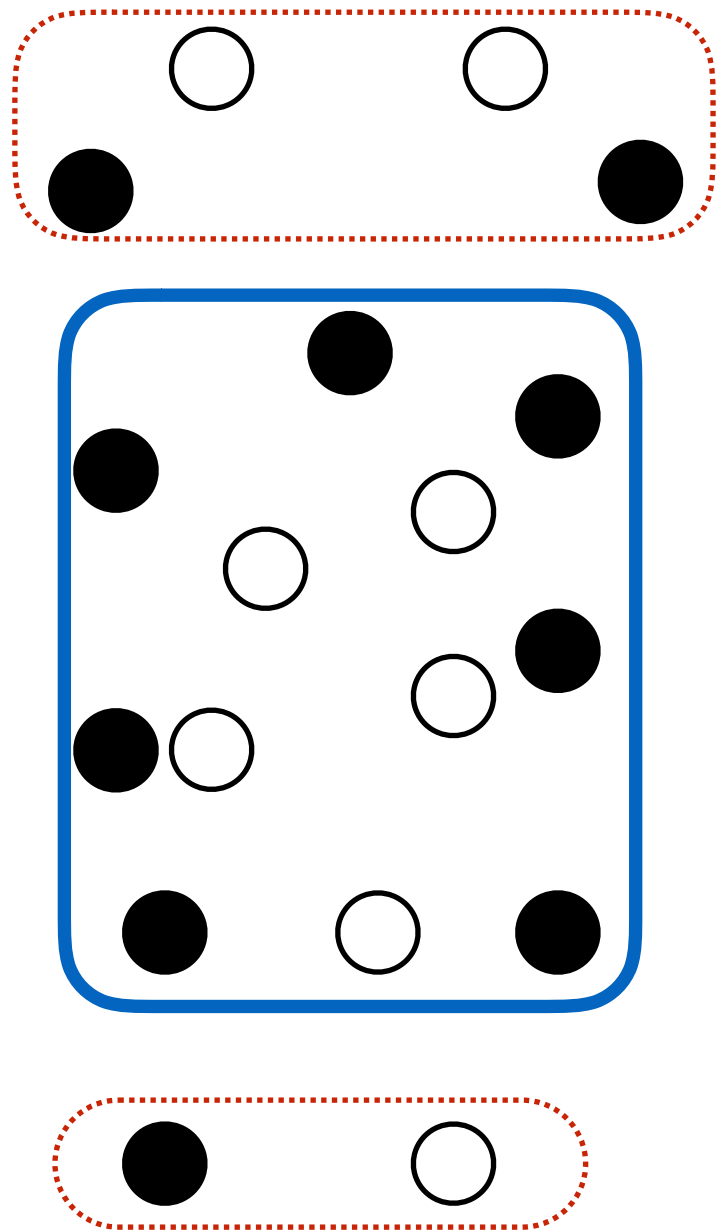


Idea 2: % of nodes in enriched clusters

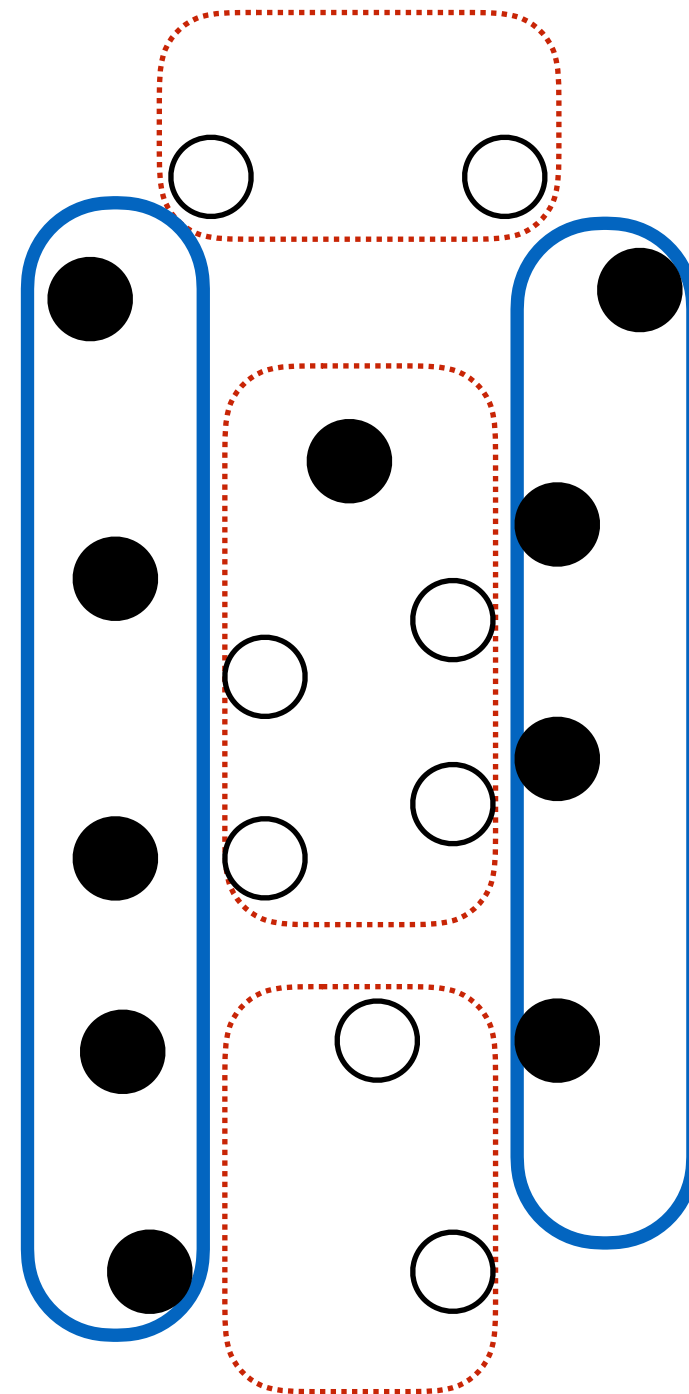


● = “annotated with function f ”

