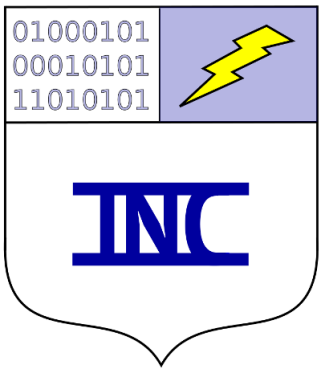
Near Cliques for Biology

Brent Kirkpatrick *

February 27, 2024

Intrepid Net Computing



www.bowtiecomputing.com

*bbkirk@bowtiecomputing.com

Document Revision History

June 30, 2022 Draft with abstract.

July 1, 2022 Draft with revisions.

February 27, 2024 Published on www.bowtiecomputing.com.

Abstract

Clique finding is one of the oldest problems in graph theory. Here we discuss cliques that are missing edges. These are called near cliques.

A clique in a biological network is a group of vertices that form a complex subgraph. A **near clique** is a group of vertices where the paths through the vertices are from a subset of permutations where the identity is included. For example a clique is also a near clique where every permutation is included. As we know from algebra, every subset of permutations is a group because every subset has an inverse.

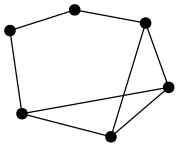


Figure 1: **Example Near Clique.** This near clique is a near clique because each permutation of vertices that is in the near clique is a cycle.

For example this graph is a near clique. The identity is the path that connects all the vertices in the cycle. The identity can always be drawn planar and convex as cords of the graph.

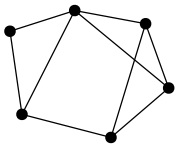


Figure 2: **Example Asymmetric Near Clique.** This near clique is asymmetric for biology.

A near clique has biological relevance. In protein-protein interaction networks, a near clique might be an active complex or might be a pathway. In RNA expression data a near clique might be splicing for a gene, or might be genes for the same pathway.

when the permutations are few to write, we can check each permutation

when there is a basis–it is a group–the permutations are few to write [1]

For biology, we want asymetries. For example, this near clique is not symmetric.

So, this graph is not rotationally or bilatterally symmetric. These graphs are more prominent in biology. In biology, these graphs are used for phylogenies, protein-protein interaction networks, chemical pathways, and gene networks.

The discussion of graph structures includes algorithms for finding near structures. For example, the perfect phylogeny is discovered as a near perfect phylogeny. For example, trees can be discussed as near trees.

Finding near cliques in a graph is hard. This is a vague statement due to the recent discoveries in complexity theory [2]. Clique finding was proven to be NP-complete and proven to be found in polynomial time. We can find all the cliques in a graph. It would appear that finding near cliques is hard.

A near clique can be built as a composit from cliques, like building blocks. But there is no proof of a polynomial length list of near cliques.

A near clique can also be built as a collection of simple cycles. But in this case the cycles have to be few in the graph.

One more criteria for near cliques that are biologically relevent is for each vertex in the near clique the number of edges incident to near clique neighbors must be greater than the number of edges into the near clique.

The algorithm presented here is inspired by clique finding. We take a graph with near cliques and randomly add edges. It is provable that every connected component in the original graph remains a connected component in the new graph. Then we look for cliques [3].

In this paper, we discover near cliques. These are approximate cliques, or partial cliques. To obtain a clique from a near clique, we add the missing edges.

Approach

This approach is called random transitive closure. The original graph is assumed to contain near cliques. We randomly add edges to the graph. Then we search for cliques in the graph. The cliques in the new graph will be near cliques in the original graph.

A **near clique** is defined as a set of vertices for which every permissible permutation of the vertices is a cycle. A permissible permutation is a path in the graph. An **organizing clique** is a near clique in a graph that no vertex can be added to the near clique with at most the number edges as the number of vertices in the near clique. When we add a vertex to an organizing clique, we add a circuit. This is to say that an organizing clique is a clique in a graph that has more structure than the edges leaving the clique.

Lemma 1. *An organizing clique less than doubles in size when adding one vertex.*

The proof is left as an exercise. The remark is that we do not build organizing cliques. Rather we focus on finding them in a larger graph. The goal is clique finding.

Our approach is to add random egdes to the graph. These random edges are between pairs of vertices that are already connected. We randomly select two edges of vertices that are already connected, and we add the third edge, the transitive closure.

Lemma 2. *The connected components are the same when transitive closure edges are added.*

Proof. Pick a vertex v. Follow two edges. Add the transitive closure selected from the two edges. These vertices are in the connected component. □

In this paper, we discuss a random algorithm for clique finding that is older than time. The contribution here is to analyze and bound the random algorithm in terms of symmetries and cycles.

Conclusions

We define an object of importance for biology, a near clique. A near clique models the reactome. These do not have to be symmetric. These do not have to be complete cliques. The algorithm here is simple, random, and finds connected components that are near cliques.

References

[1] Rudolf Mathon. A note on the graph isomorphism counting problem. *Inf. Process. Lett.*, 8(3):131–132, 1979.

[2] B. Kirkpatrick. Cycles in graphs. *TIPS*, 2022.

[3] B. Kirkpatrick. Cliques in graphs. *TIPS*, 2022.