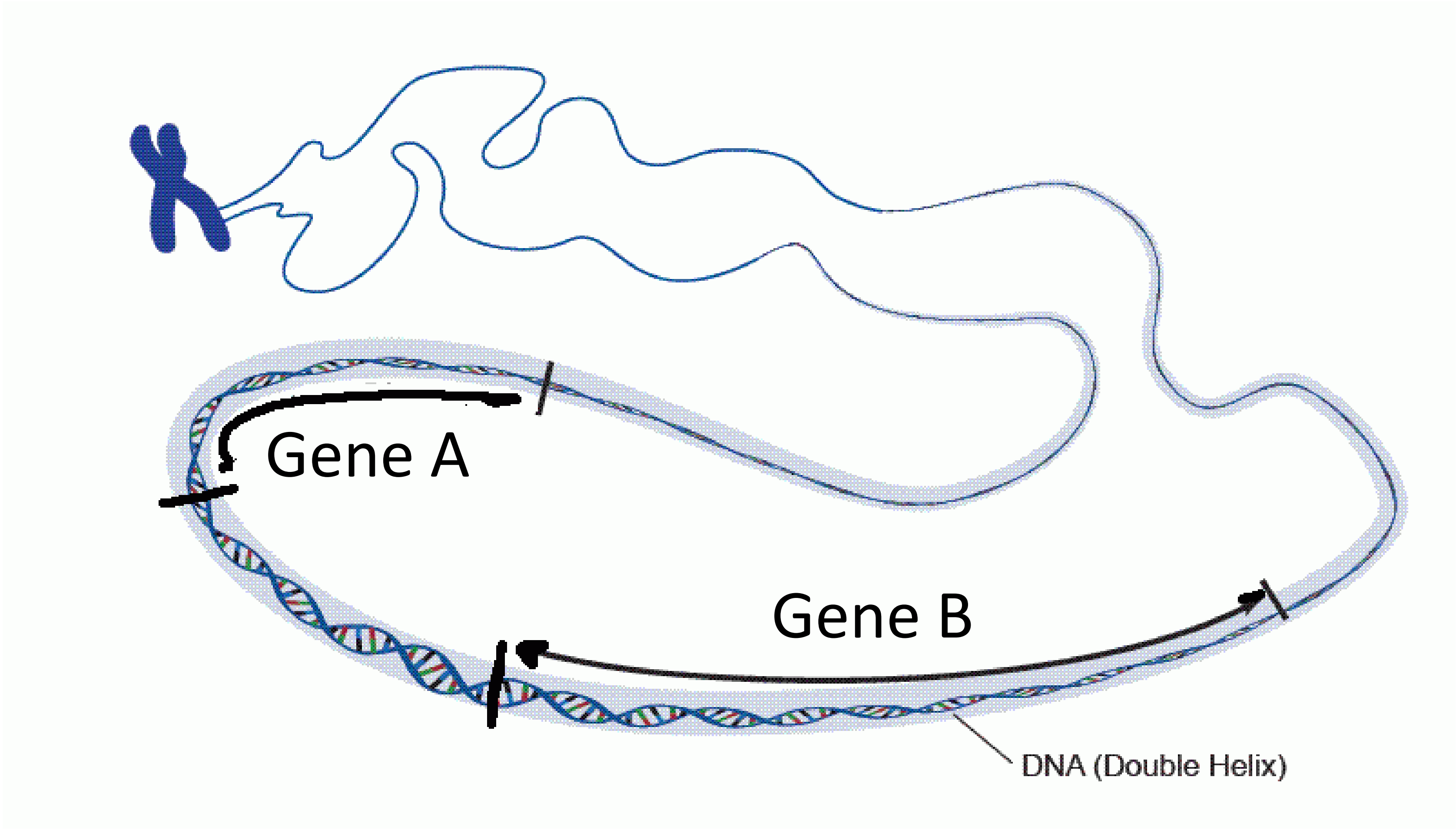
Inference of Directed Acyclic Graphs Using Spectral Clustering

Allison Paul

Fifth Annual MIT PRIMES Conference

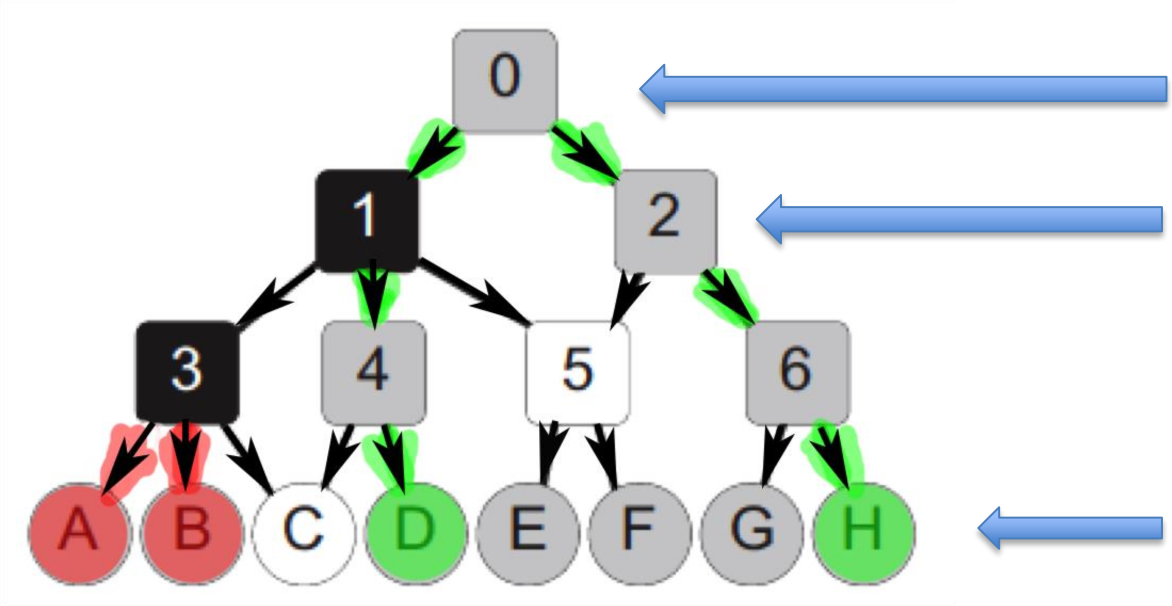
May 17, 2015

Introduction



Genes A and B are involved in the same process

Gene Ontology (GO)



Gene Ontology Terms

Genes

Examples of Gene Ontology Terms: oxygen binding, response to x-ray, sympathetic nervous system development

This type of network is a **directed acyclic graph (DAG)**

Goal: Infer this graph using gene similarities

What is gene similarity?

