HGNN Algorithm

Jarred M. Kvamme[1], Boyu Zhang[2], and Audrey Q. Fu[1,3]

[1]*Department of Bioinformatics and Computational Biology - University of Idaho*
[2]*Department of Computer Science - University of Idaho*
[3]*Department of Mathematics and Statistical Science - University of Idaho*

August 24, 2023

1 Preprocessing

1.1 Given the node feature matrix $\mathbf{X} \in \mathbb{R}^{\mathbf{N}\times\mathbf{p}}$ compute the adjacency matrix $\mathbf{A_0}$ corresponding to the input graph $\mathcal{G}_0(E, N)$ of E edges and N vertices by one of the following methods

1.1.1 **Correlations method:** Compute the correlation matrix $\mathbf{R} \in \mathbb{R}^{\mathbf{N}\times\mathbf{N}}$ from $\mathbf{X}$. Convert the correlation matrix $\mathbf{R}$ into an adjacency matrix $\mathbf{A_0}$ such that

$$a_{i,j} = \begin{cases} 1 & \text{if} \;\; r_{i,j} > \rho \\ 0 & \text{else} \end{cases}$$

where ρ represents a minimum correlation threshold to consider an edge $e_{i,j}$ between nodes i and j.

1.1.2 **K-neighbors method: For all nodes** i - compute the distance from node i to all other nodes. Link node i to the first k nodes with smallest distances

1.1.3 **Precision method:**

1.2 Construct the consensus hierarchy $\mathcal{H}_0$ of the input graph $\mathcal{G}_0$..

1.2.1 **For** $i \in \{1 : n\}$ **repetitions:** Using the Louvain Algorithm, construct a hierarchy $\mathcal{H}^{(i)}$ which represents a stochastic partition of $\mathcal{G}_0$ into a set of $\ell - 1$ sub-graphs such that $\mathcal{H}^{(i)} = \{\mathcal{S}_0, \mathcal{S}_1^{(i)}, \mathcal{S}^{(i)}, \cdots, \mathcal{S}_{\ell-1}^{(i)}\}$ where $\ell = |\mathcal{H}^{(i)}|$ is the number of layers in the hierarchy with $\mathcal{S}_0 = \mathcal{G}_0$ representing the starting (or bottom) layer.

1.2.2 Let $M = \{m_i\}_{i=1}^{n}$ contain the number of super layers in all n hierarchies such that $m_i = |\mathcal{H}^{(i)}| - 1$. Next let $\zeta \in \{1, 2, ...(\ell - 1)\}$ denote the set of possible numbers of super-layers for a given hierarchy. Compute the number of super-layers for the consensus hierarchy $\mathcal{H}_0$ via popular vote:

$$m_0 = \arg\max_{\zeta} f(M) \;\; \text{where} \;\; f(M) = \frac{\sum_{i\in\zeta} I(M = i)}{n}$$

1.2.3 Compute the number of communities in each super-layer of the consensus hierarchy $\mathcal{H}_0$ by agglomerating the community assignments for a given layer across all n computed hierarchies from **step 1.2.1**.
For all $j \in \{1 : m_0\}$**:** agglomerate the community assignments for all i subgraphs $\mathcal{S}_j$ corresponding to the j^{th} super layer in all i hierarchies.

1.3 For each layer $\mathcal{S}_j$ in $\mathcal{H}_0$, gather all edges connecting $\mathcal{S}_j$ to its adjacent super-layer $\mathcal{S}_{j+1}$. These are the **"down-to-up"** edges. Let $p_i^{(j)}$ represent the collection of edges which maps the path from node η_i in the bottom layer to the j^{th} super-layer. Then this path represents the message-passing channel by which the i^{th} node N_i can communicate with k^{th} super-level node N_k in the j^{th} super layer

1.4 For each super

1.5 Set up HC-GNN model

1.6 Generate within-level community detection autoencoders

2 Training