Applications of Graph Contraction Algorithms

Aaron Schankler
Motivation: Tracing Heroin Distribution

“$V = \{v_1, \ldots, v_{18}\}$ represents the list of the 18 detected cutting agents and corresponds to the vertex of the graph; $E = \{v_i; v_j\}$ represents the edges between the vertices, which appear if two cutting agents are found together”

What we need

- The co-occurrence of cutting agents “shows a promising potential in an intelligence perspective for analysing the local distribution process of heroin” (Terrettaz-Zufferey et. all)
- To do this, we need to find common ancestors of different drug samples
- We also need to determine which related samples are more closely related (diverged lower in the supply chain)
Contractions

**Definition: Edge contraction**
An edge is deleted and its two incident vertices are merged into a new vertex $v$. $V$ is adjacent to every vertex that was adjacent to either of the two original vertices.
Definition: Vertex Expansion

A vertex $w$ is split into two connected vertices, $u$ and $v$. All vertices adjacent to $w$ must be adjacent to $u$ or adjacent to $v$. 
The Problem

Given two graphs:

- Can we determine if they share a common contraction?
- If they do share a common contraction, can we efficiently find it?
- Can we uniquely determine the source graph?
Next Steps

- Implement algorithms that naively find all possible expansions and contractions of a graph
- Apply these to simple graphs and look for patterns or features that persist across expansions
- Refine algorithms to look for these “features of interest”
- Redefine the common contraction problem to apply to drug profiling, perhaps using contractions as a similarity metric to relate different samples