Functional similarity: gene expression

Physical similarity

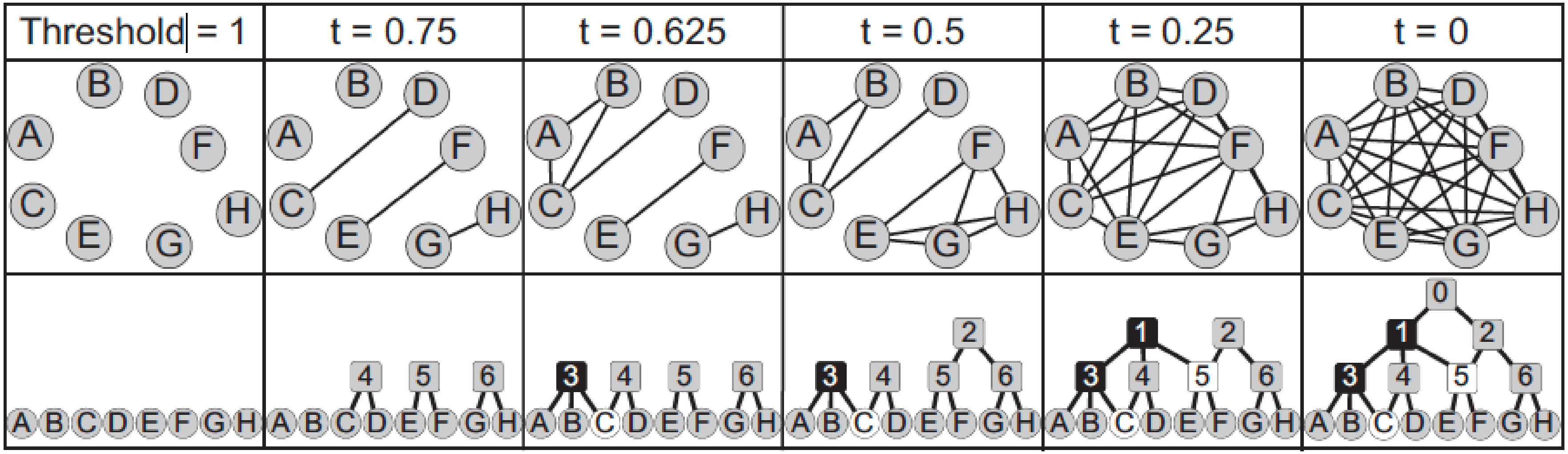
Problem Statement: Given a gene similarity matrix, find the directed acyclic graph

Inferring such a graph using a gene similarity matrix is NP-hard in general.

Current Method

Bottom-up algorithm using maximal cliques (Kramer et al. 2014)

