# Persistent Homology Guided Force-Directed Graph Layouts

Ashley Suh, Mustafa Hajij, Bei Wang, Carlos Scheidegger & Paul Rosen
Presented by Javier Castillo-Arnemann

# Background

- Graphs/networks are commonly used to encode relationships among entities, used for data exploration.
- Used to model social networks, digital networks, biological interactions, etc.
- Their abstractness makes them difficult to analyze.
- A good graph vis should present structure quickly and clearly, and support further investigation of the data.

# Background

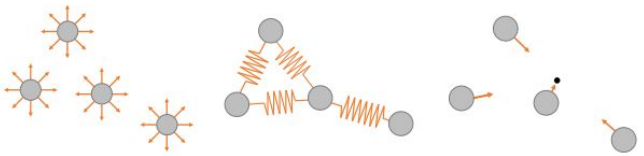
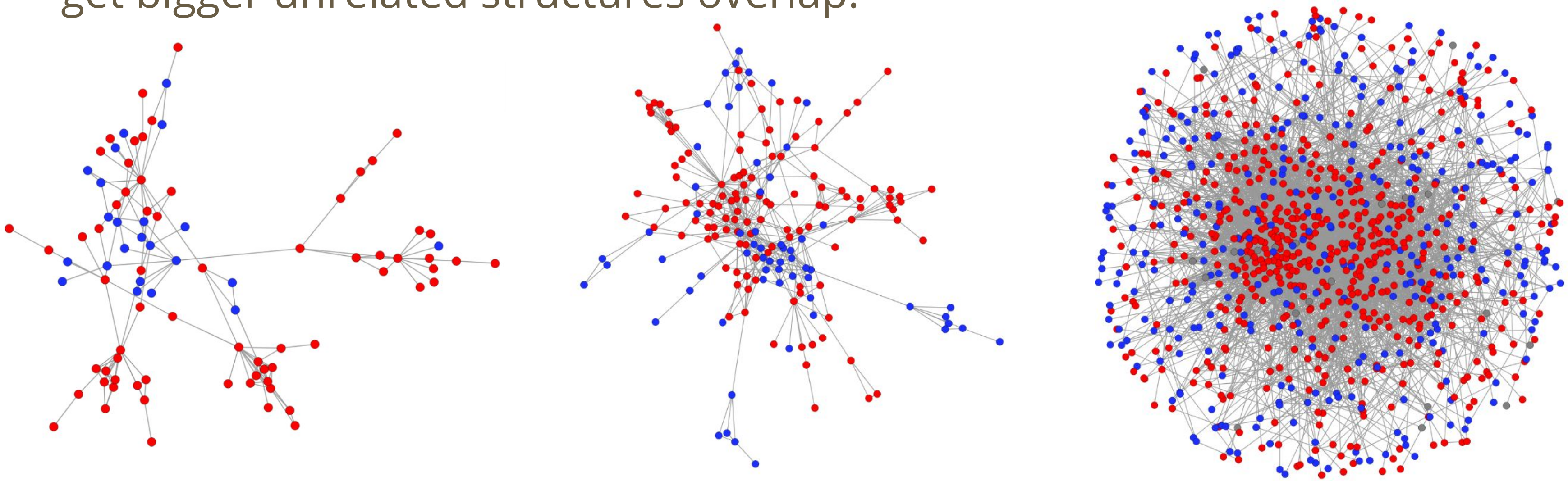


Fig. 7

- Force-directed/spring-mass layouts are the most popular graph layout for interactive applications.
- Good at showing topological structures in sparse graphs, but as graphs get bigger unrelated structures overlap.



# Approach

- **What?** Improve force-directed graph layout visualization.
- **Why?** Identify, compare and separate important topological structures.
- **How?** Use persistence homology features to enable interactive manipulation of the graph layout, visually separating PH features.

