Hierarchical Clustering and Overlapping Module Detection in Protein-Protein Interaction Networks

Ilse ten Boske* Nicola Greco†

1 Background

Proteins are essential building blocks in all living organisms, serving as the machinery for biological processes within cells. They perform diverse functions, such as providing structural support, catalyzing biochemical reactions through enzymes, and facilitating signal transmission via signaling proteins. Rather than operating independently, proteins typically interact to form complexes, which are assemblies that collaborate to execute specific biological functions. These protein-protein interactions (PPIs), primarily governed by electrostatic interactions, are crucial for nearly all cellular processes, emphasizing the importance of studying these interactions to comprehend cellular mechanisms. Significant advancements in detecting and predicting PPIs have been achieved through both traditional and high-throughput techniques [1], which nowadays allows the rapid identification of thousands of interactions, offering a broader view of cellular activities. This progress has led to the generation of extensive interaction data, enhancing our understanding of PPI networks and enabling the construction of large-scale PPI networks.

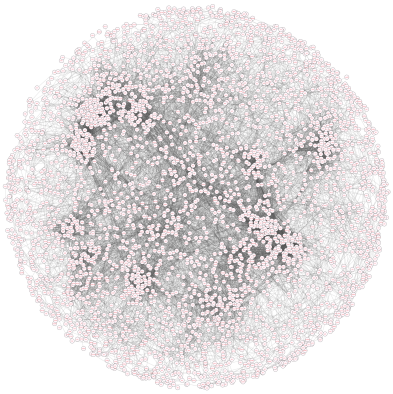

Figure 1: **Example of PPI network of *Saccharomyces Cervisiae*** Obtained from STRING database[2].

Large-scale PPI data are naturally represented as undirected graphs, in which vertices correspond to proteins and edges to their interactions (figure 2). These networks share topological features common to real-world networks, such as those in social and communication domains. Notably, PPI networks are scale-free, characterized by a power-law degree distribution, exhibit the small-world property, marked by short path lengths, and shows a high degree of clustering, indicating a tendency for nodes to form tightly-knit groups [3].

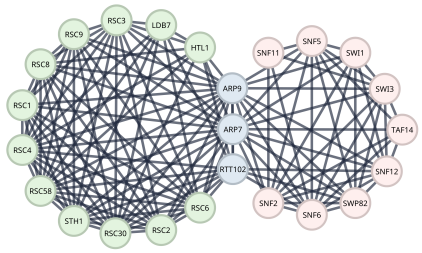

Figure 2: **Example of Overlapping Protein Complexes in PPI network or *S. Cervisiae*.** RSC and SNF complexes in green and pink respectively, in ligth-blue their shared components.

Researchers often concentrate on functional

*Ilse ten Boske is with the Graduate School of Natural Sciences, Utrecht University, Utrecht, The Netherlands. Student id: 6787177. Email: i.m.tenboske@students.uu.nl

†Nicola Greco is with the Graduate School of Life Sciences, Utrecht University, Utrecht, The Netherlands. Student id: 2327775. Email: n.greco@students.uu.nl

modules, groups of proteins that share functional properties and are localized in the same network area [1]. These modules, which include one or more protein complexes, perform specific biological functions in distinct temporal and spatial contexts. The identification of these modules or protein complexes within PPI networks is crucial as it elucidates cell structure-function relationships, aids in predicting functions for unannotated proteins, and improves understanding of cellular organization. Furthermore, recognizing these protein complexes is essential for uncovering disease mechanisms and identifying potential therapeutic targets [1].

Graph clustering, an unsupervised learning technique, effectively groups nodes to ensure denser connections within clusters compared to between them. Among various clustering algorithms, the hierarchical nature of PPI networks, where modules exist at multiple scales, makes hierarchical clustering algorithms particularly suited to community detection within PPI [3].

Hierarchical clustering organizes the network's inherent structure into a tree-like hierarchy. These algorithms are categorized into two types: agglomerative (bottom-up) and divisive (top-down). Agglomerative clustering starts with each vertex as a separate cluster and iteratively merges them based on connectivity weights in descending order. Divisive clustering, on the other hand, splits the network into subgraphs using heuristic rules like edge betweenness or minimum cut.