Clique: a subset of nodes in which each pair of nodes is connected by an edge

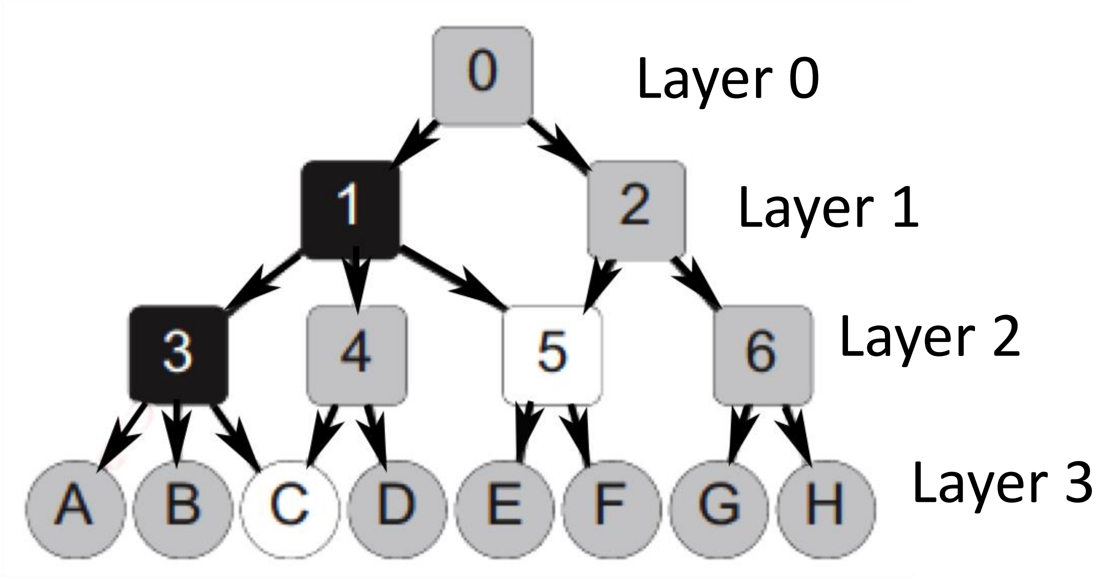


Computational complexity: $\mathcal{O}(3^{n/3})$

Our Approach

We propose an approximate algorithm that finds quasi-cliques among the genes

Top-Down Algorithm: we infer nodes at layer ℓ using nodes at layer ℓ - 1



Layer 0

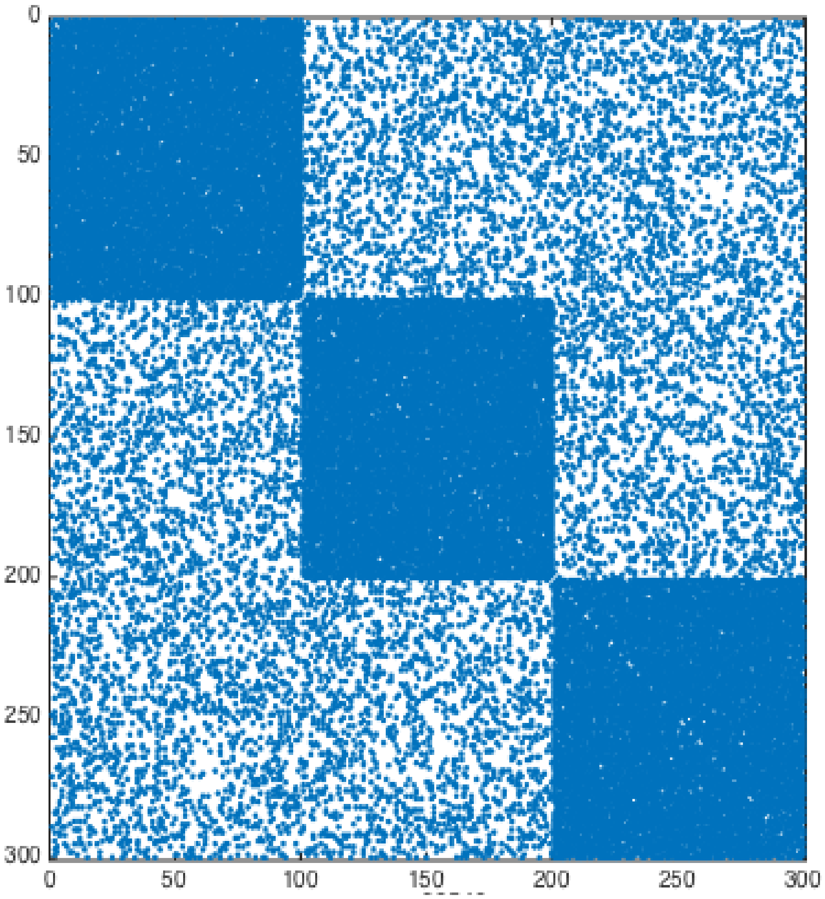
Layer 1

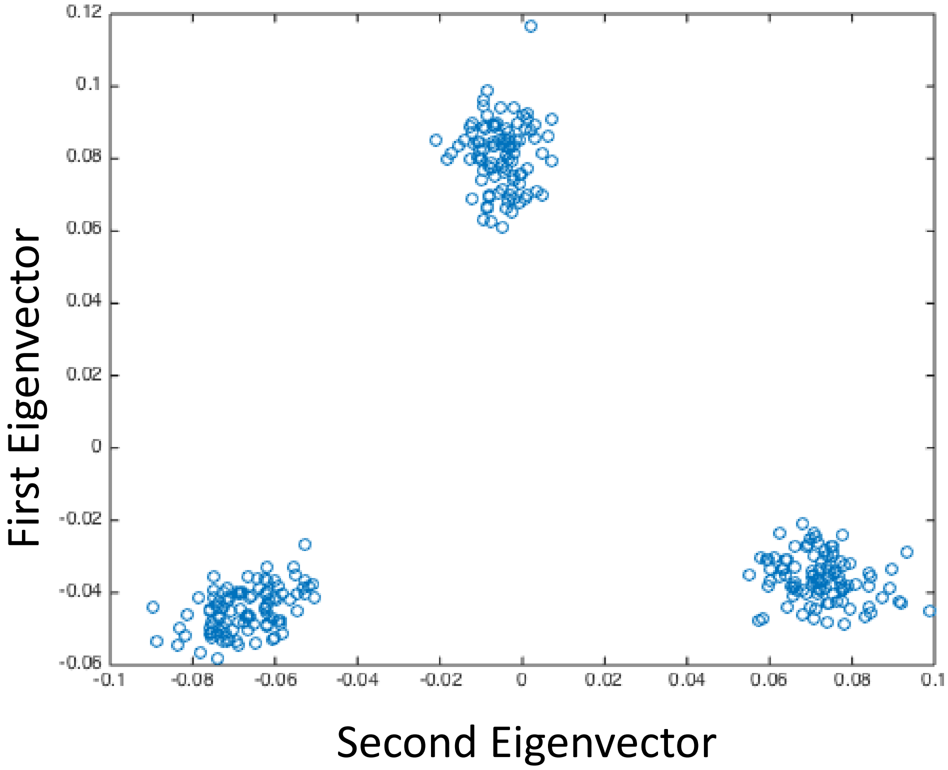
