FastEnsemble: A new scalable ensemble clustering method

Presenter: Minhyuk Park

Yasamin Tabatabaee, Eleanor Wedell, Tandy Warnow

University of Illinois Urbana-Champaign

12/10/24

Community Detection a.k.a. Clustering

- Input: Simple graph without a distance matrix
- Output: Partitioning of nodes into disjoint sets

- E.g., Louvain (modularity), Leiden (modularity, CPM), Stochastic Block Models (SBM), Infomap, Markov Cluster Algorithm (MCL)

Why Ensemble Methods?

- Community detection methods often include randomness

- Ensemble methods gather reliable signal from multiple clustering outputs

Prior Literature

- FastConsensus (Tandon et al., 2019):
  - Start with multiple runs of a clustering method on an input network
  - Create consensus matrix (co-classification matrix)
  - Remove weak links
  - Perform triadic closure
  - Repeat from first step until convergence

- Ensemble Clustering for Graphs - ECG (Poulin and Théberge, 2018):
  - Start with multiple runs of the Louvain algorithm (modularity)
  - Create consensus matrix (co-classification matrix)
  - Set minimum edge weight to edges not in a 2-core in the original graph
  - Run the Louvain algorithm on this new matrix

FastEnsemble Design Goals

- Avoid iterations to improve runtime
- Generalize the ensemble step to allow for arbitrary clustering methods

FastEnsemble Algorithm Design

- FastEnsemble takes 2 parameters:
  - *np* - num partitions (clusterings)
  - *t* – threshold

- Given a network:
  - Generate *np* clusterings on the network
  - Generate a new weighted network
  - Remove edges with weight less than *t*
  - Run a clustering method on the new weighted network

- Note: **Strict consensus** is FastEnsemble with *t* equal to 1

Experiments in This Study

- We show results for Louvain, Leiden-mod, and Leiden-CPM

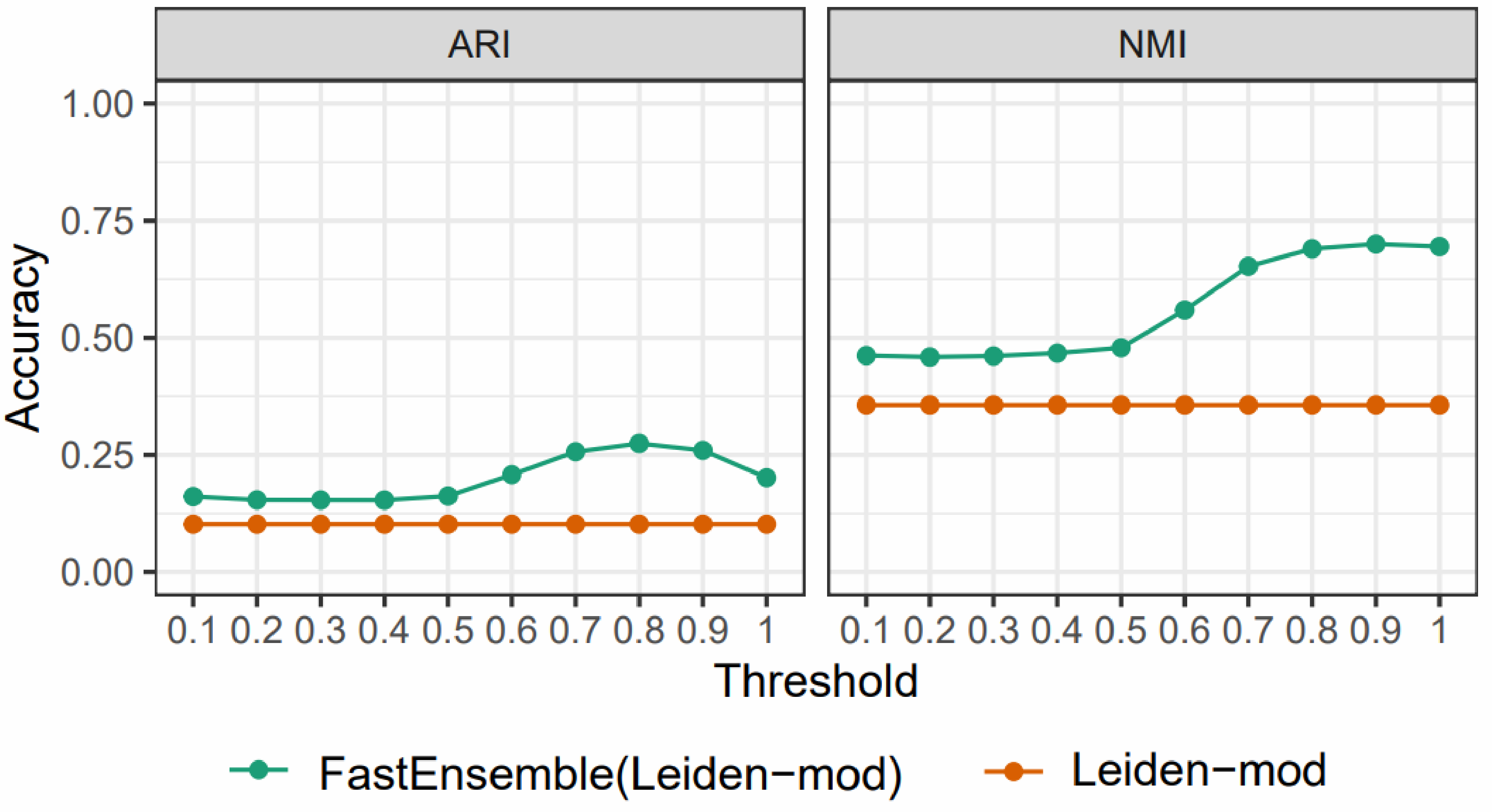
- Experiments:
  - 1: Default parameter exploration
  - 2: Evaluation of modularity pipelines (ECG, FastEnsemble, FastConsensus)
  - 3: Clustering on random graphs (results not shown here)
  - 4: Resolution limit experiment (ring-of-cliques)
  - 5: Evaluation of Leiden-mod and Leiden-CPM with FastEnsemble

