Dear Future Mathematicians

Graph theory is fun, but working with simplicial complexes is like multi-dimensional graph theory. Instead of a single metric for neighbors and distance, one must carefully define various kinds of neighbors and distances depending on the dimensions involved. However, this added complexity allows simplicial complexes to model situations that traditional graphs cannot. For example, my research focused on authorship structures . In a graph, one can connect two vertices (authors) with an edge if they coauthored a paper. But if A, B, C are pair-wise connected, it's hard to tell whether they coauthored *one* paper together or wrote *three* papers pairwise. A simplicial complex , is a graph that allows three or more objects to be connected by a higher-dimensional "link." In this context, the links are edges, triangles, tetrahedra, or their higher-dimensional analogs (simplices). These structures solve many problems: in the case of authorship networks, a paper can be represented as a single simplex. So if A, B, C wrote one paper together, they are connected by a triangle. But if they each coauthored three pairwise papers, we instead get three edges and no triangle.

Once one build a whole complex out of these objects based on coauthorships, one can begin to do math on it. For example, I developed tools to calculate various kinds of distances between clusters of authors. These distances depend on what kinds of connections one allow: edges and up, triangles and up, tetrahedra and up, and so on. Using edges and above is equivalent to the traditional Erdős number scheme. However, at higher dimensions, if people are still connected, one can suspect they are more substantively linked—rather than merely being connected through a well-networked person who bridges unrelated fields.

Once I had the complex, I could calculate how many holes it had in each dimension. An n-dimensional hole is defined as an $(n+1)$-dimensional object that has all of its faces present in the complex but is itself missing. For example, if we have edges $(A, B), (A, C), (B, C)$, this forms the boundary of a triangle, but without the triangle itself present, it defines a 1-dimensional hole. The hole doesn't need to be a triangle: a cycle like $(A, B), (B, C), (C, D), (D, A)$ also forms a 1-dimensional hole without forming the boundary of a single 2-simplex. Although I spent much of this building the tools to find holes in simplicial complexes. My best interpretation of what it means in the context of authorship strucutres is that a hole may represent a collaboration likely to occur in the future but which hasn't yet happened.

With both distance and hole we can now "raise the bar." One problem with the basic complex is that it doesn't distinguish between a single and repeated collaboration. For example, if A, B, C coauthor a paper, and then A, B write another paper together, this second paper is not visible in the complex. To fix this, we can introduce a threshold: at level 2, a simplex is only included if that subset of vertices appears in at least two papers. In this way, A, B would be included at level 2. If later A, B, D coauthor a paper, then A, B would still be detected at level 3. This provides a measure of robustness —connections at higher levels are less likely to be incidental and more likely to reflect meaningful collaboration.

Future Directions This Project Could Take

- Apply the tools I've created
  Find new and exciting datasets and analyze them using the tools developed this summer.
- Explore authorship networks
  I spent so much time building the tools that I had little time left to fully explore my dataset.
- Optimize the existing tools
  While the code functions, it is not always optimal. Several inefficiencies are marked in the comments.
- Build new tools
  There is a lot to explore with Simplicial complexes, and many metrics that can be developed. Read the math, then apply ones understanding to expand the toolbox.

Hope this is helpful,
Sam Reiter (Class of 2027)
August 7, 2025

P.S. *Perseus* is an existing tool that implements many of these functions far more efficiently, though it focuses only on identifying holes and cannot compute distances. If one are only interested in homology (i.e., finding holes), I recommend using *Perseus*, or at least studying it for inspiration to improve my tools.