Supplementary Material: AOC; Assembling Overlapping Clusters

Akhil Jakatdar[1], Baqiao Liu[1], Tandy Warnow[*1], and George Chacko[†1,2]

[1]Department of Computer Science, University of Illinois Urbana-Champaign, Urbana, IL 61801
[2]Office of Research, Grainger College of Engineering, University of Illinois Urbana-Champaign, Urbana, IL 61801

September 27, 2022

[*]warnow@illinois.edu
[†]chackoge@illinois.edu

1 Line Graph Proof

An analytical solution to calculate the size of a graph after line graph transformation of an input graph. The example used is the Curated Exosome Network (CEN), which has 13,989,436 vertices and 92,051,051 edges.

1.1 Derivation of Line Graph Nodes and Edges

In transforming any undirected graph $G(V, E)$ into its corresponding line graph $L(G(V', E'))$, the number of vertices $|V'| = |E|$ as all edges in the original graph are transformed into vertices in its corresponding line graph.

We can also see that every edge $e' \in E'$ can be described as a set of 2 edges $\{e_1, e_2\}$ where $e_1, e_2 \in E$ where both edges are adjacent to some common vertex $v \in V$. Thus in order to find the total number of edges $e' \in E'$ for some fixed vertex $v \in V$, we formulate the number of edges as $\binom{d_v}{2}$ where d_v represents the degree of v. Thus summing over all vertices in V, we get the following:

$$\begin{aligned} |E'| &= \sum_{v}^{V} \binom{d_v}{2} \\ &= \sum_{v}^{V} \frac{d_v(d_v - 1)}{2} \\ &= \sum_{v}^{V} \frac{{d_v}^2}{2} - \sum_{v}^{V} \frac{d_v}{2} \\ &= \frac{1}{2} \sum_{v}^{V} {d_v}^2 - \sum_{v}^{V} \frac{d_v}{2} \end{aligned} \tag{1}$$

The expression $\sum_{v}^{V} \frac{d_v}{2}$ can be simplified to $|E|$ as the sum of the degrees of all vertices in G divided by 2 is equivalent to the total number of edges in G. Thus our closed form equation for $|E'|$ is as follows:

$$|E'| = \frac{1}{2} \sum_{v}^{V} {d_v}^2 - |E| \tag{2}$$

We can now use these two closed form solutions to identify the number of edges and vertices in the corresponding line graph to the Exosome Citation Network. Given that this network $N(V_n, E_n)$ has $|V_n| = 13989436$ vertices and $|E_n| = 92051051$, we

can solve for $|V_n'| = 92051051$ and plug in the degree distribution of the network into our closed form solution for the number of edges to find the following:

$$\begin{aligned} |E_n'| &= \frac{1}{2}\sum_{v}^{V_n} {d_v}^2 - |E_n| \\ &= \frac{1}{2}(321041864344) - 92051051 \quad = 160428881121 \end{aligned} \tag{3}$$

Thus, the corresponding line graph $L(G_n(V_n', E_n'))$ to our Exosome Citation Network will have $|V_n'| = 92,051,051$ and $|E_n'| = 160,428,881,121$.

2 Allowing Overlapping Clusters

This program runs the step 5 of the Allowing Overlapping Clusters method and saves it to your local directory. To run this program, run the following code.

In order to run the `overlapping_kmp_pipeline.py` program, the following code must be run in the following format.

```
$ python3 overlapping_kmp_pipeline.py \
--network-file [path_to_tsv_network file] \
--clustering [path_to_input_kmp-valid_clustering_file.csv] \
--output-path [path_to_output_overlapping_clustering] \
--min-k-core [min_k_of_input_clustering] \
--rank-type [percent, percentile] \
--rank-val [top_n_percent_of_candidate_nodes_to
   consider_adding] \ #opposite if percentile
--inclusion-criterion [k, mcd] \
--candidate-criterion [total_degree,indegree,random,seed] \
--candidate-file [path_to_custom_candidate_list] \
--experiment-name [name_of_experiment_being_run] \
--experiment-num [experiment_number_being_run] \
--config [bool, bool, bool, bool]
# (run overlapping?, display cluster stats?, include marker
   node analysis?, save outputs?)
```

Shown below are the flags and what purpose each flag accomplishes.