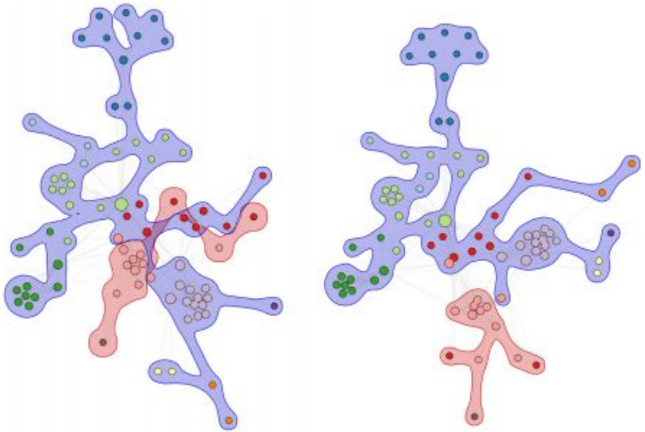


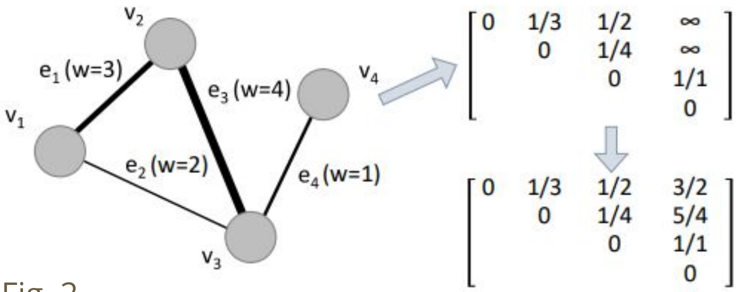
Fig. 7

# Approach

1. Embed a **weighted undirected graph in metric space** by inducing a distance between all nodes.
2. Extract **Persistence Homology (PH) features** of the metric space structure and sort them based on persistence.
3. PH features can be used either to **contract the nodes that created the feature**, or to **repulse the graph into two subsets**, depending on user input.

# Approach

1. Embed an undirected graph in metric space by inducing a distance between all nodes.
    - The inverse of the edge weight is the distance, and the shortest path between two nodes is the distance between them.
    - The rest of the algorithm operates in this metric space.



$$\begin{bmatrix} 0 & 1/3 & 1/2 & \infty \\ & 0 & 1/4 & \infty \\ & & 0 & 1/1 \\ & & & 0 \end{bmatrix}$$

$$\Downarrow$$

$$\begin{bmatrix} 0 & 1/3 & 1/2 & 3/2 \\ & 0 & 1/4 & 5/4 \\ & & 0 & 1/1 \\ & & & 0 \end{bmatrix}$$

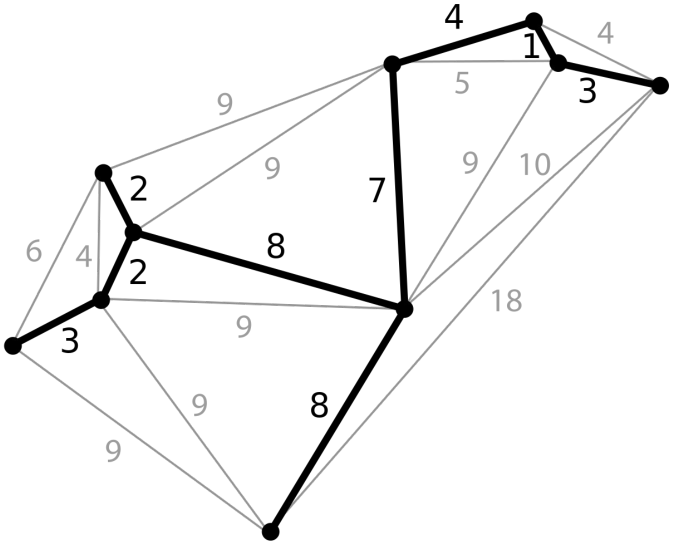
Fig. 2

If the graph is unweighted, the Jaccard index, edge centrality or other measures can be computed and used as weights.

# Approach

2. Extract **Persistence Homology (PH) features** of the metric space structure and sort them based on persistence.

- The persistence homology of the graph can be calculated by finding its Minimum Spanning Tree (MST).
- Subset of edges that connects every node without cycles and minimizing the edge weights.