Datasets used in the study

Network	Expt.	nodes	edges	mixing param
LFR Training	1,2	10,000	58272-59584	0.196-0.978
Erdős-Rényi	3	1000	470-50,025	1.0
Erdős-Rényi+ LFR	3	2000	4776-53,917	0.486-0.572
Ring-of-Cliques	4	90-10,000	4140-460,000	0.02
LFR cit_hepph	2,5	34,546	$\sim 431K$	0.086-0.781
LFR wiki_topcats	2,5	1,791,489	$\sim 24M$	0.199-0.793
LFR cen	2,5	3,000,000	$\sim 21M$	0.180-0.646
LFR OC	2,5	3,000,000	$\sim 55M$	0.129-0.871
LFR cit_patents	2,5	3,774,768	$\sim 16M$	0.114-0.807

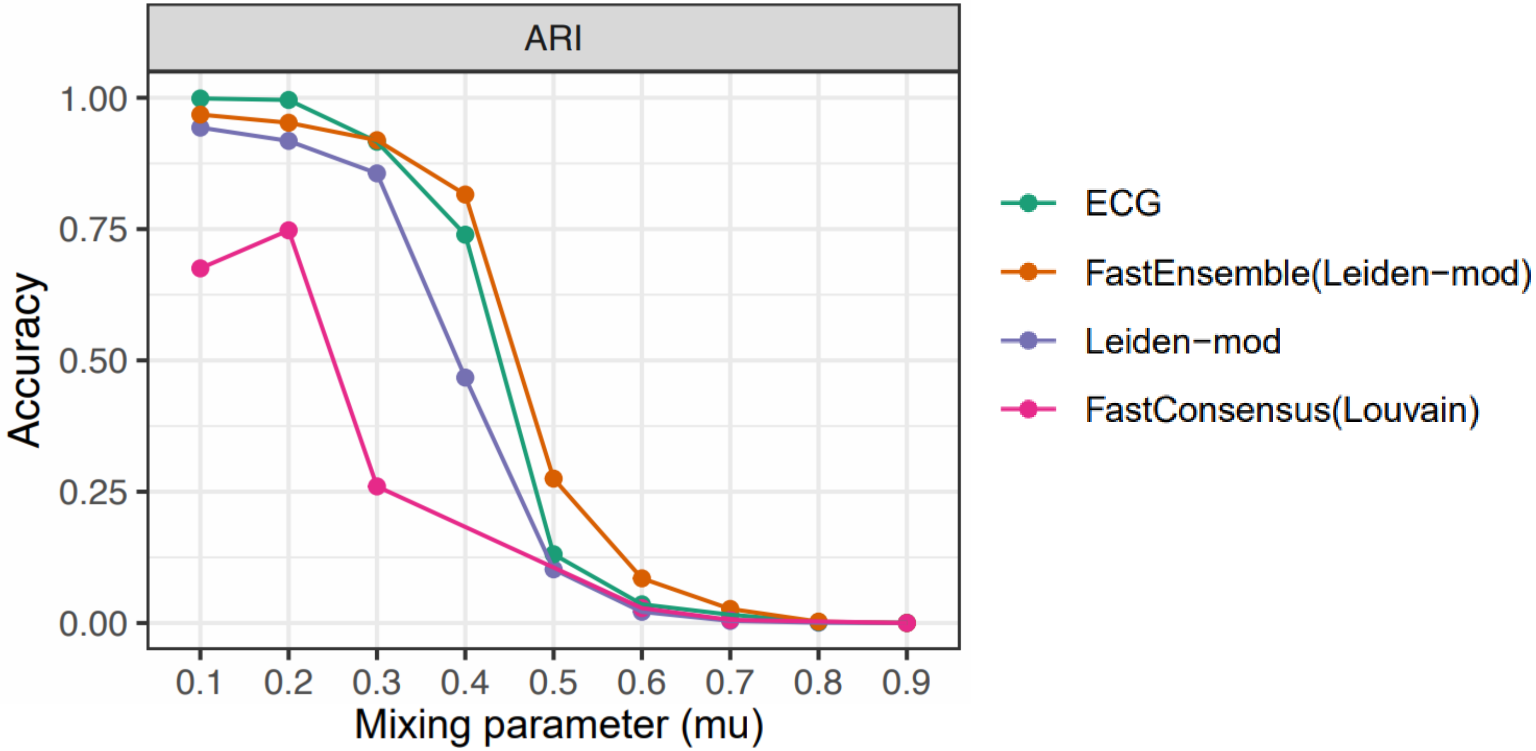
- Low mixing parameter networks are easy to cluster