Layer 2

Layer 3

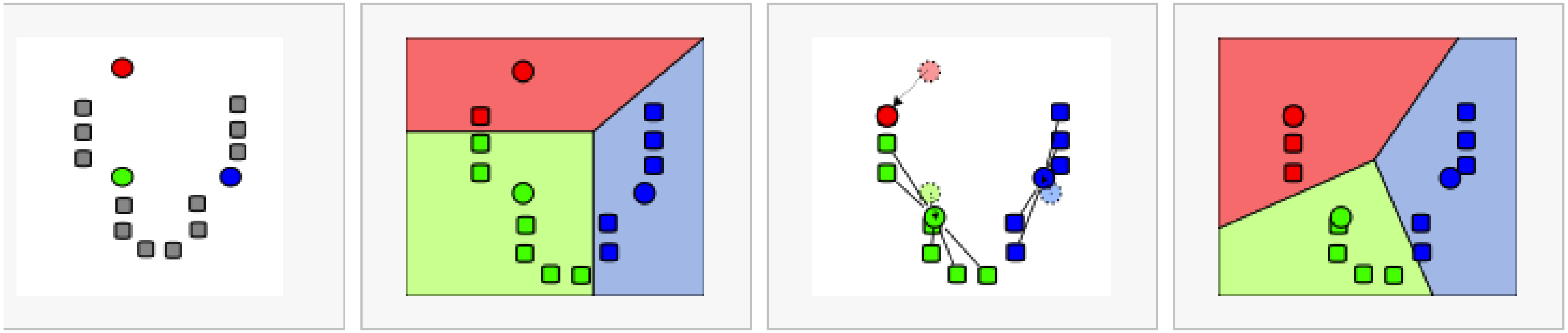
Spectral Clustering

We analyze the top k-1 eigenvectors of the similarity matrix



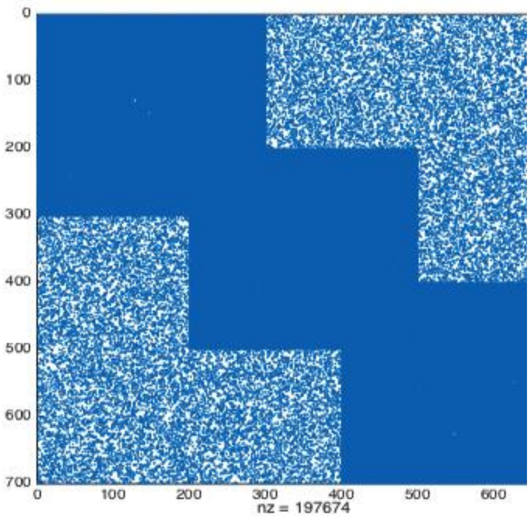


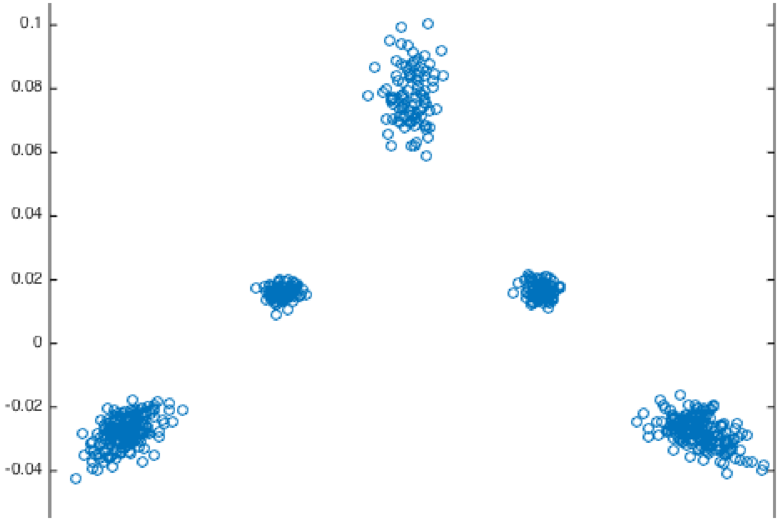
K-Means Algorithm

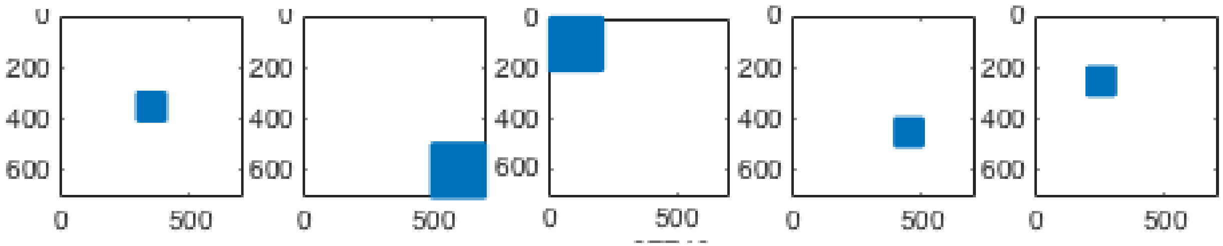


Greedy algorithm that identifies clusters among points in $\mathbf{R}^n$

Overlapping Clusters

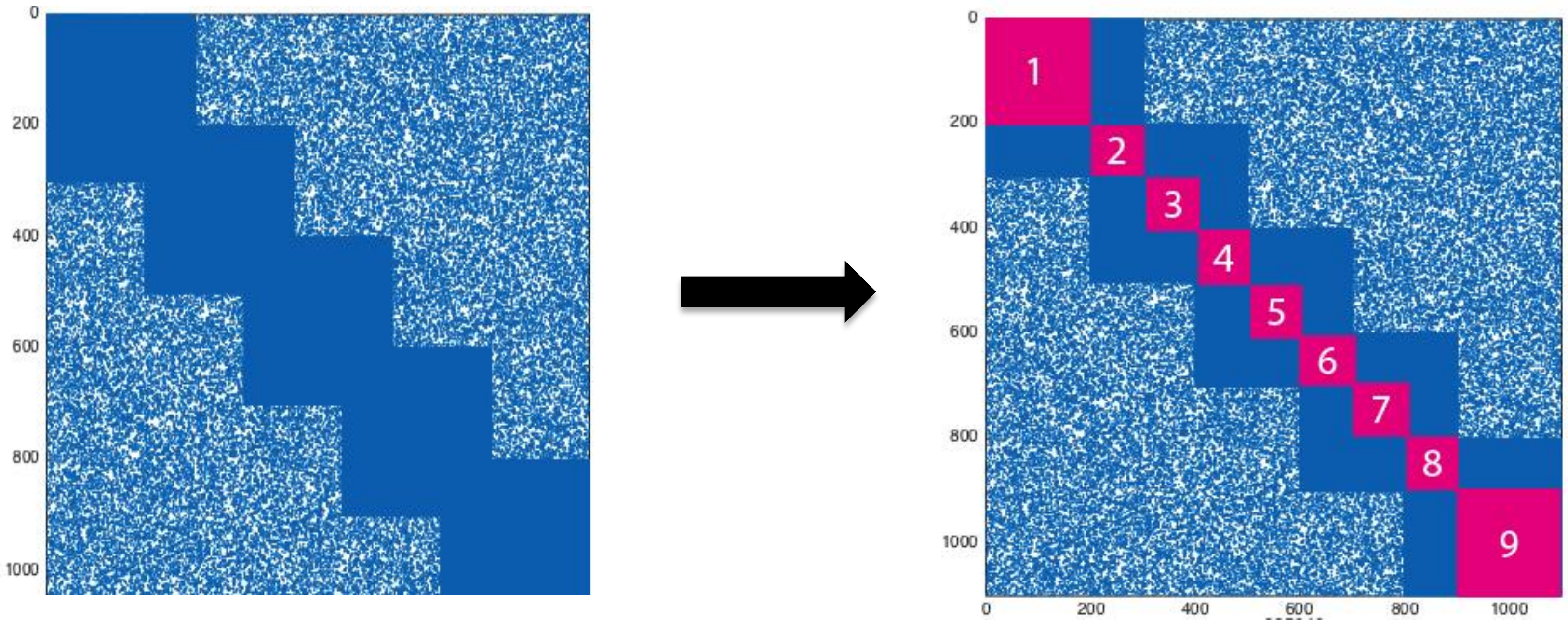