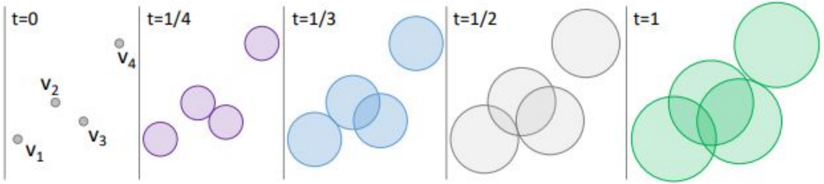


Wikipedia

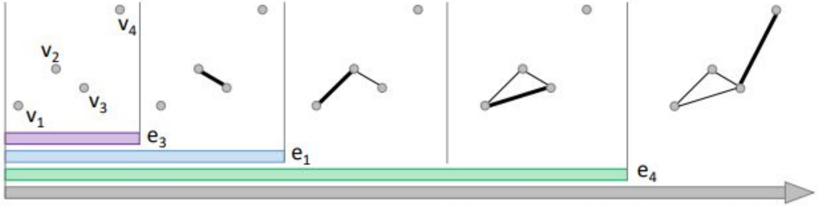
# Approach

2. Extract **Persistence Homology (PH) features** of the metric space structure and sort them based on persistence.

- Every node corresponds to a point in the metric space.
- Consider the set of balls centered at every point with a diameter *t* (metric space resolution).
- Every time two balls (components) merge as *t* grows, one of the components disappear and its “time of death” *t* is recorded.
- The lifetime of a component is its **persistence** (0, 1 / *w*).



(b) Conceptual Construction of PH Features Using Growing Balls



(c) Computational Construction of PH Features Using 1-Simplicies

Fig. 2

# Approach

## 2. Extract **Persistence Homology (PH) features** of the metric space structure and sort them based on persistence.

Every PH feature is associated with:

- **Persistence** measure 1/*w*
- **Cause of death** *u* and *v*
- **Subsets of nodes** $V_u$ and $V_v$, the sets of connected nodes after removing the feature from the MST.
- **Subset ratio** $|V_u| : |V_v|$, measures the centrality of the feature within the MST.

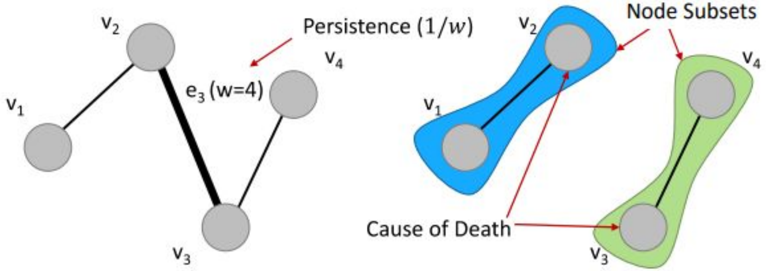


Fig. 3

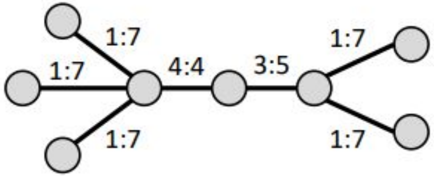


Fig. 4

# Approach

3. PH features can be used either to **contract the nodes that created the feature**, or to **repulse the graph into two subsets**, depending on user input.

- Video Demo

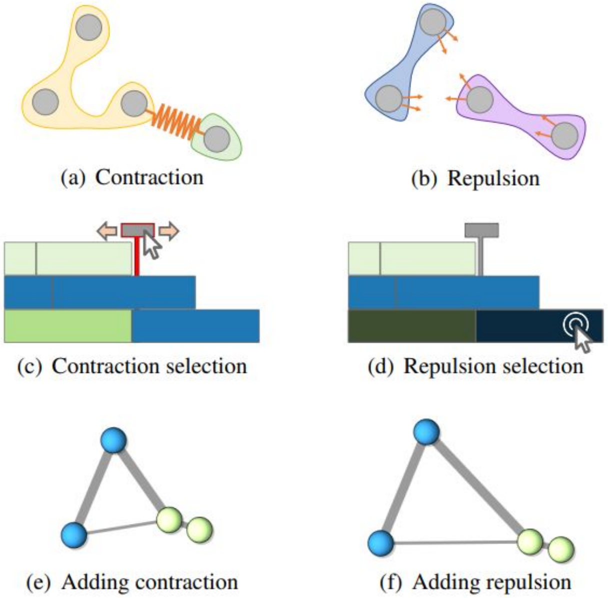

(a) Contraction
(b) Repulsion
(c) Contraction selection
(d) Repulsion selection
(e) Adding contraction
(f) Adding repulsion

Fig. 6

# Comparison

- Comparing the PH of the traditional force-directed **source** layout and the user-selected **target** layout.
- PH calculated on Euclidean distance between the nodes, not topological distance.

