Supplementary Material for SASCA: Scalable Agent-based Simulator for Citation Analysis

Minhyuk Park[1], João AC Lamy[2], Esther CC Rodrigues[2], Felipe M Ferreira[2], Tandy Warnow[1], and George Chacko[1]

[1] Siebel School of Computing and Data Science, University of Illinois Urbana-Champaign, Urbana, IL 61801
George Chacko chackoge@illinois.edu,
https://siebelschool.illinois.edu/
[2] Insper Instituto de Ensino e Pesquisa, São Paulo,
Brazil

1 Software Commands and Versions

PyABM:

- Code location: https://github.com/illinois-or-research-analytics/abm_citations/tree/v3-dev
- Commit: ae0aa1eeb09eefd199f58fd8afd4a98ea2333f26

```
export NUMBA_NUM_THREADS=<num threads>
python abm_citations/generators/batch/batch_run_gilbertian_model.py --config ./base_config
```

Listing 1.1. PyABM command

```
[Environment]
cluster_data=<csv file with first two columns listing node id and publication year>
edge_list=<csv file with (source,target) formatted edgelist>
gamma=3
c=1
growth_rate=<growth rate e.g., 0.03 for 3%>
num_years=<number of cycles e.g., 30>
out_folder=<output folder>
recency_table=<two column csv with (n,k) showing the number of citations (k) citing a publication from n years ago, derived from a real-world network>
reference_count_table=<two column csv with the second column being the possible out-degrees for each agent, derived from a real-world network>
same_year=<proportion of nodes citing the same year e.g., 0.12 for 12%>

[Agent]
no_superstars=true
preferential_weight=<"random"or a floating point number indicating the weight>
recency_weight=<"random"or a floating point number indicating the weight>
fitness_weight=<"random"or a floating point number indicating the weight>
alpha=<"random" or a floating point number>
```

Listing 1.2. base_config file example

SASCA:

- Code location: https://github.com/illinois-or-research-analytics/SASCA
- Commit: 655368ebb30ece7d528be7f9c1364721739a9d01

```
abm --edgelist <INPUT_EDGELIST> --nodelist <INPUT_NODELIST> --out-degree-bag <
    OUTDEGREE_BAG> --recency-probabilities <RECENCY_PROBABILITIES> --growth-rate <
    GROWTH_RATE> --fully-random-citations <FULLY_RANDOM_CITATIONS> --num-cycles <
    NUM_CYCLES> --same-year-proportion <SAME_YEAR_PROPORTION> --output-file <
    OUTPUT_FILE> --auxiliary-information-file <OUTPUT_AUX> --log-file <OUTPUT_LOG>
    --num-processors <NUM_THREADS> --log-level <LOG_LEVEL> --neighborhood-sample <
    Number of nodes to sample from each neighborhood. -1 for no sampling> --
    preferential-weight <-1 for random or otherwise a floating point number
    indicating the weight> --recency-weight <-1 for random or otherwise a floating
    point number indicating the weight> --fitness-weight <-1 for random or
    otherwise a floating point number indicating the weight> --alpha <-1 for random
     or otherwise a floading point number>
```

Listing 1.3. SASCA command

2 Out_degree distribution sampled from PubMed

To better approximate realistic citation patterns, we collected a random sample of PubMed articles to empirically estimate reference out-degrees. A sequence of PubMed IDs (PMIDs) was generated by identifying the range between a randomly selected article from 2020 and another from 2025. From this range, 10,000 PMIDs were randomly sampled from a total of 8,584,929 possible PMIDs and their corresponding XML records were retrieved using Biopython's Entrez API. Each XML record was parsed to count the number of references cited by the article, either all or restricted to ArticleId IdType="pubmed", which indicates links to other articles in the PubMed dataset. In retrieving articles for our random sample, we filtered out any articles where the PublicationType was not Journal Article. Records that lacked this label or did not contain a valid reference list were excluded from analysis. Records with less than 10 references or more than 249 were discarded. This procedure was repeated across five independent runs of 10,000 articles each, resulting in a total sample of 50,000 articles. The output consisted of CSV files containing each article's PMID and its associated out-degree (reference count), which were then used to construct an empirical citation distribution to use as input. The final distribution consisted of 46,031 out_degree values ranging from 10 to 249 with first quartile, median, and third quartile values of 23, 35, and 52 respectively.

3 Community Structure

To understand the impact on community structure, we used SASCA-s to simulate the growth of two different kinds of networks with different tendencies for local citations by using two extreme alphas (0.01 and 0.99). We clustered these two networks with the Leiden algorithm optimizing for CPM criterion with three different resolution values: 0.1, 0.01, and 0.001. As shown in Figure 1, at each resolution values, the clusterings of the alpha=0.99 networks resulted in lower conductance and higher modularity, indicating a stronger community structure. However, we did notice a slight decrease in the normalized connectivity values which are computed by dividing the minimum edge cut size of each cluster by the log base 10 of the cluster size. Table 1 shows the cluster sizes for each resolution value where we can see that the median cluster sizes for the alpha=0.99 setting are almost double the median cluster sizes for alpha=0.01 which may help explain the lower normalized connectivity values for alpha=0.99 since the denominator in general would be larger.