The original problem can be thus simplified to the inference problem of overlapping clusters in a network.

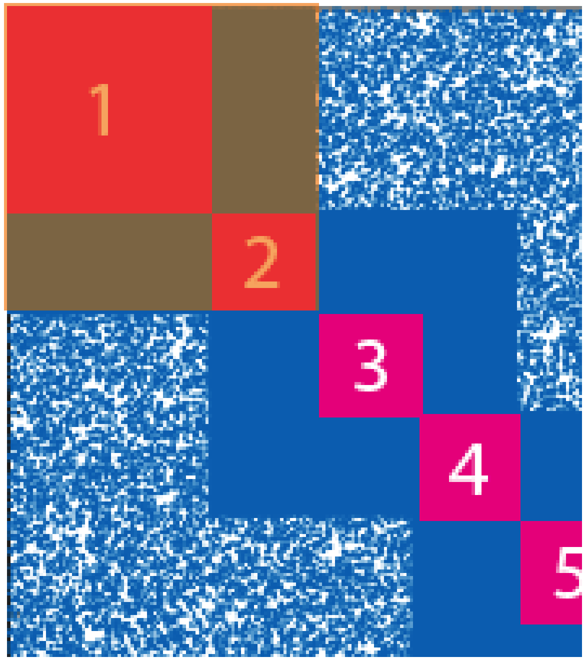
Spectral Clustering



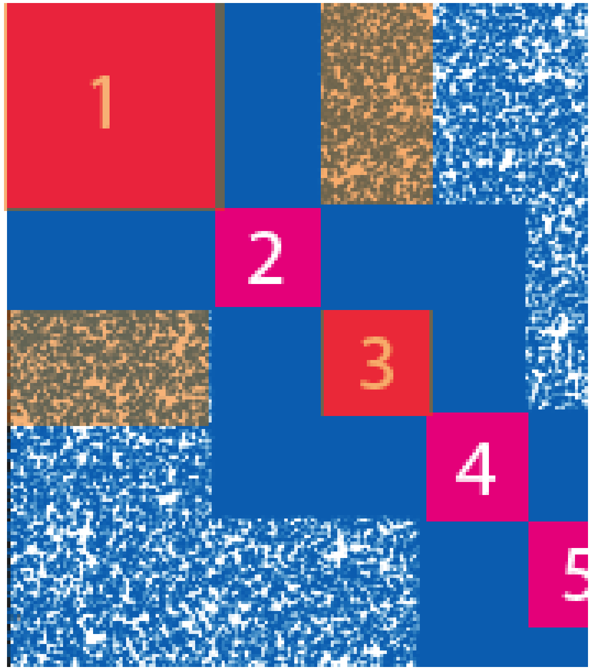
Use spectral clustering methods to partition network into k clusters

Metric for combining clusters

$W(C_A, C_B) = \text{density}(C_A U C_B) –$
$\text{average}(\text{density}(C_A), \text{density}(C_B))$