Idea 2: % of nodes in enriched clusters



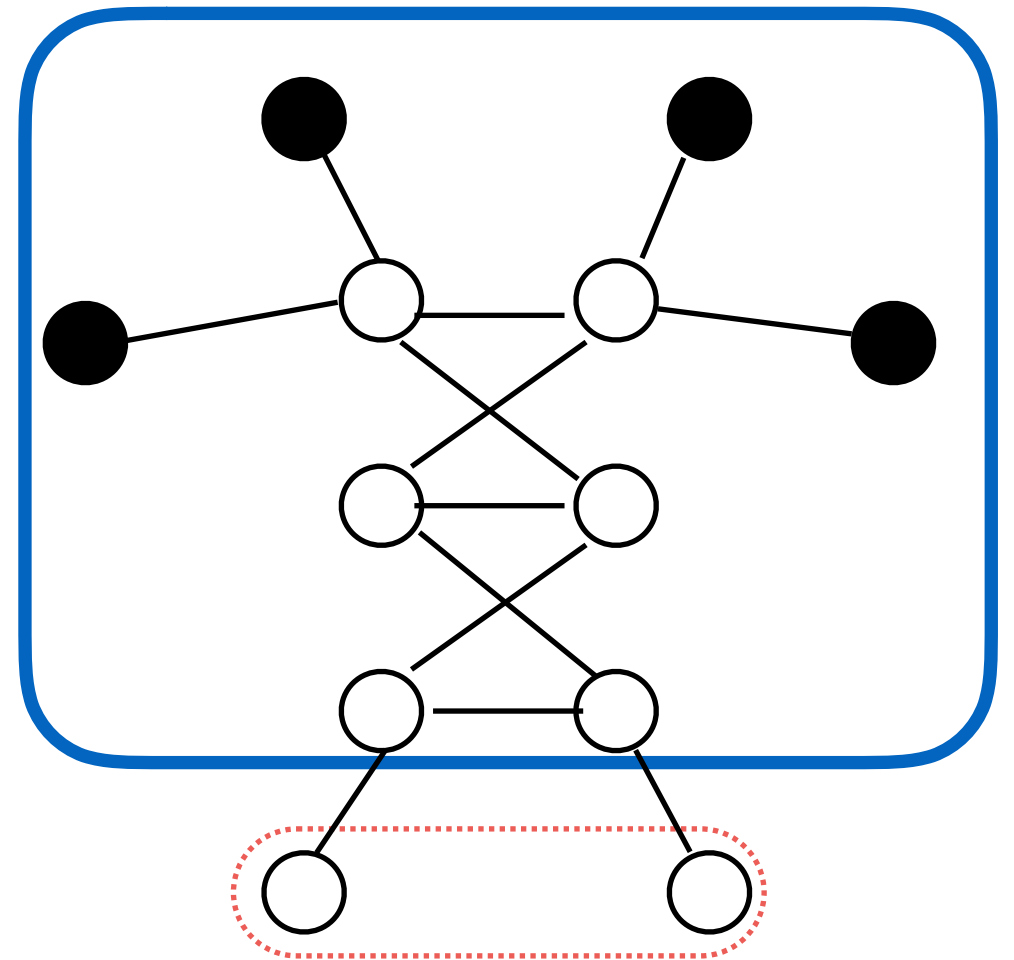
12 of 18 nodes in enriched clusters (67%)



9 of 18 nodes in enriched clusters (50%)