Setting the default threshold *t*



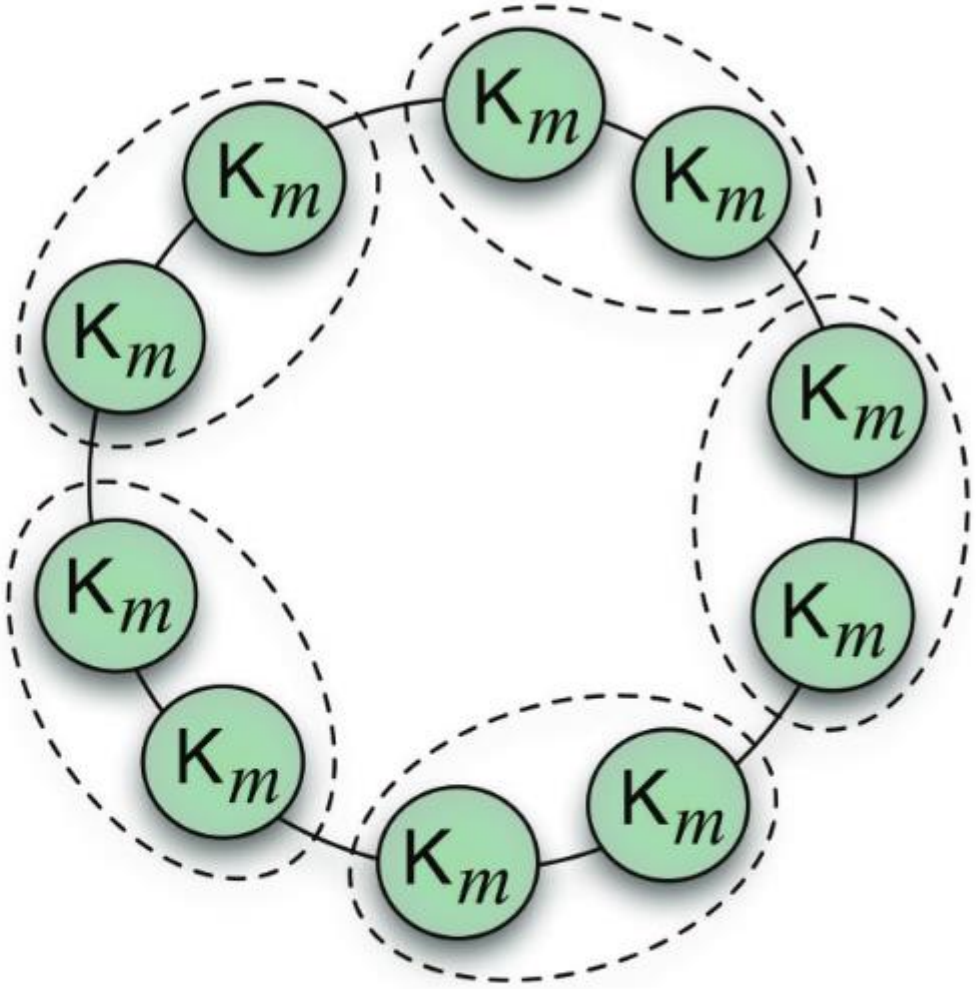
- Training dataset used was 10k-node LFR synthetic networks with varying mixing parameters
- Results shown here are for mixing parameter 0.5
- t = 0.8 selected