$W(C_1, C_2) = - 0.03$



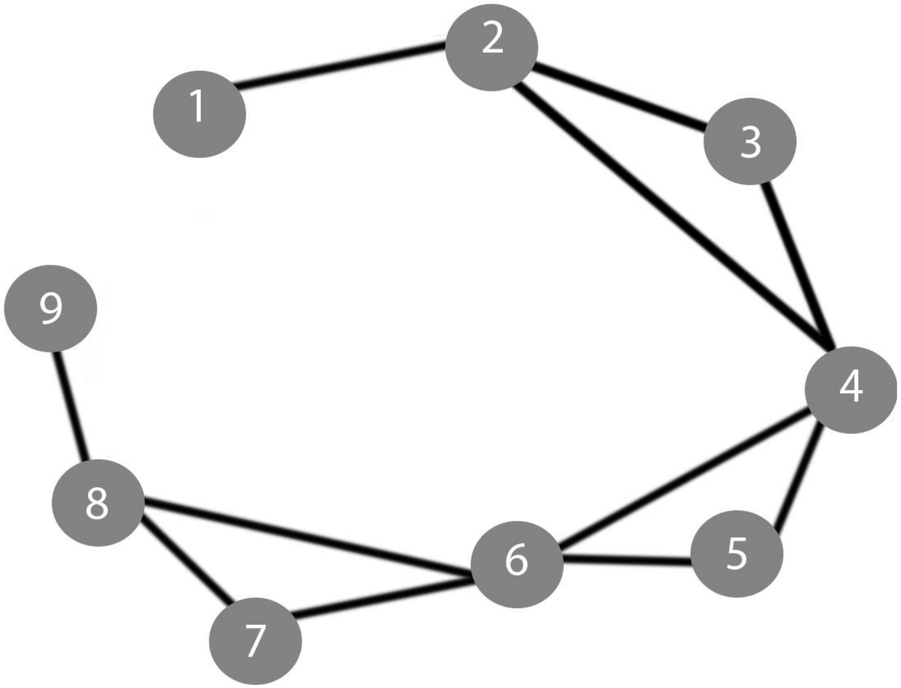
$W(C_1, C_3) = - 0.2$

Cluster Similarity Matrix

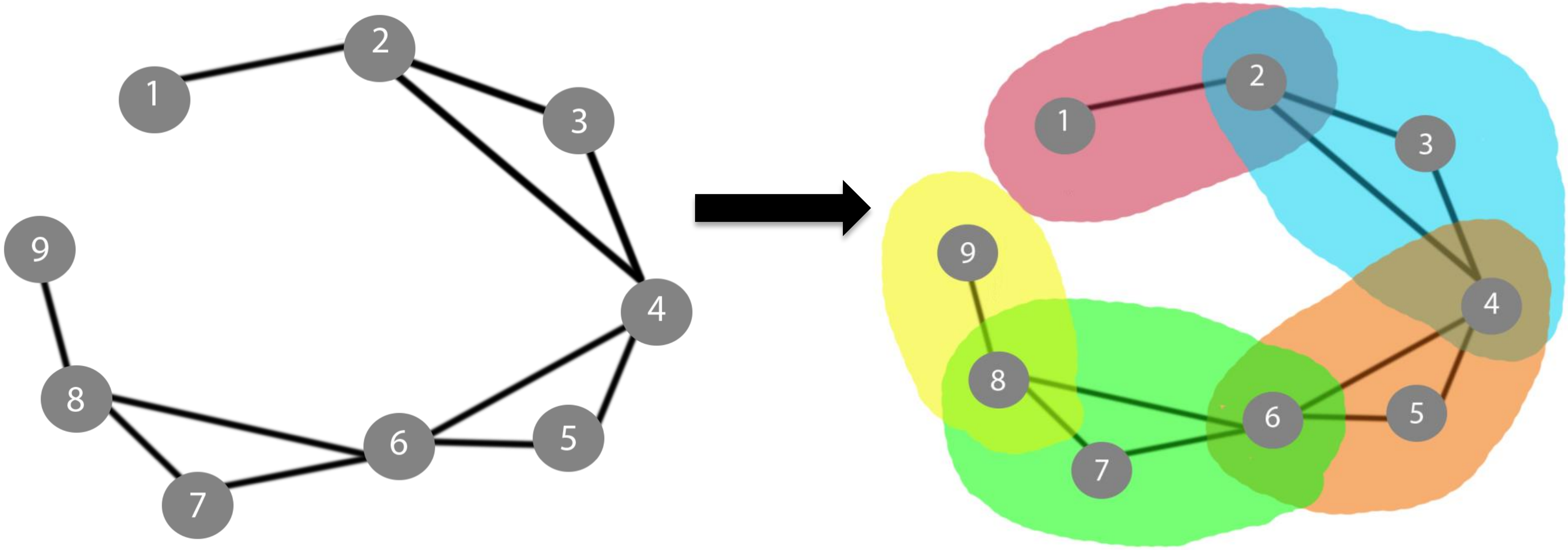
$$M_{i,j} = W(C_i, C_j)$$

	1	2	3	4	5	6	7	8	9
1	0	-.02	-.172	-.20	-.082	-.273	-.122	-.321	-.273
2	-.02	0	-.031	-.019	-.091	-.304	-.14	-.102	-.177
3	-.172	-.031	0	-.041	-.155	-.203	-.37	-.088	-.209
4	-.20	-.019	-.041	0	-.027	-.012	-.221	-.298	-.078
5	-.082	-.091	-.155	-.027	0	-.034	-.098	-.120	-.192
6	-.273	-.304	-.203	-.012	-.034	0	-.017	-.038	-.232
7	-.122	-.14	-.37	-.221	-.098	-.017	0	-.044	-.311
8	-.321	-.102	-.088	-.298	-.120	-.038	-.044	0	-.029
9	-.273	-.177	-.209	-.078	-.192	-.232	-.311	-.029	0