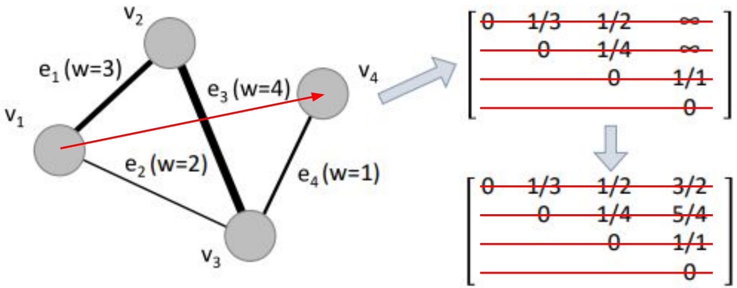
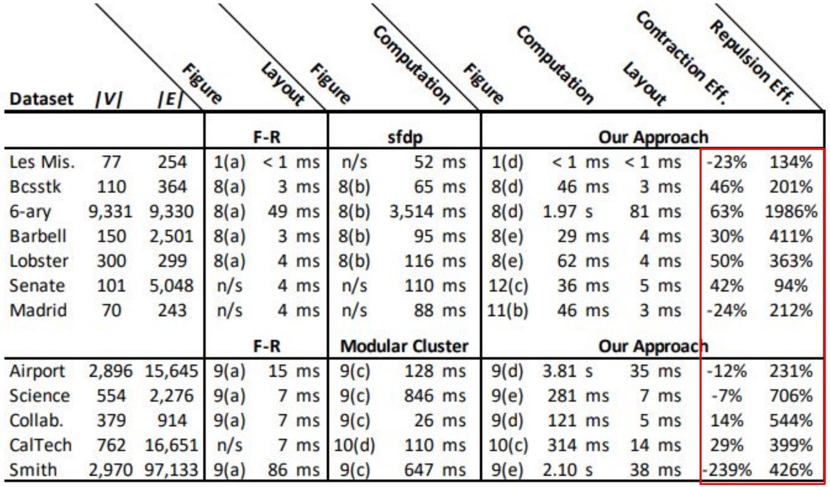
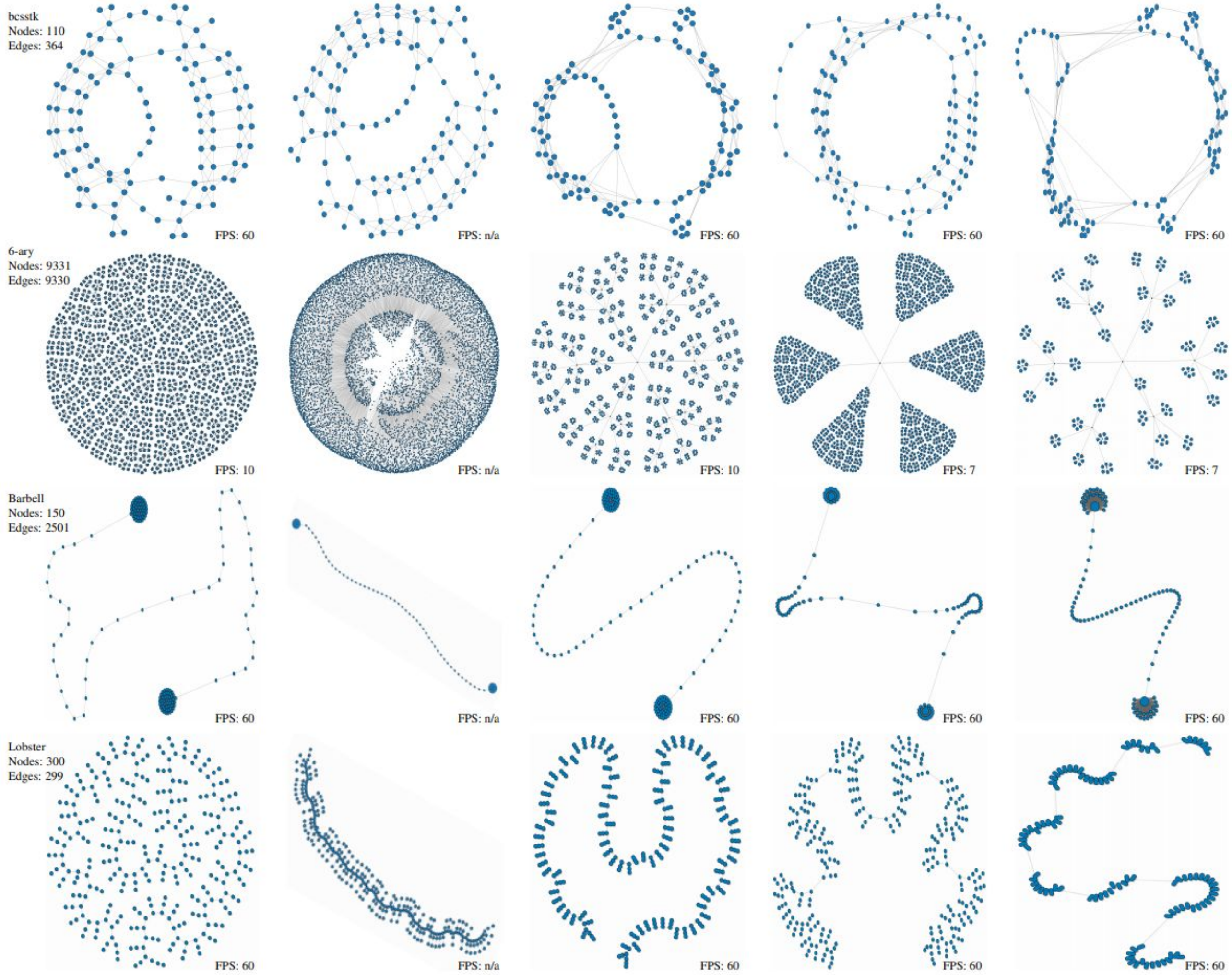