Results on Synthetic LFR Networks with Varying Mixing Parameter



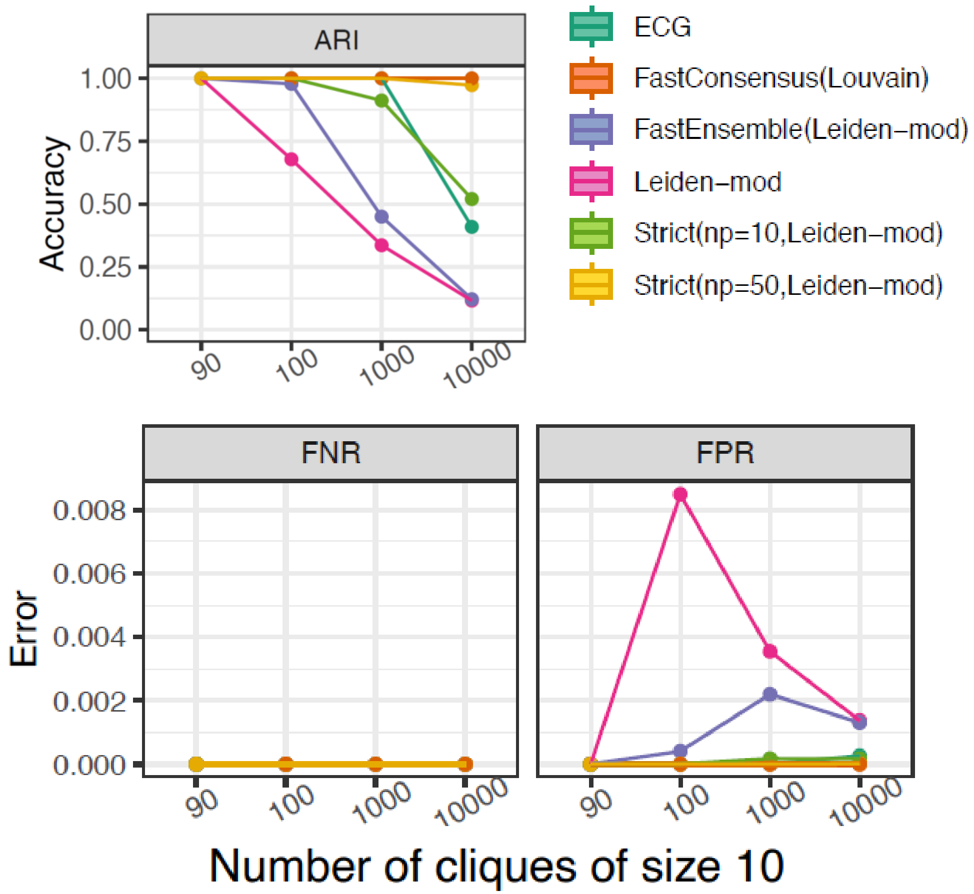
- Training datasets with varying mixing parameters
- ECG best for mixing parameters < 0.4
- FastEnsemble best for mixing parameters >= 0.4

Ring-of-cliques and Resolution Limit



- Figure from Fortunato and Barthelemy. PNAS 2007
- Modularity optimization will group adjacent cliques into a single cluster as the number of cliques increases
- Theory predicts correct clustering given at most 90 cliques of size 10 but afterwards will merge cliques

Ring of 10-cliques Network Results



- Strict: FastEnsemble(*t* =1)
- Trends:
  - All methods have 0 FNR (no cliques split)
  - Leiden-mod had the worst accuracy
  - FastEnsemble second worst accuracy
  - ECG and Strict(np=10) nearly as accurate as top methods
  - FastConsensus and Strict(np=50) most accurate