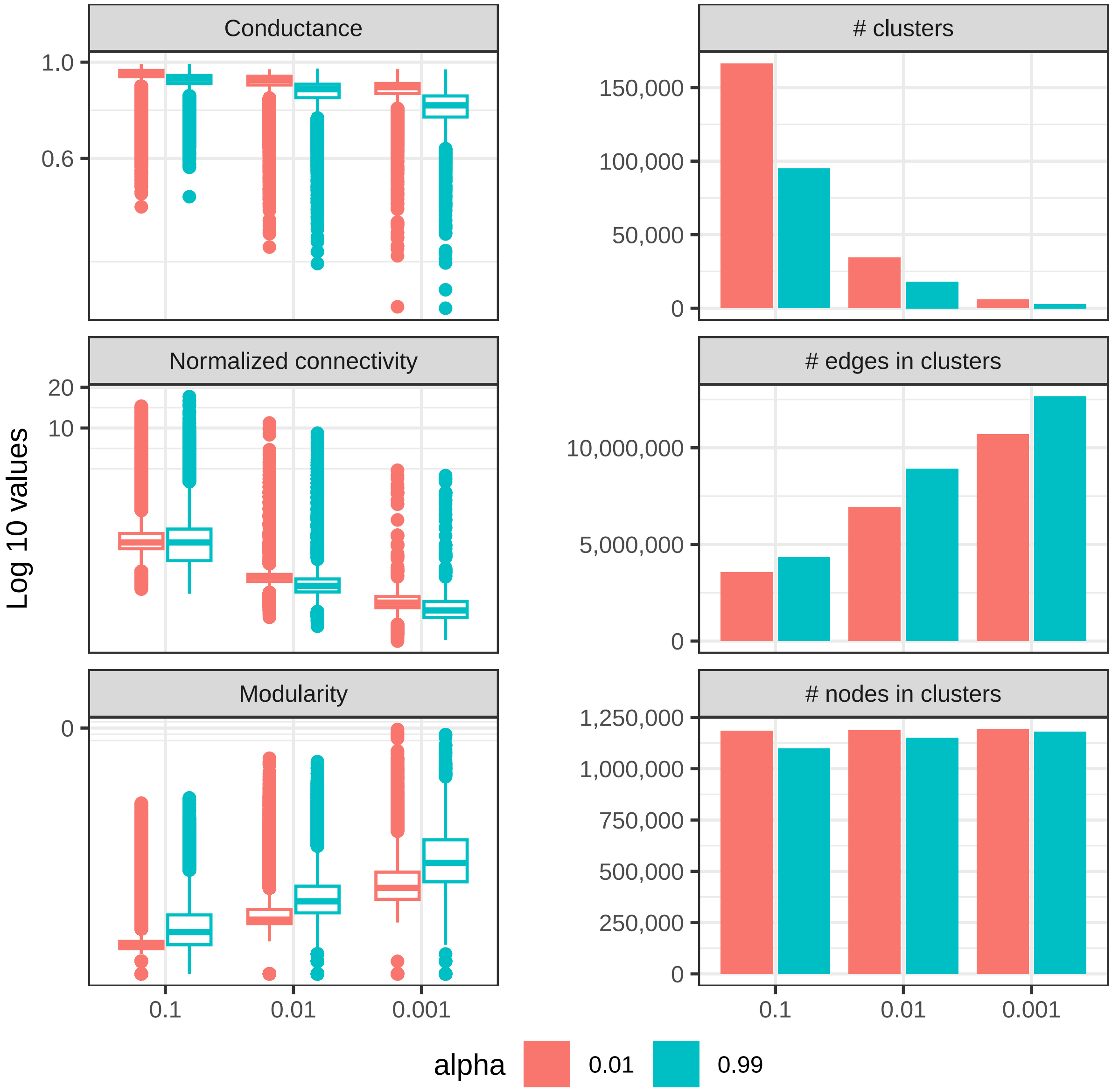

Fig. 1. Clustering SASCA-s networks. Various statistics at the cluster levels are shown for the Leiden-CPM clusterings of the SASCA-s networks.

res value	α	min	median	max	node_cov
0.1	0.01	2	5	360	0.99
0.1	0.99	2	9	435	0.92
0.01	0.01	2	19	3790	1.00
0.01	0.99	2	32	3102	0.96
0.001	0.01	2	91	20146	1.00
0.001	0.99	2	188	18395	0.99

Table 1. Clustering SASCA-s networks SASCA-s networks were generated over 30 years and at 3% growth using the *sj* as seed with α set to either 0.99 or 0.01. Each network consists of 1,193,102 nodes and was clustered with the Leiden algorithm [**?**] optimizing the Constant Potts model at three different resolution values. While node coverage (the fraction of the network in cluster of at least size 2 is consistently high, cluster size increases with α and as resolution value is decreased.