Fig. 2



| Dataset | \|V\| | \|E\| | Figure | Layout | Figure | Computation | Figure | Computation | Layout | Contraction Eff. | Repulsion Eff. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | F-R | | sfdp | | Our Approach | | | | |
| Les Mis. | 77 | 254 | 1(a) | < 1 ms | n/s | 52 ms | 1(d) | < 1 ms | < 1 ms | -23% | 134% |
| Bcsstk | 110 | 364 | 8(a) | 3 ms | 8(b) | 65 ms | 8(d) | 46 ms | 3 ms | 46% | 201% |
| 6-ary | 9,331 | 9,330 | 8(a) | 49 ms | 8(b) | 3,514 ms | 8(d) | 1.97 s | 81 ms | 63% | 1986% |
| Barbell | 150 | 2,501 | 8(a) | 3 ms | 8(b) | 95 ms | 8(e) | 29 ms | 4 ms | 30% | 411% |
| Lobster | 300 | 299 | 8(a) | 4 ms | 8(b) | 116 ms | 8(e) | 62 ms | 4 ms | 50% | 363% |
| Senate | 101 | 5,048 | n/s | 4 ms | n/s | 110 ms | 12(c) | 36 ms | 5 ms | 42% | 94% |
| Madrid | 70 | 243 | n/s | 4 ms | n/s | 88 ms | 11(b) | 46 ms | 3 ms | -24% | 212% |
| | | | F-R | | Modular Cluster | | Our Approach | | | | |
| Airport | 2,896 | 15,645 | 9(a) | 15 ms | 9(c) | 128 ms | 9(d) | 3.81 s | 35 ms | -12% | 231% |
| Science | 554 | 2,276 | 9(a) | 7 ms | 9(c) | 846 ms | 9(e) | 281 ms | 7 ms | -7% | 706% |
| Collab. | 379 | 914 | 9(a) | 7 ms | 9(c) | 26 ms | 9(d) | 121 ms | 5 ms | 14% | 544% |
| CalTech | 762 | 16,651 | n/s | 7 ms | 10(d) | 110 ms | 10(c) | 314 ms | 14 ms | 29% | 399% |
| Smith | 2,970 | 97,133 | 9(a) | 86 ms | 9(c) | 647 ms | 9(e) | 2.10 s | 38 ms | -239% | 426% |