- **network-file** - str - file path to network tsv file of the format `node_id[\space] node_id`

- **clustering-file** - str - file path to clustering tsv file of the format `cluster_id[\space]node_id`
- **min-k-core** - int - integer defining the min-k-core by which to add candidate nodes if selecting k as inclusion criterion. It is still mandatory even if not using k, so a dummy value must be filled in
- **rank-type** - [percent, percentile] - choose whether to use percent or percentile when calculating list of candidate nodes to generate
- **rank-val** - float - the percent or percentile value used to generate the list of candidate nodes
- **inclusion-criterion** - [k, mcd] - utilize k or mcd of cluster as the inclusion criteria for adding a candidate node to a cluster
- **candidate-criterion** - [total_degree, indegree, random, seed] - criterion by which the rank type and rank val sort the list of nodes in the network in deciding the candidate nodes
- **candidate-file** - str - only unrequired field that, if included, will override the other candidate selection values and simply use the nodes specified in the candidate file as candidate nodes. They must be in the format `node_id[\n]node_id`
- **experiment-name** - str - string to specify the name of the experiment in order to name the statistic csv files that are ouputted with the final clustering file
- **experiment-num** - int - must correspond to a directory in the environment where the program will be run with the format `experiment_{experiment-num}`
- **config** - [bool, bool, bool, bool] - list of four boolean values that correspond to the following criteria. 1. run the overlapping step of the kmp pipeline. 2. Display the cluster statistics to the console. 3. Include an analysis of marker node coverage. 4. Save output clustering files and cluster statistics.

cluster_id	ikc	aoc_m	aoc_k	ikc_perc	aoc_m_perc	aoc_k_perc
1	24	49	211	2	5	21
2	39	44	140	4	4	14
3	167	434	618	16	43	61
4	416	921	932	41	90	91
5	1.	5	14	0	0	1
6	9.	9	58	1	1	6
8	0	0	4	0	0	0
9	2	3	12	0	0	1
10	0	0	1	0	0	0
11	0	0	4	0	0	0
12	3	6	40	0	1	4
13	1	3	9	0	0	1
14	0	0	1	0	0	0
15	0	2	2	0	0	0
16	11	20	37	1	2	4
18	0	0	18	0	0	2
19	0	0	3	0	0	0
20	18	76	152	2	7	15
21	1	1	3	0	0	0
22	0	0	2	0	0	0
23	0	0	2	0	0	0
24	6	15	52	1	1	5
25	310	310	710	30	30	70
30	1	1	1	0	0	0
34	0	0	3	0	0	0
37	11	12	28	1	1	3
50	0	0	1	0	0	0
53	0	0	1	0	0	0
66	1	1	3	0	0	0
88	0	2	2	0	0	0
116	0	2	7	0	0	1

Table 1: Percentages of 1021 marker nodes found in AOC clusters. Markers in clusters are shown both as actual counts (cols 2-4) and as percentages of 1021 (cols 5-7); Clusters 3, 4, and 25 are in the 90th percentile of marker node concentration in AOC_k clusters. Data are shown for clusters with a non-zero value in the aoc_k column. Please note that the cluster numbering is arbitrary on account of the workflow that matches IKC to AOC_m/k clusters and does not reflect the order in which IKC cores are extracted.

	tag	nodes			edges			mcd		
		min	median	max	min	median	max	min	median	max
1	ikc	14	79.00	214,877	78	993.50	4,159,555	10	16.00	53
2	aoc_m	14	80.00	274,355	78	1090.00	6,550,067	10	16.00	53
3	aoc_k	14	93.50	291,154	78	1272.50	7,681,827	10	11.00	18

Table 2: Summary Statistics for Clusters generated by IKC, AOC_m, and AOC_k

	ikc_cid	node_count	edge_count	mcd	tag	aoc_cid
7	2	1,869	83,247	49	ikc	25
1	2	1,871	83,411	49	aoc_m	25
4	2	9,241	347,263	10	aoc_k	25
8	30	170,413	4,159,555	26	ikc	3
2	30	190,427	6,369,455	26	aoc_m	3
5	30	242,857	7,681,827	10	aoc_k	3
9	72	214,877	2,724,699	14	ikc	4
3	72	274,355	6,550,067	14	aoc_m	4
6	72	291,154	7,075,034	10	aoc_k	4

Table 3: Cluster statistics for marker-rich clusters (Fig 6 in Jakatdar et al.)The mapping of AOC cluster ids to the original clusters generated by IKC is shown in the first and last columns. Whereas aoc_m preserves mcd, aoc_k results in a decrease in the mcd of IKC clusters