■ > threshold



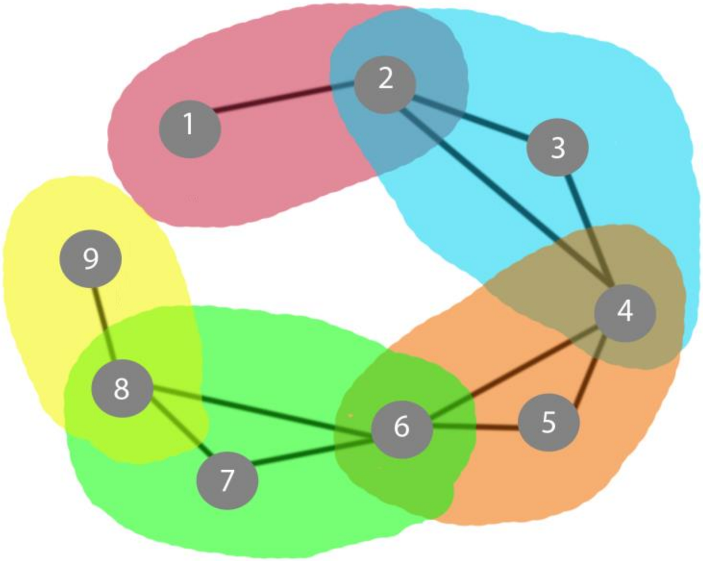
Finding Maximal Cliques

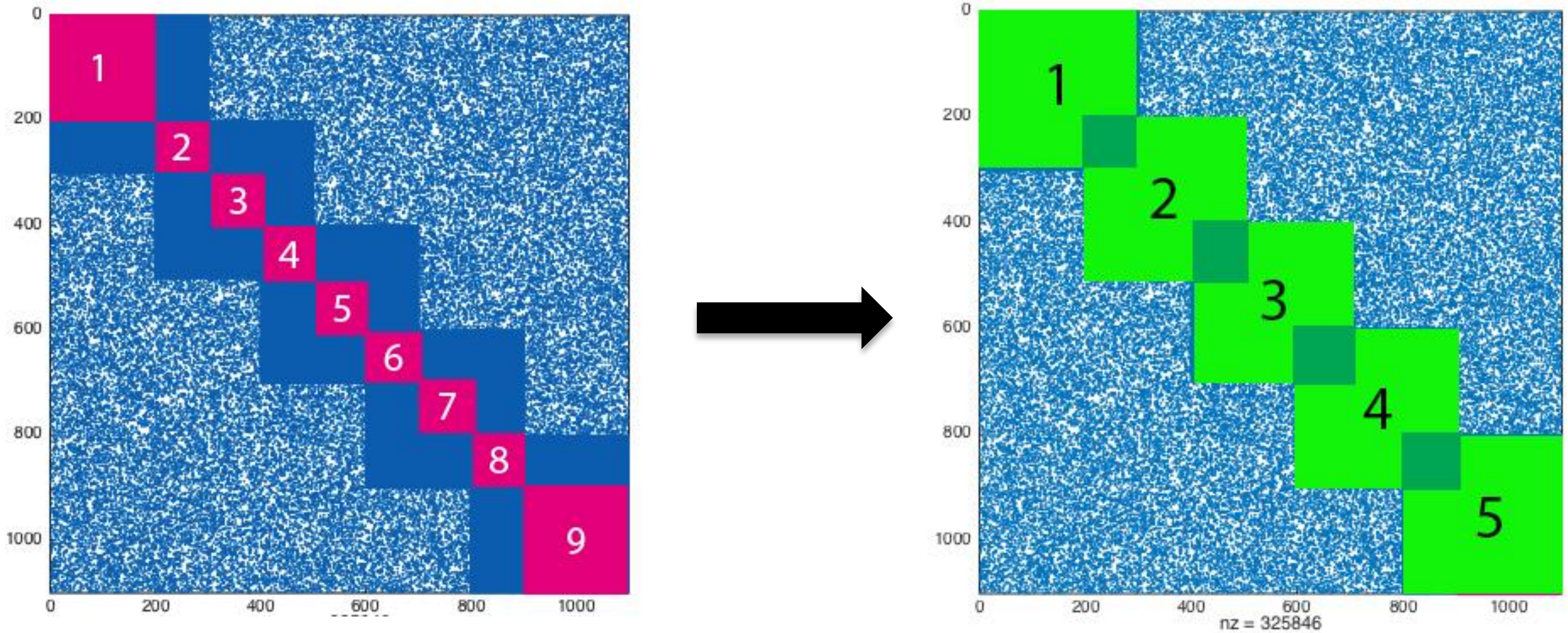


We are left with the same problem as before: identifying overlapping clusters.

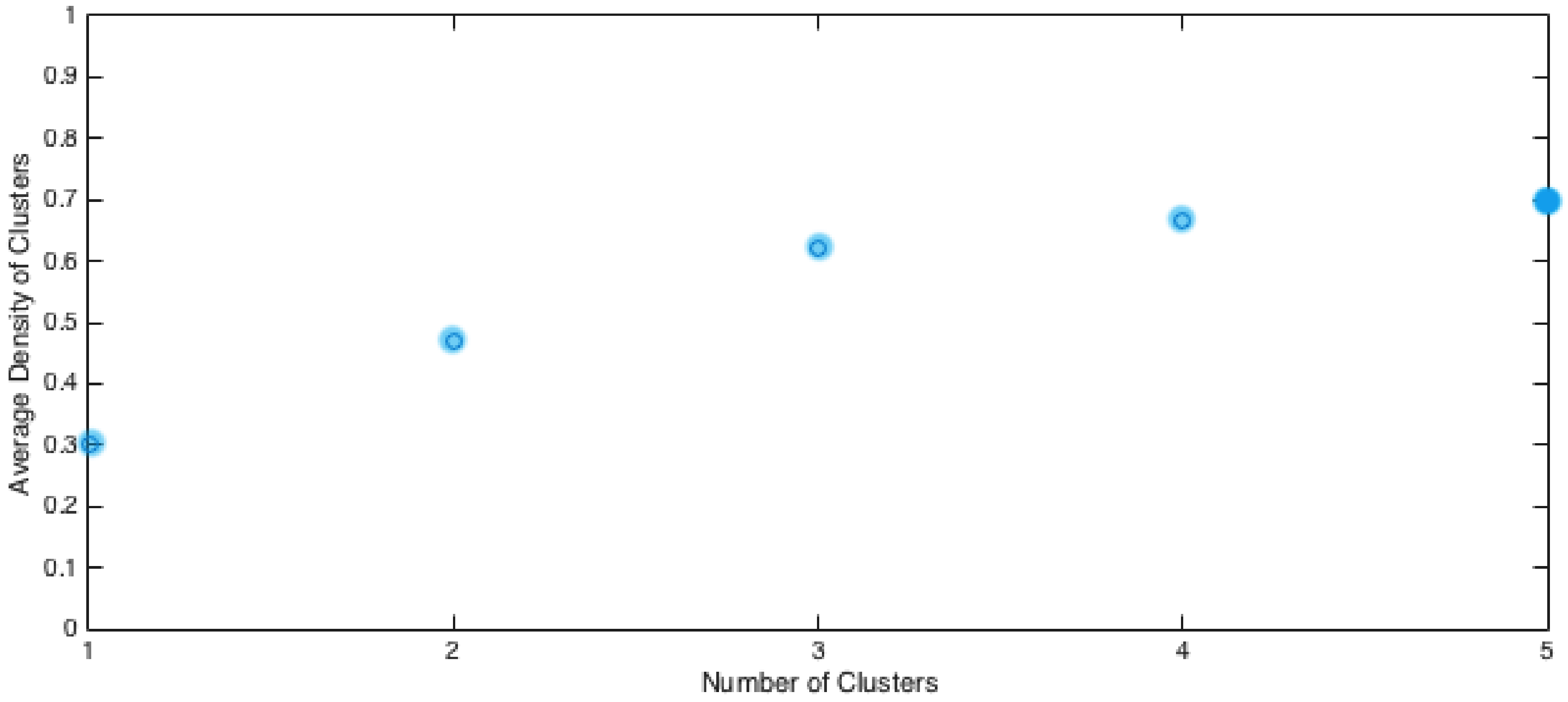
Except, we have greatly reduced the dimension of the problem!