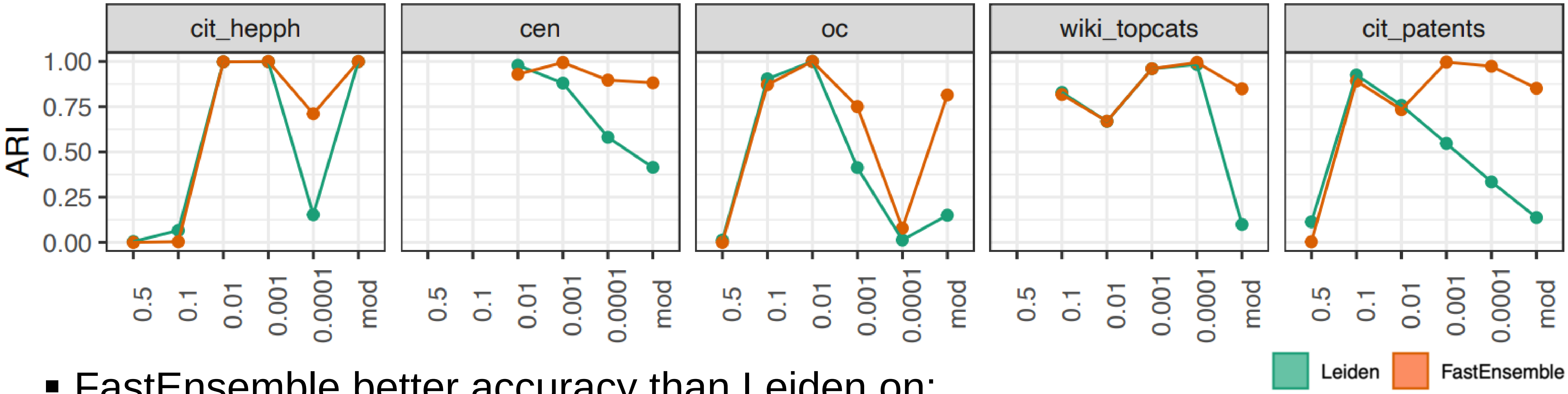
LFR Networks from Park et al. CNA 2023, PLOS Complex Systems 2024

- Leiden-CPM(γ) where γ is the resolution parameter
- Dataset generation:
  - Compute numeric parameters based on an empirical network and clustering
  - Provide numeric parameters to LFR

- Evaluation:
  - Re-cluster LFR network using the same clustering method
  - Cluster LFR network using FastEnsemble given the same clustering method

- Note: some LFR created networks were omitted
  - LFR failed to compute on CEN 0.1, 0.5 with provided parameters
  - wiki_topcats 0.5 has disconnected ground truth clusters

Real-world inspired Synthetic LFR Network Results



- FastEnsemble better accuracy than Leiden on:
  - Leiden-mod based networks
  - Low resolution value Leiden-CPM based networks

- Note: Mixing parameter small for Leiden-mod and Leiden-CPM with low resolution parameter values, increases with resolution parameter

Scalability to Large Datasets (3 million nodes)

		NMI	runtime
LFR cen mod	FastEnsemble(default)	0.988	12h 8m 47s
	FastConsensus	n.d.	>2d
	ECG	0.980	12h 38m 1s
	Leiden-mod	0.897	2m 31s
LFR oc mod	FastEnsemble(default)	0.989	1d 3h 52m 6s
	FastConsensus	n.d.	>2d
	ECG	0.948	21h 58m 30s
	Leiden-mod	0.838	3m 37s

- n.d. indicates no output after 48 hours
- Leiden-mod extremely fast but less accurate
- FastConsensus fails to complete on these networks
- ECG vs FastEnsemble: similar runtimes, slight accuracy improvement for FastEnsemble

Summary

- FastEnsemble increases robustness of input clustering method, especially for small mixing parameters
- FastEnsemble vs ECG:
  - ECG more accurate on lower mixing parameter
  - FastEnsemble more accurate on higher mixing parameters
- FastEnsemble vs FastConsensus:
  - Mixed relative accuracy
  - FastEnsemble more scalable
- StrictConsensus almost as accurate as FastConsensus on ring-of-cliques network:
  - Useful for avoiding false discovery

Future work

- Combining different clustering methods
- Evaluation based on FNR, FPR, and AGRI (Poulin, V. and Théberge, F., IEEE Transactions on Pattern Analysis and Machine Intelligence 2020)
- Input graphs with edge weights

Grainger College
of Engineering

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN