Hierarchical versus Flat Communities

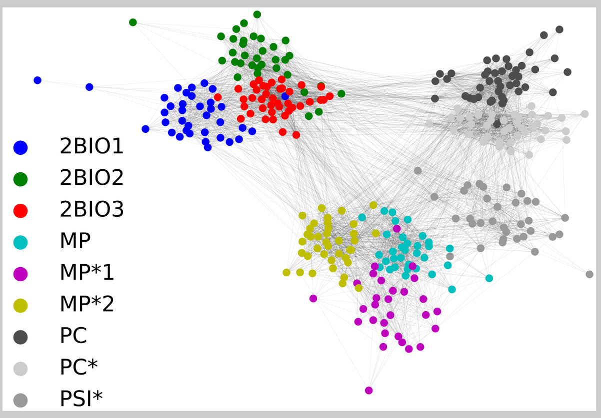
Daichi Kuroda, Maximilien Dreveton, Matthias Grossglauser, Patrick Thiran

INDY (Information and Network Dynamics), EPFL

Background & Motivation

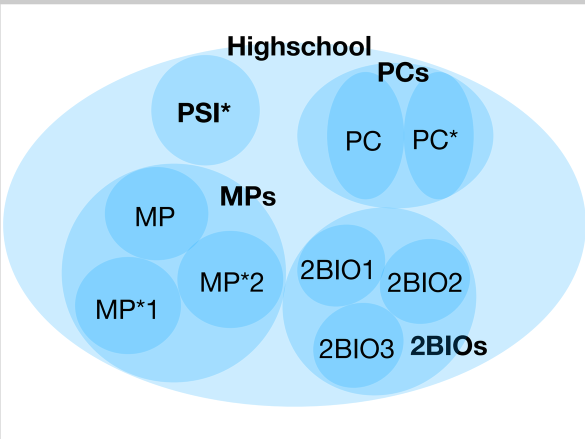
Community Detection

- Networks are commonly used to represent datasets of pairwise interactions (e.g, human interactions [1], biological networks, transportation networks).
- Most datasets have community structure → Community Detection.

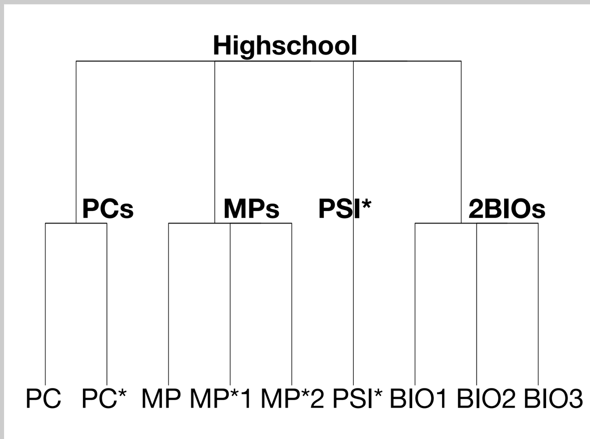


Hierarchical Community Detection

Community structure is often hierarchical.



(a): Classes and specializations



(b): Dendrogram

Several Methods for Hierarchical Community Detection

Bottom-up [2]:

- first identify the primitive communities $\boldsymbol{b} = \{b_1, b_2, \cdots, b_K\}$ and then repeatedly merge the communities using a *linkage* method;
- recall of the super communities is dependent on the accuracy of recovering $\boldsymbol{b}$.

Others: Recursive bipartioning [3], Bayesian [4], Louvain [5] , etc.

Proposed Definition of the Hierarchical Tree

Definition (Hierarcchical Tree) A rooted tree $\mathcal{T}$ is a *hierarchical tree* for the primitive communities $\boldsymbol{b} = \{b_1, \cdots, b_K\}$ w.r.t. a similarity function $s()$ if any vertex t on $\mathcal{T}$ satisfies

$$\min_{b_i, b_j \in \mathcal{L}_{\mathcal{T}[t]}, b_k \notin \mathcal{L}_{\mathcal{T}[t]}} s(b_i, b_j) - s(b_i, b_k) > 0, \tag{1}$$

where $\mathcal{L}_{\mathcal{T}}[t]$ are the leaves of the subtree $\mathcal{T}[t]$ of $\mathcal{T}$ rooted at t.

Among all the hierarchical trees, we focus on the one with the largest number of vertices = *maximum-vertices hierarchical tree*

Motivation

- i, j, k, s.t. $\mathrm{dlca}_{\mathcal{T}}(b_i, b_j) > \mathrm{dlca}_{\mathcal{T}}(b_i, b_k)$ should indicate $s(b_i, b_j) > s(b_i, b_k)$, where $\mathrm{dlca}_{\mathcal{T}}(b_i, b_j)$ is tree distance from the root to the least common ancestor of b_i b_j.
- If primitive communities b_i and b_j belong to a super community, b_i is more similar w.r.t. $s()$ to b_j than to any primitive community b_k which does not belong to the super-community.
- Maximum-vertices hierarchical tree is the most informative.
- A star graph, i.e., a tree where all leaves are directly connected to the root, always satisfies the condition → this can be regarded as flat communities.

Theorem (Uniqueness) The maximum-vertices hierarchical tree for a graph G with primitive communities $\boldsymbol{b}$ w.r.t. a similarity function $s()$ is unique.

Reference

1. R. Mastrandrea, J. Fournet, and A. Barrat, “Contact patterns in a high school: A comparison between data collected using wearable sensors, contact diaries and friendship surveys,” *PLOS ONE*, vol. 10, no. 9, pp. 1–26, Sep. 2015.
2. M. Dreveton, D. Kuroda, M. Grossglauser, and P. Thiran, "When does bottom-up beat top-down in hierarchical community detection?," 2023. arXiv: 2306.00833 [cs.SI].
3. T. Li, L. Lei, S. Bhattacharyya, et al., “Hierarchical community detection by recursive partitioning,” *Journal of the American Statistical Association*, vol. 117, no. 538, pp. 951–968, 2022.
4. P. Tiago, “Hierarchical Block Structures and High-Resolution Model Selection in Large Networks.” *Physical Review X*, 4.1, ,2014: 011047.
5. V. D Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, 2008, "Fast unfolding of communities in large networks.", *Journal of statistical mechanics: theory and experiment* 2008, 10 (2008), P10008.
6. B. Kamiński, P. Pralat, and F. Théberge, “Artificial benchmark for community detection (abcd)—fast random graph model with community structure,” *Network Science*, vol. 9, no. 2, pp. 153–178, 2021.

Recovering the Hierarchical Trees

Algorithm 1

Input: Primitive communities $\boldsymbol{b} = \{b_1, b_2, \cdots, b_K\}$, similarity matrix S, where $S_{b_i b_j} = s(b_i, b_j)$.

Output: Hierarchical tree $\mathcal{T}$.

Process:

1. Apply *linkage* algorithm to the similarity matrix S and obtain a binary tree $\mathcal{T}_{bin}$.
2. Initialize $\mathcal{T}$ as $\mathcal{T} = \mathcal{T}_{bin}$.
3. For all vertex $t \in \mathcal{T}_{bin}$ (starting from the bottom of the tree), merge t to parent vertex of t as done by the following steps, iff t does not satisfies the condition (1);
   - connect links between the parent vertex of t and the children vertices of t:
   - then delete t from $\mathcal{T}$.

Return: $\mathcal{T}$

Theorem (Recovering the Hierarchy) Algorithm 1 recovers the maximum-vertices hierarchical tree.

Finding Hierarchy in Practice

Definition (Approximately Hierarchical Tree) A rooted tree $\mathcal{T}$ is *approximately hierarchical tree* for the primitive communities $\boldsymbol{b} = \{b_1, \cdots, b_K\}$ w.r.t. a similarity function $s()$ if any vertex t on $\mathcal{T}$ satisfies

$$\frac{\sum_{\{b_i, b_j, b_k\} \in \mathcal{L}^2_{\mathcal{T}[t]} \times (\mathcal{L}_{\mathcal{T}[\text{parent of } t]} \setminus \mathcal{L}_{\mathcal{T}[t]})} \mathbf{1}_{\{\hat{s}(b_i, b_j) - \hat{s}(b_i, b_k) > \epsilon\}}}{\left| \{b_i, b_j, b_k\} \in \mathcal{L}^2_{\mathcal{T}[t]} \times \left(\mathcal{L}_{\mathcal{T}[\text{parent of } t]} \setminus \mathcal{L}_{\mathcal{T}[t]} \right) \right|} \geq 1 - \delta. \tag{2}$$