Use the maximal cliques to combine clusters

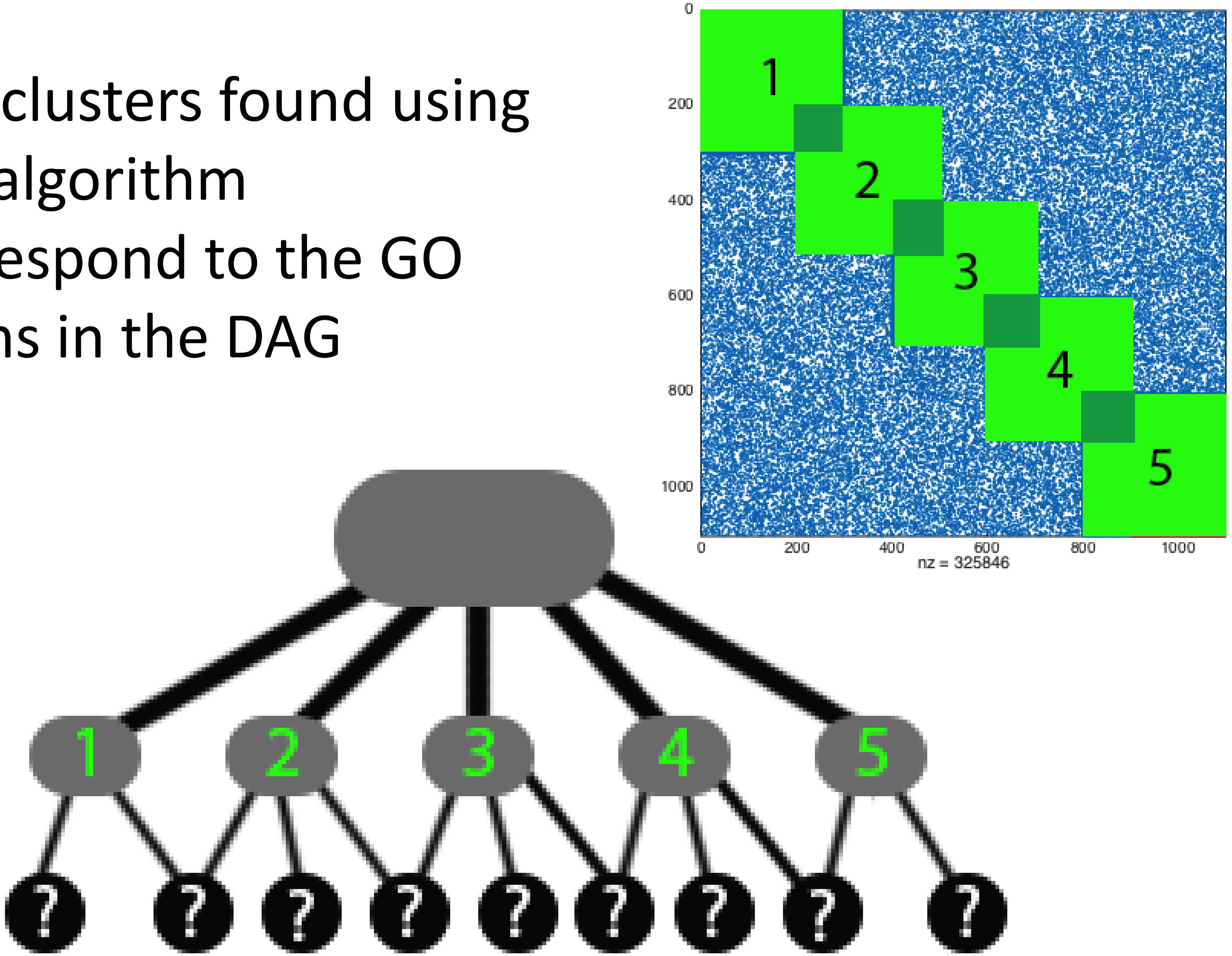




Average density of clusters vs. number of clusters (k = 1,2,…,10)



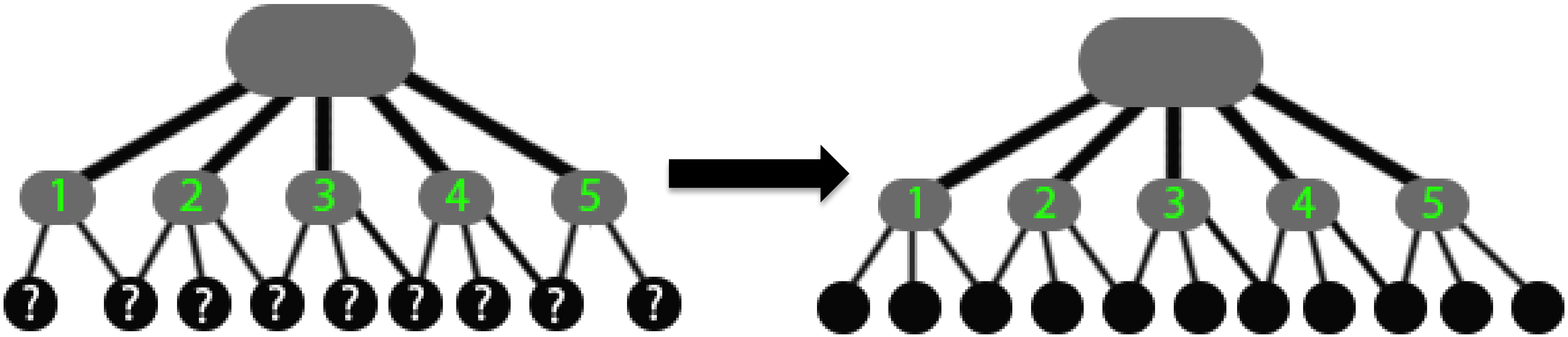
The clusters found using the algorithm correspond to the GO terms in the DAG



Genes: 1-200 200-300 300-400 400-500 500-600 600-700 700-800 800-900 900-1100

Next Steps

Applying this algorithm successively to a real gene similarity matrix to infer the entire DAG



Acknowledgements

I would like to thank my mentor, Soheil Feizi, for all his help!

Also, thank you PRIMES for this great experience!