Several hierarchical clustering algorithms have been developed for PPI networks. Luo et al. proposed an agglomerative algorithm called MoNet, which redefines modules by extending the concept of vertex degrees to subgraphs and merging clusters based on modularity, defined as the ratio of indegree to outdegree within subgraphs [9]. Another example is NeMo, introduced by Mete et al., which detects modules using shared neighbors to measure similarity and a collapse procedure to prune insignificant structures, offering a linear time alternative to traditional methods [**?**]. These methods illustrate the ongoing efforts to enhance hierarchical clustering techniques to better capture the complex, hierarchical nature of protein communities.

Despite their utility, hierarchical clustering faces challenges, particularly in addressing the overlapping nature of protein communities. Proteins often belong to multiple complexes and perform varying functions depending on the context, complexities that traditional hierarchical methods sometimes fail to capture [1] [3].

To address these limitations, Wang et al. introduced a novel hierarchical clustering algorithm, OH-PIN [4], which enhances the detection of hierarchically organized modules by introducing the ability to identify overlapping modules. Their studies demonstrated that OH-PIN outperformed traditional non-overlapping hierarchical algorithms in identifying known protein complexes within a Saccharomyces cerevisiae (yeast) network.

Similarly, out of the biological field, Zhao et al. proposed an overlapping hierarchical clustering (OHC) algorithm that also aimed to address the limitations of traditional agglomerative hierarchical clustering. By considering overlapping communities, they found that their OHC algorithm performed better in capturing the community structures compared to non-overlapping agglomerative methods [8].

As the OH-PIN authors specifically tested their algorithm on a single PPI interaction network, we wanted to build on the promising results observed and assess again its performance, to further prove the added value of considering overlapping modules in PPI. In this experimental study, we will so re-evaluate the OH-PIN algorithm's ability to recover known protein complexes in a different and larger PPI network, the Escherichia coli PPI network, and compare its performance against other hierarchical and non-hierarchical clustering algorithms not previously explored by Wang et al. Furthermore, we will test these algorithms at various resolutions or cuts, exploiting a key feature of hierarchical clustering—its ability to produce clusterings at different granularities. Our objective is to further assets whether incorporating the inherent overlap of protein modules into the hierarchical clustering process can indeed improve the identification of protein complexes.

2 Methods

2.1 Clustering Algorithms

To provide a comprehensive analysis, we compared OH-PIN against two other prominent clus-

tering algorithms: the Louvain method and the Girvan-Newman hierarchical clustering algorithm.

2.1.1 OH-PIN

In their study, the authors employed a hierarchical clustering algorithm to identify overlapping and hierarchical functional modules, beginning with overlapping subgraphs as initial clusters.

Drawing on biological models regarding the evolution of proteins and their interactions, the concept of $B_{cluster}$ was introduced to identify the initial overlapping clusters for the OH-PIN algorithm. A $B_{cluster}$ for an edge (u, v) is defined as comprising vertices u and v along with their common neighbors, if any:

$$B_{cluster}(u, v) = \{u\} \cup \{v\} \cup C(u, v) \text{ if } |C(u, v)| > 0$$

$$B_{cluster}(u, v) = \emptyset \text{ if } |C(u, v)| = 0$$

where $C(u, v)$ denotes the set of common neighbors of vertices u and v. To prevent redundancy, any $B_{clusters}$ that are subsets of other $B_{clusters}$ are eliminated. The non-redundant clusters, termed $M_{clusters}$, serve as the starting point for the algorithm.

Following the establishment of initial clusters, the OH-PIN algorithm conducts an initial merging phase targeting highly overlapping clusters using an overlapping score (OS) to measure overlap between cluster pairs $C1$ and $C2$:

$$OS(C1, C2) = \frac{|\{C1 \cap C2\}|^2}{|C1| * |C2|}$$

Clusters with an OS exceeding 0.5 are deemed highly overlapping and are iteratively merged until no pair of initial clusters exhibit an OS above this threshold. This merging phase consolidates highly overlapping clusters before proceeding with actual hierarchical clustering.