Table 1



(a) F-R Layout (b) Neato Layout (c) Ours: Contracted (d) Ours: Repulsed (e) Ours: Combination

Fig. 8

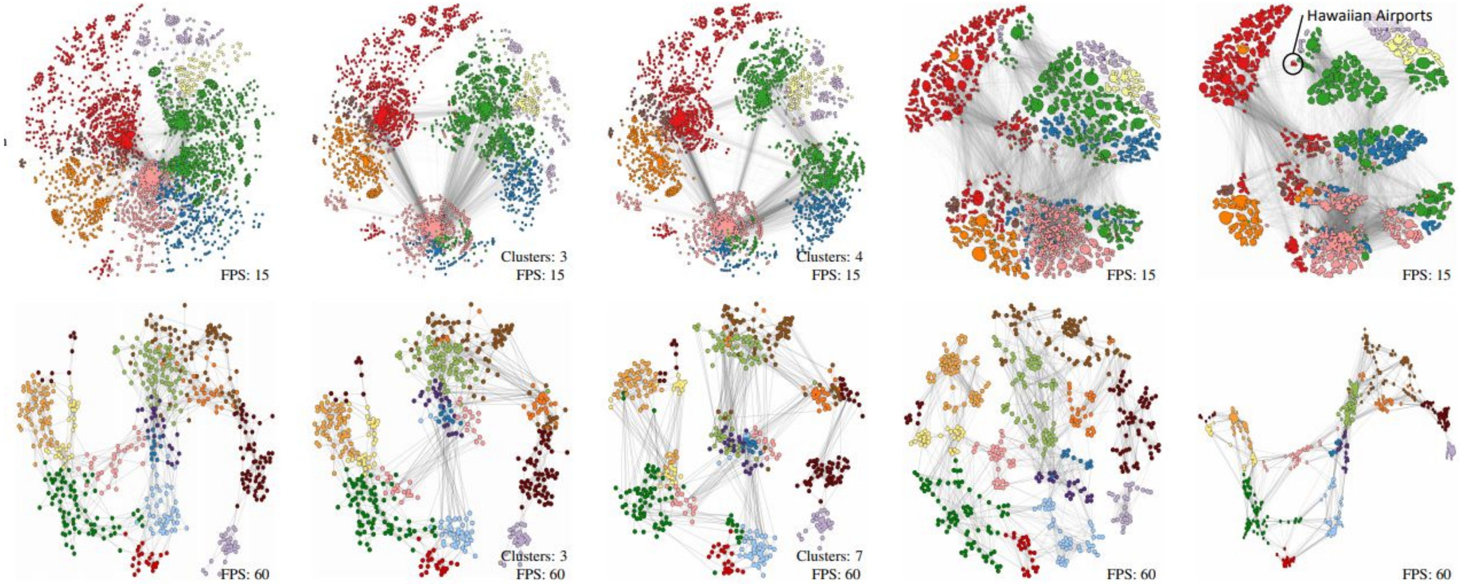

FPS: 15
Clusters: 3
FPS: 15
Clusters: 4
FPS: 15
FPS: 15
Hawaiian Airports
FPS: 15
FPS: 60
Clusters: 3
FPS: 60
Clusters: 7
FPS: 60
FPS: 60
FPS: 60

Fig. 9

# Discussion

Strengths:

- Real-time selection of the PH features instead of choosing number of clusters beforehand.
- Combination of repulsive and attractive forces allows better separation than either force alone.
- Can be extended to other force-directed layouts.
- Performance comparable to traditional force-directed layouts.

Weaknesses:

- Shallow explanation of the modifications to the force-directed algorithm.
- Performance with unweighted graphs. Extra computations for weights?
- Can this approach be used with other community detection/network clustering methods instead of PH features?
- What happens when the number of PH features is too high to make the barcode impractical?