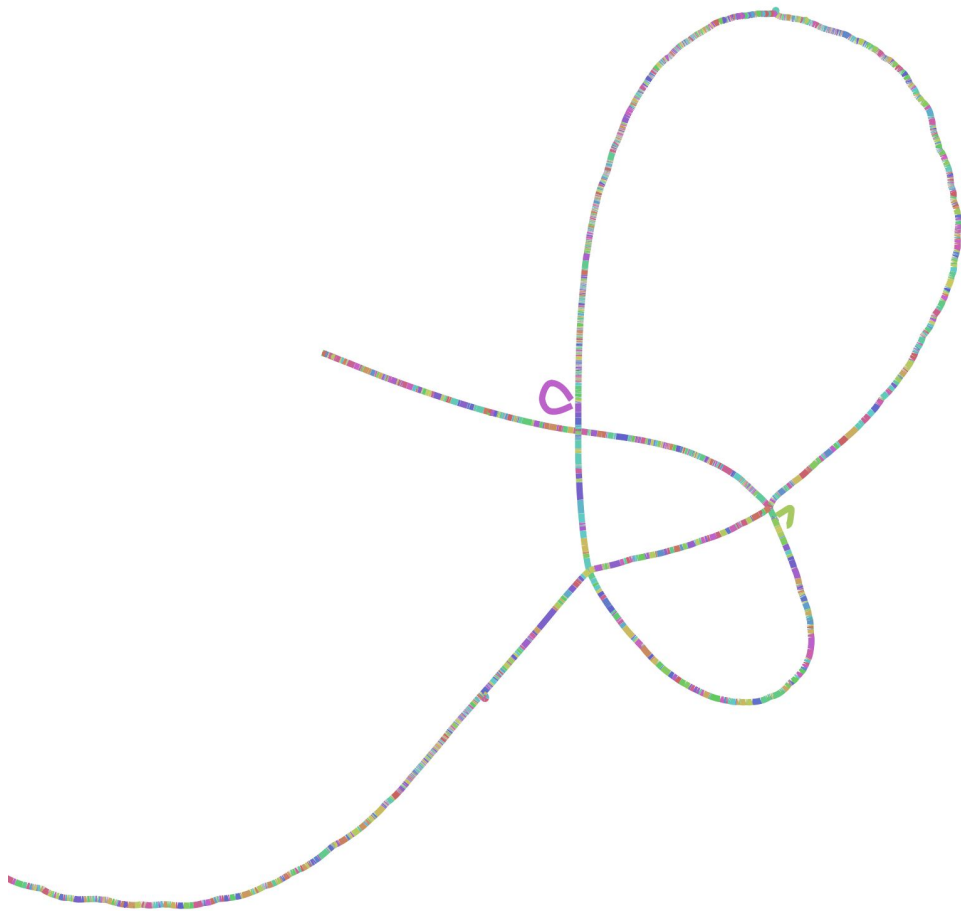


Minigraph Cactus

- `graphtype: "MC"`
- `debug_small_graphs: false`
- `minnodelen: 1.0`
- `nodeseglen: 1.0`
- `edgelen: 0.1`
- `nodelenpermb: 1000000`

The default setting for most of these variables may make the minigraph cactus graph look really bad, so we may need to do some experiments on that. (the setting above can be a start point)



Bandage Variables

- `graphtype`: Minigraph (the one we have implemented now), or Minigraph Cactus
- `debug_small_graphs`: (bool) if true, each node's drawn length is exactly the node's length in base pair
- `minnodelen`: (float) minimum node length to draw. For every node, if it's drawn length is smaller than **minnodelen**, we use **minnodelen** as their drawn length. (default: 5.0, min: 1.0, max: 100.0)
- `nodeseglen`: (float) max length of every OGDF nodes. (eg. if node length is larger than **nodeseglen**, the node will be splitted into several OGDF nodes). (default: 20.0, min: 1.0, max: 1000.0)
- `edgelen`: (float) edge length between nodes (default: 5.0, min: 0.1, max: 100.0)
- `nodelenpermb`: (float) node length per megabase. A scaling factor for the node length drawn. If **debug_small_graph** is false, we use **nodelenpermb** to calculator the node length that will be drawn. (default: 1000.0, min: 0, max: 1000000.0)
 - $\text{drawnNodeLength} = \text{NODELENPERMB} * \text{Node Length in bp} / 1000000$
 - Setting **nodelenpermb** to 1000000 has the exact same effect as setting **Debug_small_graphs** to true (it's better to use **nodelenpermb**, and I may delete **debug_small_graphs** eventually)