The final step of the OH-PIN algorithm involves agglomerative hierarchical clustering, where clusters demonstrating high connectivity are merged. The authors introduced a specific clustering coefficient CVV for cluster pairs to quantify their connectivity. This coefficient is based on the premise that a higher number of connecting edges between two clusters signifies a denser connection, indicating that these clusters likely belong to the same functional module.

Extending this idea further, the authors suggest that clusters connected by edges whose vertices share many common neighbors are more likely to be in the same module. Therefore, the weight of an edge (u, v) is defined as the number of common neighbors, $|C(u, v)|$, between the vertices u and v:

$$w_{u,v} = |C(u, v)|$$

Using this edge weight, the clustering coefficient between two clusters $C1$ and $C2$ is calculated as:

$$CCV(C1, C2) = \frac{\sum_{u \in C1, v \in C2} w_{u,v}}{|V_{C1}| \cdot |V_{C2}|}$$

Here, V_{C1} and V_{C2} denote the vertex sets of $C1$ and $C2$ respectively, with $|V_{C1}| \cdot |V_{C2}|$ representing the maximum potential connections between the two clusters. A higher clustering coefficient implies a greater likelihood of the clusters sharing the same functional module.

Clusters are merged based on the highest clustering coefficients until all are consolidated into λ-modules, defined by a specific threshold λ. A subgraph $C^* \subseteq G$ qualifies as a λ-module if $\lambda_C \geq \lambda$, where:

$$\lambda_C = \frac{\sum_{v \in C} k^{in}(v, C)}{\sum_{v \ inC} k^{out}(v, C)}$$

In this context, $k^{in}(v, C)$ and $k^{out}(v, C)$ are the counts of edges connecting vertex v within and outside the cluster C, respectively.

The clustering process continues until all clusters meet the λ threshold, at which point the current partitioning is returned. The merging step is then repeated with the next λ value. This iterative process enables the algorithm to produce partitions at various levels of resolution, dictated by the progressively increasing λ values provided as input.

Initially, we planned to use the same λ values as those in the original OH-PIN paper. However, our preliminary analysis revealed that these values were too large for our network, which is larger and denser than the network used in the original study. The highest parameters used in the original work resulted in partitions with modularity values close to zero for our network, indicating that nearly the entire network was merged into a single cluster. Consequently, we

opted for a smaller range of λ values (Table 1). As recommended by the authors, we also increased the parameter step size as the λ values grew.

Given that the authors of OH-PIN did not provide any implementation, we implemented the algorithm ourselves in Python from the pseudocode they provided [4]. We used Python3 and NetworkX to handle the graph data structures. The implemented algorithm, as well as all other scripts used in this study, are available on GitHub for reproducibility here.

2.1.2 Girvan-Newman Hierarchical Algorithm

The Girvan-Newman algorithm is a renowned hierarchical clustering method which detect communities within complex networks by focusing on the edges' betweenness centrality. This metric, indicating the number of shortest paths passing through an edge, guides the iterative removal of edges with the highest centrality until the network divides into distinct communities. The procedure involves: calculating the betweenness centrality for each edge, removing the edge with the highest centrality, recalculating centrality for remaining edges, and repeating this process until all communities are isolated. The output is a dendrogram that depicts the community structure, which can be cut at various levels to determine the optimal community divisions. We utilized an existing implementation of the Girvan-Newman algorithm in Python [7], which we modified to allow for the extraction of clusters at various cut levels of the dendrogram. To facilitate a comparative analysis between the OH-PIN and Girvan-Newman algorithms, we generated multiple clustering from the same dendogram to produce partitions with a number of clusters corresponding to those generated by OH-PIN at various λ thresholds. Further information on the clustering methods can be found in the supplementary materials.

2.1.3 Louvain Algorithm

The Louvain algorithm optimizes the modularity of a network's partition, reflecting the density of edges within versus between communities. It operates through two iterative phases: modularity optimization, where each node is evaluated for potential community reassignment based on modularity gain, and community aggregation, where a new network is constructed from the identified communities, and the optimization process repeats. By adjusting the resolution (Table 2), we ensure that the number of clusters detected by the Louvain algorithm aligns with the results from OH-PIN, allowing for a direct comparison of their performance. We used an implementation of the algorithm that is in the Python package NetworkX.