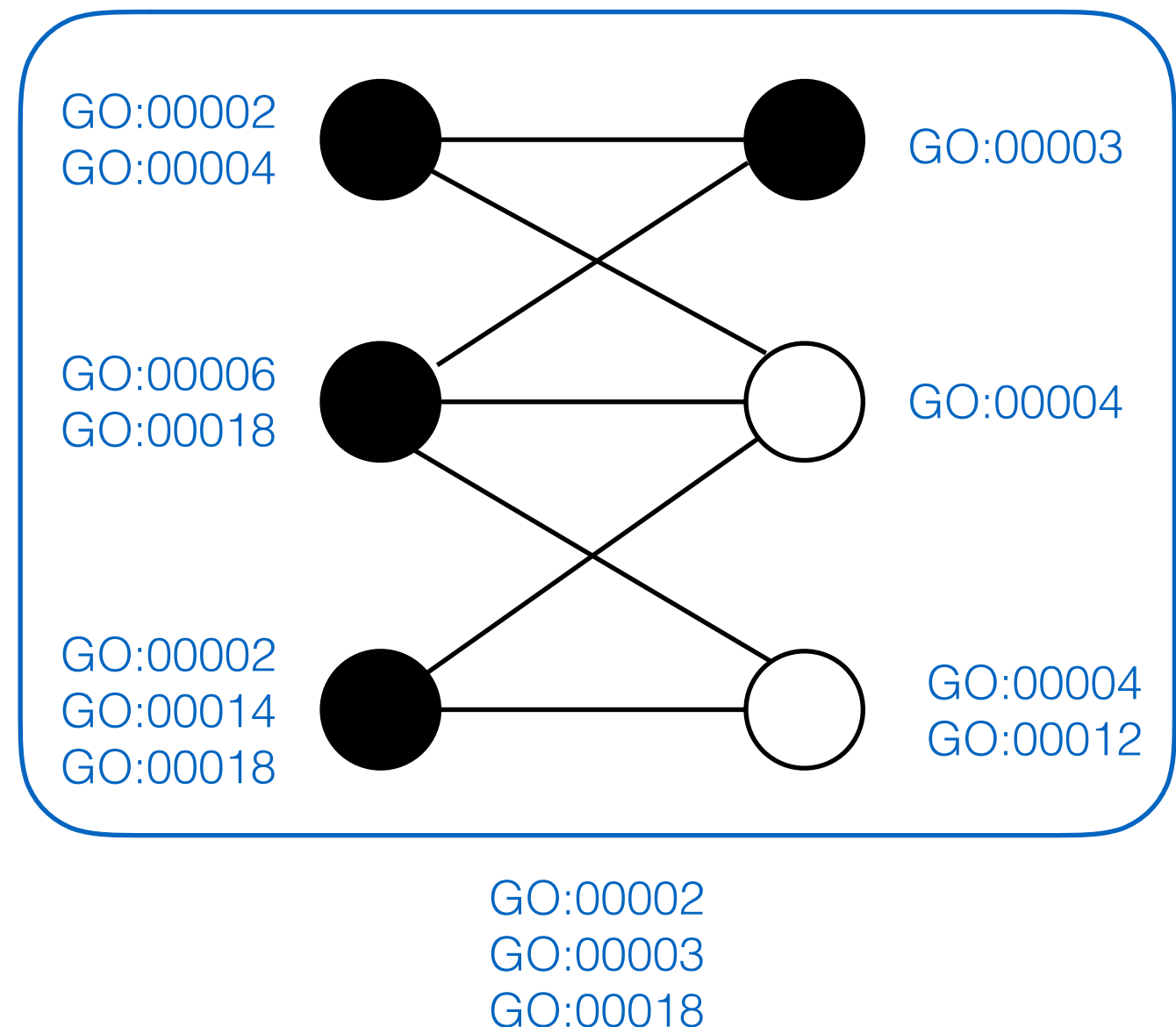
Idea 2: % nodes in enriched clusters

- Addresses the issue with many small clusters
- But: favors large clusters (in which many nodes may not be relevant, or may not “belong” in the cluster)