Motivation

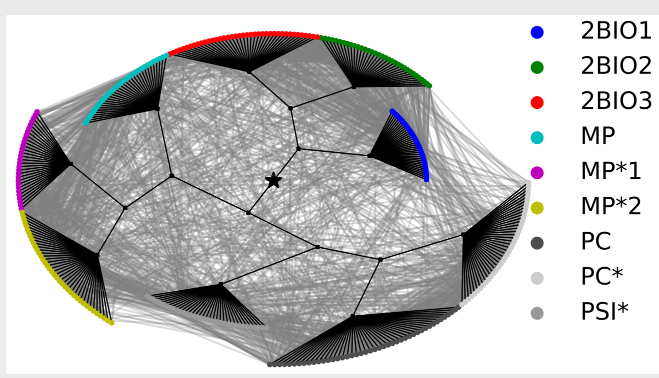
- In reality, you normally only have access to noisy observation $\hat{s}$.
- Want to avoid finding spurious levels → introduce ϵ.
- Want to have some buffer to be resistant to some noise or outlier → introduce δ.

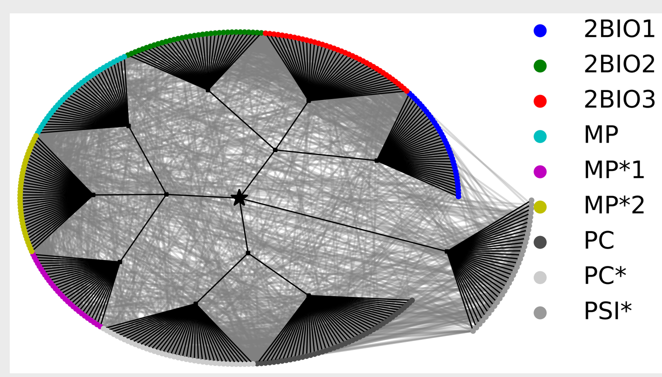
Finding the approximately hierarchical trees is done by using Algorithm 1, but instead of using condition (1), use condtition (2).

Numerical Results

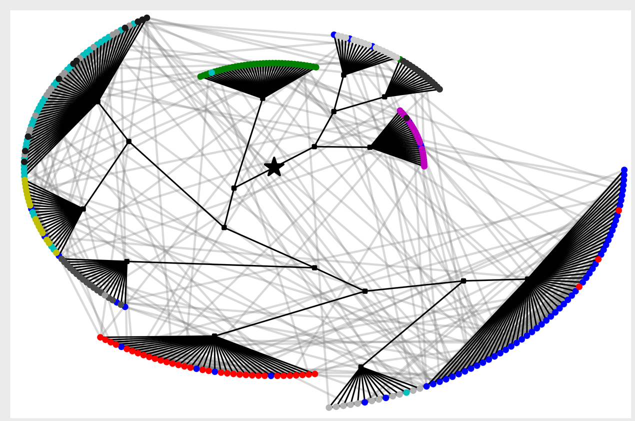
Bottom-up [2] | Algorithm 1

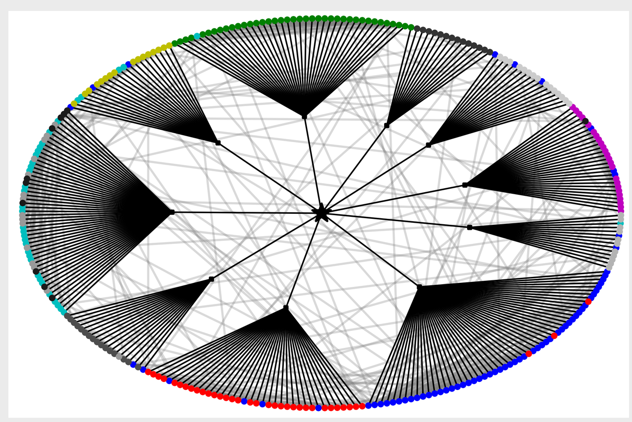
High School [1]





ABCD Model [6]





Contributions

1. Introduced a natural and concrete definition of the hierarchical trees;
   - uniqueness of maximum-vertices hierarchical tree:
   - differentiating flat communities as star graph trees.
2. Proposed an algorithm to discover the maximum-vertices hierarchical trees and established a guarantee of it.