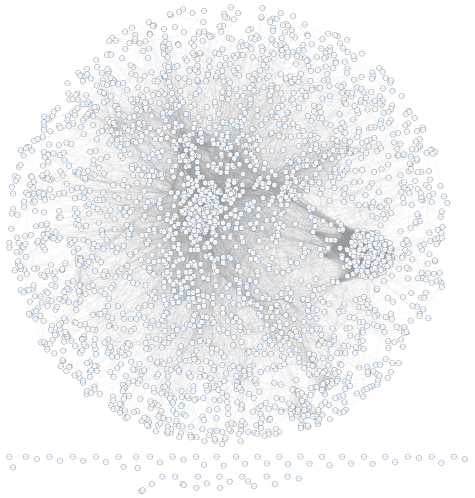

Figure 3: **Visualization of the *Escherichia coli* PPI network.** Retrieved from the STRING database. Distinct community structures, indicative of potential functional modules, are observable even from this broad overview.

2.2 Data

For the evaluation of the clustering algorithms, we utilized the protein-protein network data retrieved from the STRING database (Search Tool for the Retrieval of Interacting Genes/Proteins) [2]. The STRING database is designed to assimilate, score, and integrate all publicly accessible sources of protein-protein interaction information, supplementing these with computational predictions to provide a comprehensive resource for the study of protein interactions. Our analysis focused on the PPI network of the bacterium *Escherichia Coli*, shown in Figure 3, a widely utilized model organism in biological research. This particular network was selected due to its relatively smaller size, comprising 3,043 proteins (nodes) and 42,658 physical interactions (edges), which facilitated a more manageable computational analysis compared to the larger and more complex networks of other organisms, such as human or yeasts. The physical interactions in-

cluded in this dataset are experimentally determined, excluding computationally predicted links, to enhance the reliability of the network data.

To benchmark the performance of the clustering algorithms, we compared the clusters identified from the PPI network against the experimentally determined protein complexes of *E. Coli*. The reference data for these known protein complexes was obtained from the Complex Portal from the European Bioinformatics Institute (EBI), a manually curated database that provides comprehensive information on stable, macromolecular complexes of established biological function [5].

2.3 Comparison of Algorithms

To evaluate the performance of the clustering algorithms, they were all applied to the *E. Coli* protein interaction network. We adjusted the parameters of the Louvain and Girvan-Newman algorithms to produce a range of clustering outcomes at different resolutions, to ensure that the number of clusters generated by these algorithms was comparable to those identified by the OH-PIN algorithm at different lambda thresholds.

2.3.1 Metrics for Algorithm Comparison

Given the overlapping nature of the algorithm employed in this study, traditional metrics for assessing the efficacy of protein complex detection were found to be insufficient. Initial metrics were initially drawn from a prior study [6], but during preliminary analysis, metrics like Sensitivity exhibited abnormal behaviours due to the overlapping natures of the OH-PIN found clusters, because they did not accommodate the overlapping and repeated presence of proteins across multiple clusters. Similar problems were found with other popular methods that allows to compare clustering results to ground truth.

To address the challenges posed by the overlapping outputs of our clustering algorithms, we modified our evaluation metrics to better reflect the complexities inherent in our data. Specifically, we adjusted the criteria for matching known protein complexes with detected clusters. Instead of defining a match based solely on an Overlap Score (OS), we focused on maximizing coverage within the detected clusters. Coverage was defined mathematically as:

$$\text{Coverage} = \frac{|\text{Cluster} \cap \text{Known Complex}|}{|\text{Known Complex}|}$$

A known complex was considered accurately identified by an algorithm if the highest coverage value among the detected clusters was above 0.7 (i.e. if there was at least one cluster which recovered 70% of the components of that complex), to ensure that a significant part of the complex was recovered.

Using this refined approach, we recalculated the traditional metrics to evaluate each algorithm's performance. The metrics were defined as follows:

- **Precision**: The ratio of true positive clusters (complexes correclty matched by a found cluster) to the total number of clusters detected by the algorithm.

$$\text{Precision} = \frac{\text{True Positives Clusters}}{\text{Total Found Clusters}}$$

- **Recall**: The ratio of true positive clusters to the total number of known complexes in the dataset.

$$\text{Recall} = \frac{\text{True Positives Clusters}}{\text{Total Complexes}}$$

- **F-measure**: The harmonic mean of precision and recall, providing a balance between the two by penalizing extreme values.