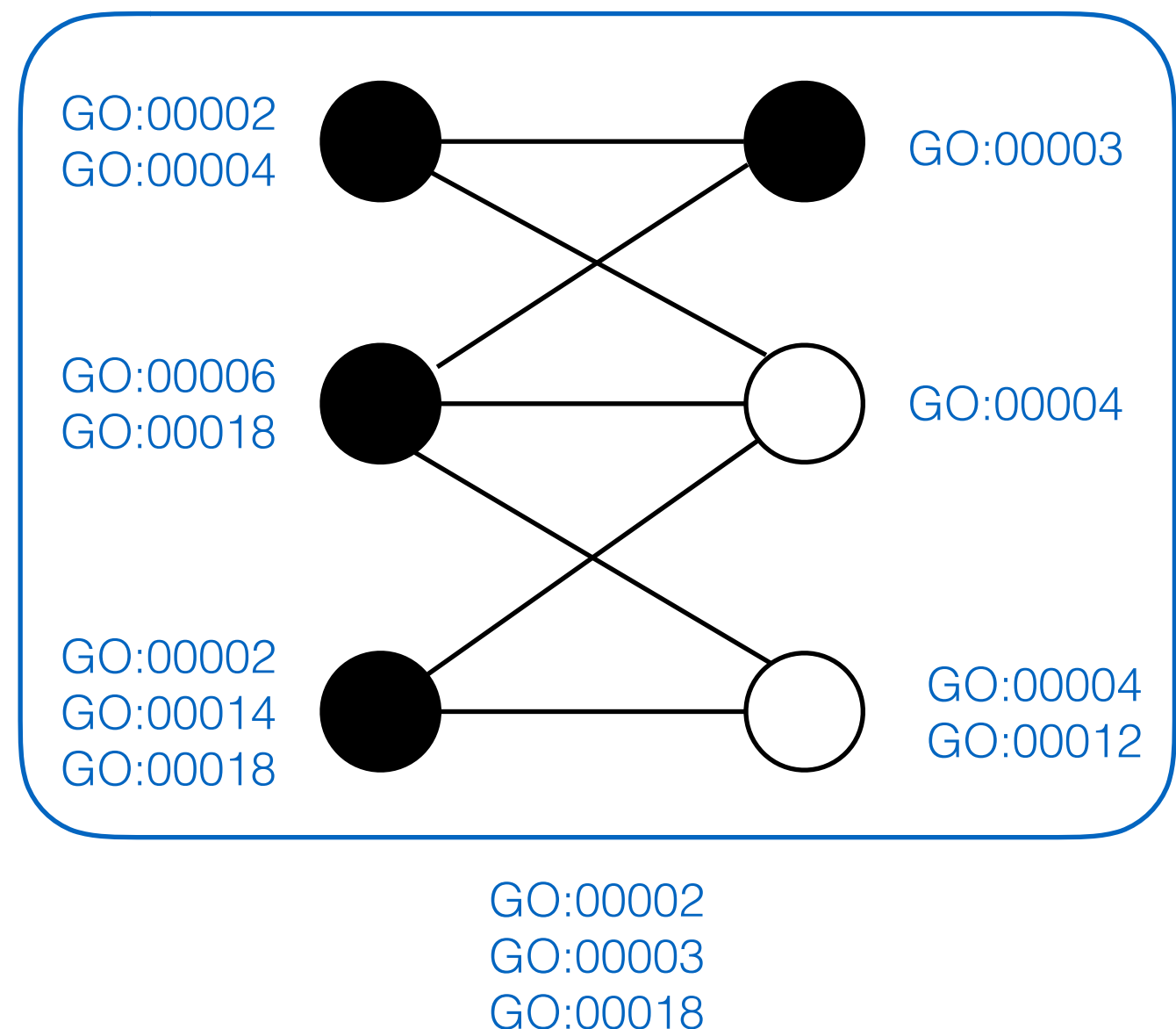
Idea 3: % “correctly clustered” nodes

- Count a node as “correctly clustered” if one of its annotations matches an annotation the cluster is enriched for as a whole