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

These modifications ensure that our metrics are adapted to evaluate the performance of clustering algorithms in scenarios where protein complexes may overlap among multiple clusters.

The last metric we used to evaluate the clustering is Modularity, a widely used metric to assess the clustering of networks. Unlike the previously presented metrics, which compare detected clusters to known protein complexes, Modularity evaluates the network's division into modules without prior biological knowledge. It is defined as:

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

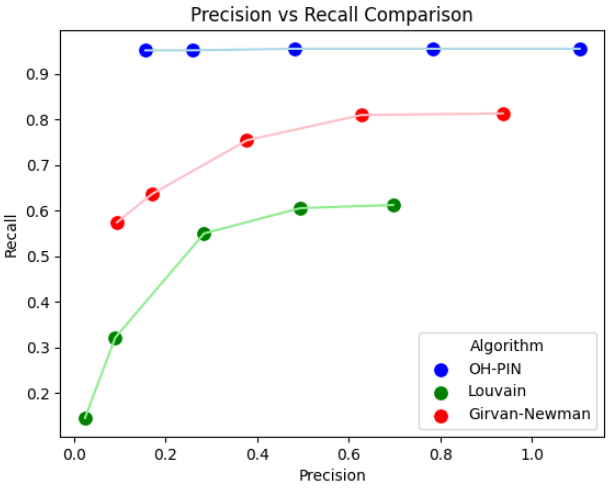

Figure 4: **Precision vs Recall** comparison for OH-PIN, Louvain, and Girvan-Newman algorithms. The OH-PIN clustering algorithm demonstrates superior performance, consistently achieving higher precision and recall values compared to the Louvain and Girvan-Newman algorithms at every level of granularity.

where A_{ij} is the adjacency matrix, k_i and k_j are node degrees, m is the total number of edges, and $\delta(c_i, c_j)$ is 1 if nodes i and j are in the same cluster. High modularity values indicate strong community structure with dense intra-cluster and sparse inter-cluster connections, making it ideal for evaluating clustering algorithms on networks without known ground truth.

3 Results

3.1 Clustering Quality

The performance of three clustering algorithms—OH-PIN, Louvain, and Girvan-Newman—was assessed across various resolutions using Precision, Recall, F-measure, and Modularity as metrics. The first three metrics were calculated against a ground truth of experimentally determined complexes, while Modularity provided a biological knowledge-unaware evaluation of clustering quality.

3.1.1 Performance of OH-PIN

The OH-PIN algorithm demonstrated superior performance across all ground truth-aware metrics when compared to Louvain and Girvan-Newman algorithms, at all tested resolutions. As detailed in Table 1, OH-PIN consistently maintained high Recall rates (approximately 0.96) across all resolution levels, indicating its robustness in correctly recovering known complexes. Notably, the Precision of OH-PIN improved as the resolution decreased, with values ranging from 0.16 at the highest resolution (1762 clusters) to 1.10 at the lowest resolution (250 clusters). This increase in Precision suggests that merging more overlapping clusters at lower resolutions effectively reduces the number of false positives. The F-measure, which balances Precision and Recall, also showed a significant increase with lower resolution, peaking at 1.02 at the lowest resolution tested.