Idea 3: % “correctly clustered” nodes

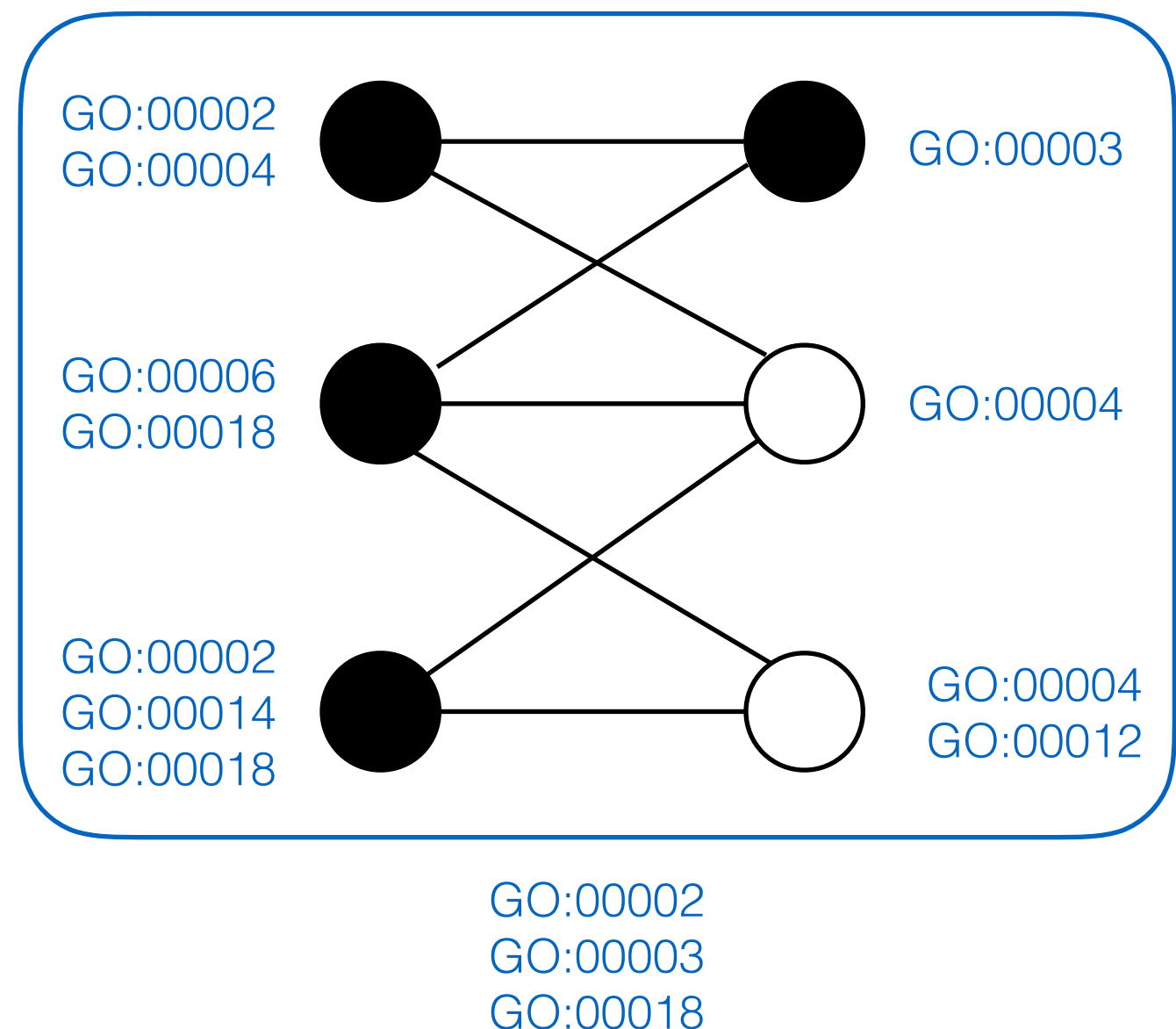
- Count a node as “correctly clustered” if one of its annotations matches an annotation the cluster is enriched for as a whole



4 of 6 nodes correctly clustered (67%)

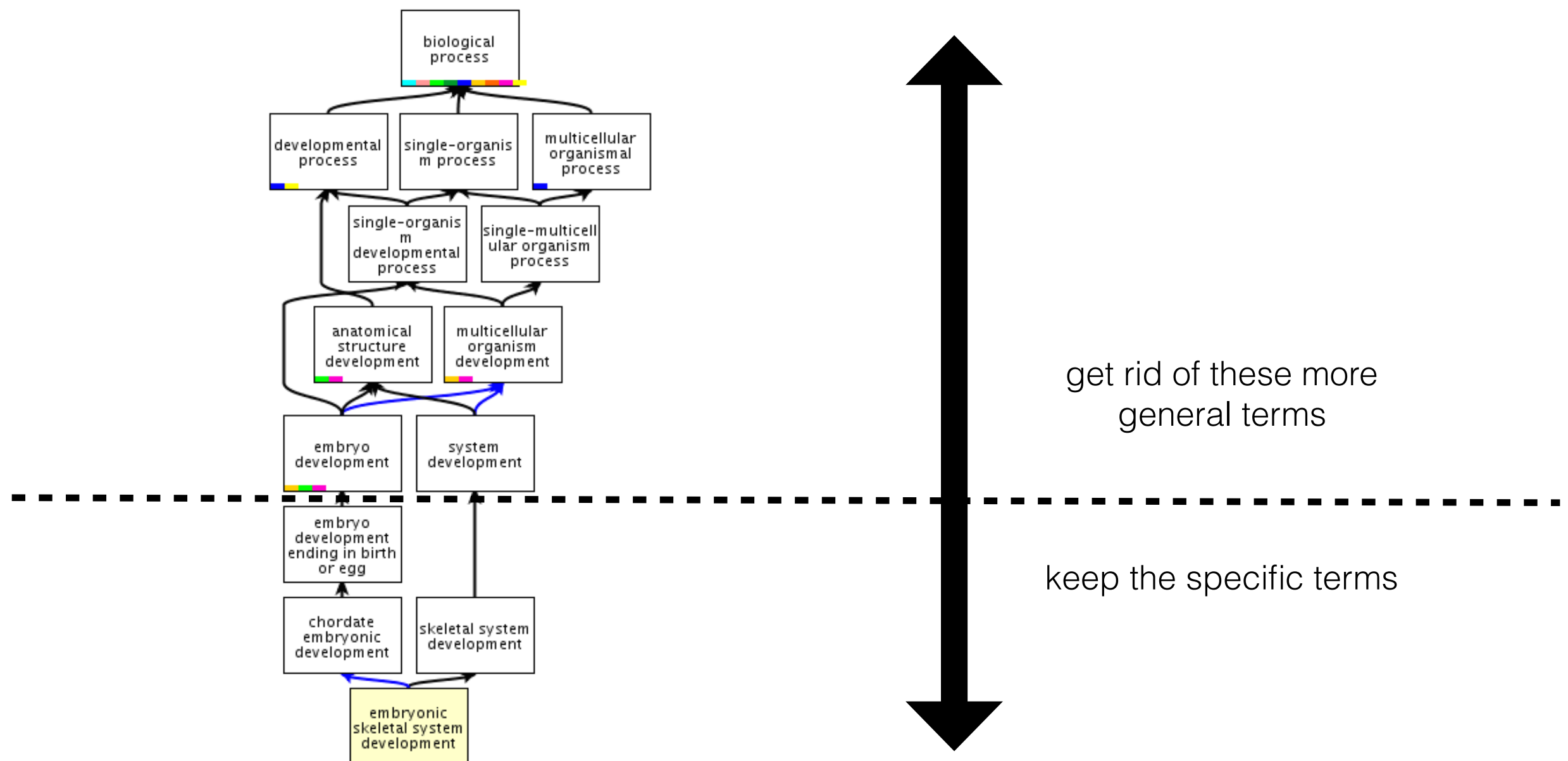
Idea 3: % “correctly clustered” nodes

- Addresses problem of irrelevant nodes in large clusters
- But: can be skewed by missing data, incorrect annotations, etc.



We looked at all 3 of the preceding measures. But, this still includes many general, uninformative annotations.

So...we also looked at enrichment limited to the 5th level of the GO and below.



Redefining neighbors

- Now, we have a way to compare partitions
- We run Louvain directly on the PPI, and compare the resulting partition to what we get when we draw a new, “detangled” graph where nodes of DSD distance $< t$ are considered neighbors.
- We try $t = 4, 4.5, 5, 5.5, 6$.

Results:

Louvain method w/ no preprocessing
(run directly on PPI network)

Method	Enriched Clusters	Total Clusters	% Enriched	# NEC	% NEC U	% NEC F	% NEC S
PPI	90.5	382	23.69%	1901	31.17%	26.03%	11.26%
4.0	147	192.5	76.36%	1355	22.23%	21.78%	19.39%
4.5	214	305	70.16%	2369.5	38.87%	35.11%	24.89%
5.0	161.5	352	45.88%	2965.5	48.65%	37.11%	16.94%
5.5	87	227.5	38.24%	3643.5	59.77%	32.70%	8.10%
6.0	55.5	180.5	30.75%	2740	44.96%	27.84%	5.26%

Louvain method detangled w/ DSD
distance (values of t)

Results:

Percent of clusters enriched ("Idea 1")



Method	Enriched Clusters	Total Clusters	% Enriched	# NEC	% NEC U	% NEC F	% NEC S
PPI	90.5	382	23.69%	1901	31.17%	26.03%	11.26%
4.0	147	192.5	76.36%	1355	22.23%	21.78%	19.39%
4.5	214	305	70.16%	2369.5	38.87%	35.11%	24.89%
5.0	161.5	352	45.88%	2965.5	48.65%	37.11%	16.94%
5.5	87	227.5	38.24%	3643.5	59.77%	32.70%	8.10%
6.0	55.5	180.5	30.75%	2740	44.96%	27.84%	5.26%

Results:

Nodes in enriched clusters



Method	Enriched Clusters	Total Clusters	% Enriched	# NEC	% NEC U	% NEC F	% NEC S
PPI	90.5	382	23.69%	1901	31.17%	26.03%	11.26%
4.0	147	192.5	76.36%	1355	22.23%	21.78%	19.39%
4.5	214	305	70.16%	2369.5	38.87%	35.11%	24.89%
5.0	161.5	352	45.88%	2965.5	48.65%	37.11%	16.94%
5.5	87	227.5	38.24%	3643.5	59.77%	32.70%	8.10%
6.0	55.5	180.5	30.75%	2740	44.96%	27.84%	5.26%

Results:

Method	Enriched Clusters	Total Clusters	% Enriched	# NEC	% NEC U	% NEC F	% NEC S
PPI	90.5	382	23.69%	1901	31.17%	26.03%	11.26%
4.0	147	192.5	76.36%	1355	22.23%	21.78%	19.39%
4.5	214	305	70.16%	2369.5	38.87%	35.11%	24.89%
5.0	161.5	352	45.88%	2965.5	48.65%	37.11%	16.94%
5.5	87	227.5	38.24%	3643.5	59.77%	32.70%	8.10%
6.0	55.5	180.5	30.75%	2740	44.96%	27.84%	5.26%

Percent of nodes in enriched clusters, with no GO level filtering ("Idea 2")

Results:

Percent of nodes in enriched clusters, *with* filtering of terms above the 5th GO level



Method	Enriched Clusters	Total Clusters	% Enriched	# NEC	% NEC U	% NEC F	% NEC S
PPI	90.5	382	23.69%	1901	31.17%	26.03%	11.26%
4.0	147	192.5	76.36%	1355	22.23%	21.78%	19.39%
4.5	214	305	70.16%	2369.5	38.87%	35.11%	24.89%
5.0	161.5	352	45.88%	2965.5	48.65%	37.11%	16.94%
5.5	87	227.5	38.24%	3643.5	59.77%	32.70%	8.10%
6.0	55.5	180.5	30.75%	2740	44.96%	27.84%	5.26%

Results:

Method	Enriched Clusters	Total Clusters	% Enriched	# NEC	% NEC U	% NEC F	% NEC S
PPI	90.5	382	23.69%	1901	31.17%	26.03%	11.26%
4.0	147	192.5	76.36%	1355	22.23%	21.78%	19.39%
4.5	214	305	70.16%	2369.5	38.87%	35.11%	24.89%
5.0	161.5	352	45.88%	2965.5	48.65%	37.11%	16.94%
5.5	87	227.5	38.24%	3643.5	59.77%	32.70%	8.10%
6.0	55.5	180.5	30.75%	2740	44.96%	27.84%	5.26%



Percent of nodes in enriched clusters that have a *correct label*, with filtering of terms above the 5th GO level ("Idea 3")

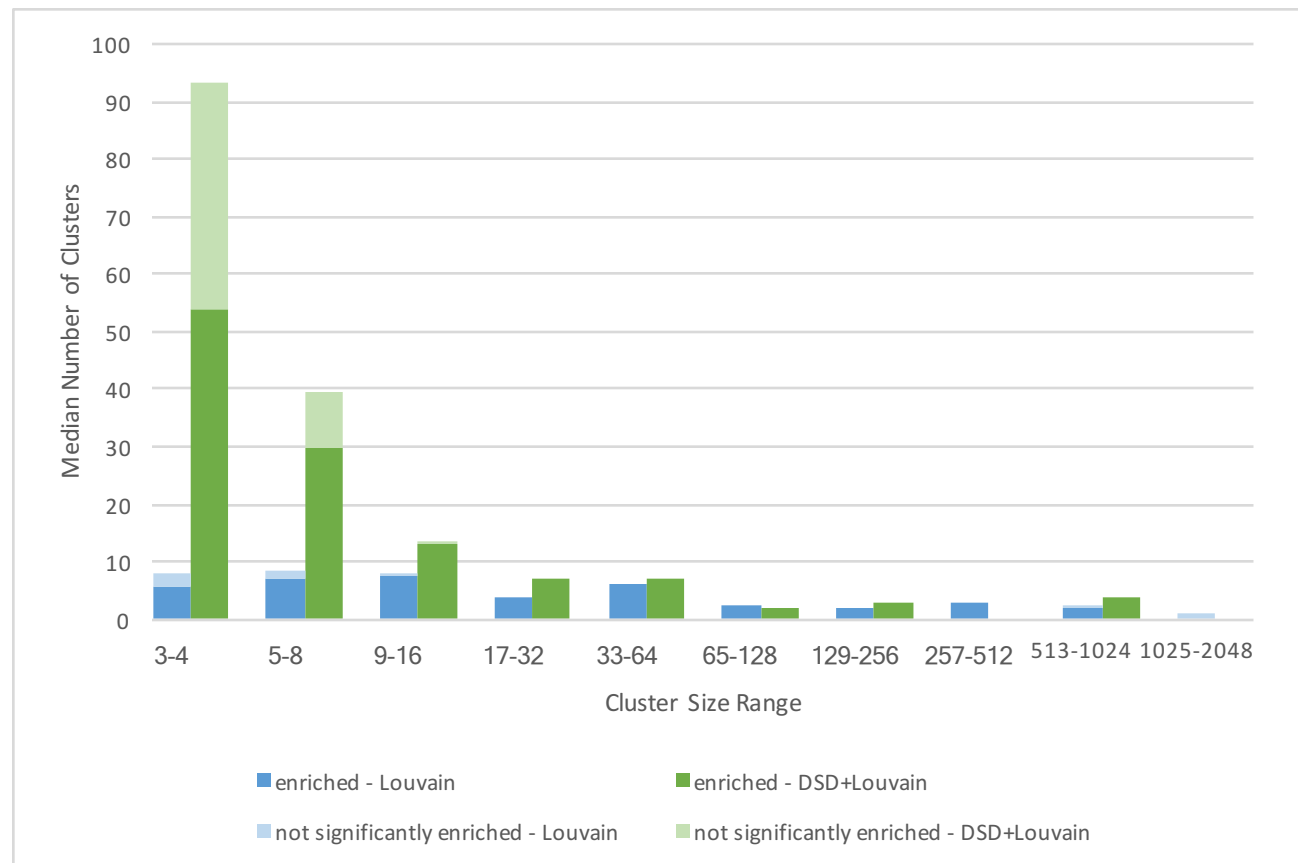
Results: (for Louvain algorithm, with cluster sizes between 3-100)

PPI only

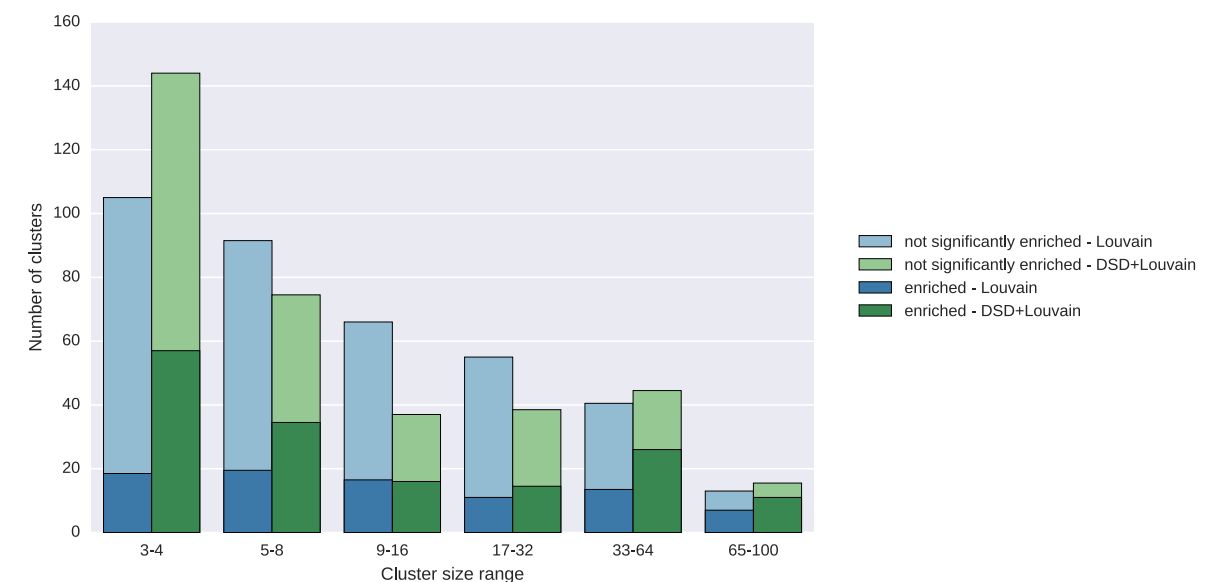
Method	Enriched Clusters	Total Clusters	% Enriched	# NEC	% NEC U	% NEC F	% NEC S
PPI	90.5	382	23.69%	1901	31.17%	26.03%	11.26%
4.0	147	192.5	76.36%	1355	22.23%	21.78%	19.39%
4.5	214	305	70.16%	2369.5	38.87%	35.11%	24.89%
5.0	161.5	352	45.88%	2965.5	48.65%	37.11%	16.94%
5.5	87	227.5	38.24%	3643.5	59.77%	32.70%	8.10%
6.0	55.5	180.5	30.75%	2740	44.96%	27.84%	5.26%

PPI + DSD

We also ran experiments without limiting cluster size, but we often get much different size distributions.



Louvain, no size restriction



Louvain, cluster sizes between 3-100

Results: (for spectral clustering algorithm, with cluster sizes between 3-100)

PPI only

Method	Enriched Clusters	Total Clusters	% Enriched	# NEC	% NEC U	% NEC F	% NEC S
PPI	262	324	80.86%	3743	61.38%	55.59%	40.13%
DSD 4.5	208	266	78.19%	1776	29.13%	29.42%	24.07%
DSD 5.0	222	309	71.84%	4143	67.96%	66.48%	43.44%
DSD 5.5	222	291	76.29%	4771	78.26%	72.91%	46.05%
DSD 6.0	204	249	81.93%	5172	84.84%	80.99%	44.79%

PPI + DSD

Results: (for Walktrap algorithm, with cluster sizes between 3-100)

PPI only

Method	Enriched Clusters	Total Clusters	% Enriched	# NEC	% NEC U	% NEC F	% NEC S
PPI	61	64	95.31%	5692	93.34%	91.67%	54.41%
DSD 4.0	111	142	78.17%	1493	24.49%	24.21%	19.90%
DSD 4.5	173	215	80.47%	3108	50.98%	48.47%	38.53%
DSD 5.0	126	174	72.41%	4950	81.20%	79.04%	53.82%
DSD 5.5	76	93	81.72%	5780	94.82%	91.67%	48.57%
DSD 6.0	70	81	86.42%	5691	93.36%	90.86%	41.85%

PPI + DSD

Future Work

- Problem is somewhat artificial (non-overlapping clusters, size limitations) - we'd like to generalize
- We did a small pilot study on the human STRING network and results looked similar, could try other species and types of networks (coexpression, regulatory, etc.)
- Experiment with other ways to control/correct for varying size distributions

Slides will be available shortly on my website:

<http://jjc2718.github.io>

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