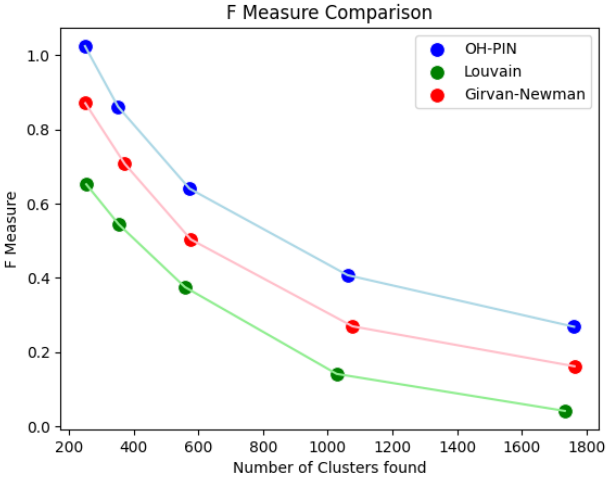

Figure 6: **F Measure** trends across varying cluster counts for OH-PIN, Louvain, and Girvan-Newman algorithms. The OH-PIN algorithm consistently exhibits higher F Measure values, indicating a better ability to recover known protein complexes compared to Louvain and Girvan-Newman. This performance advantage is maintained across different levels of granularity, as represented by the number of clusters found.

Interestingly, Precision values exceeding 1 were observed at higher resolutions where the number of detected clusters fell below the number of known complexes. This phenomenon occurs because a single detected cluster can recover multiple protein complexes, leading to more true positives than the number of clusters. This indicates that these clusters may represent functional modules comprising various protein complexes involved in the same biological function.

When considering Modularity, OH-PIN displayed a positive correlation with clustering resolution, starting at 0.27 for the highest number of clusters and peaking at 0.45 for 352 clusters. This trend reflects the increasing strength of intra-cluster connections and the effective partitioning of the network into distinct modules as resolution decreases.

Table 1: OH-PIN - Final Metrics

NumClusters	Resolution	Precision	Recall	F_measure	Modularity
1762	0.050	0.156	0.952	0.268	0.274
1062	0.100	0.259	0.952	0.407	0.328
573	0.200	0.482	0.955	0.640	0.426
352	0.400	0.784	0.955	0.861	0.446
250	1.000	1.104	0.955	1.024	0.405

Table 2: Louvain - Final Metrics

NumClusters	Resolution	Precision	Recall	F_measure	Modularity
1733	1000.0	0.024	0.145	0.042	0.036
1028	200.0	0.090	0.322	0.141	0.075
559	40.0	0.284	0.550	0.375	0.147
354	11.5	0.494	0.606	0.544	0.203
254	5.0	0.697	0.612	0.652	0.261

Table 3: Girvan-Newman - Other Metrics

NumClusters	Resolution	Precision	Recall	F_measure	Modularity
1763	-	0.094	0.573	0.162	0.400
1077	-	0.171	0.637	0.269	0.529
577	-	0.378	0.754	0.503	0.573
373	-	0.627	0.810	0.707	0.543
251	-	0.936	0.813	0.870	0.545

3.1.2 Louvain and Girvan-Newman Algorithms

The Girvan-Newman algorithm, while not matching the performance of OH-PIN, consistently outperformed the Louvain algorithm in terms of both Precision and Recall. Regarding Modularity, Girvan-Newman exhibited the highest values, starting at 0.40 for the highest number of clusters and reaching up to 0.57 for 577 clusters. This suggests that the strategy of removing edges with the highest betweenness, is particularly effective at finding partitions that minimize the edges between different clusters and at identifying strongly connected modules within the network at various resolutions. Louvain, although showing improvement in Modularity as the number of clusters decreased, peaked at 0.26 for 254 clusters, still below the levels achieved by OH-PIN and Girvan-Newman. Despite being a modularity optimization-based algorithm, these poor results may be due to it not being intended for use with such high-resolution parameters.

3.2 Computational Efficiency

In our study, we prioritized the qualitative evaluation of clustering algorithms; however, the computational efficiency of each algorithm also warrants discussion due to their varied running times and implications for larger-scale applications.

3.2.1 Overview of Running Times

Despite several optimizations that allowed us to reduce the number of operations and running time of the algorithm, our implementation of the OH-PIN algorithm exhibited a particularly lengthy runtime, requiring approximately 1 day, to process the Escherichia coli PPI network, which includes 3,043 nodes and 42,658 edges. This extended duration is primarily attributed to the specifics of the algorithm and on the programming language used (Python). Similarly, the Girvan-Newman algorithm originally demonstrated a comparable runtime of about a day for the same dataset, which we manadge to decrease significantly by parallelizing the calculation of betweenness centrality, which is recalculated in each iteration and is computationally intensive.

Contrastingly, the Louvain algorithm completed its analysis in just minutes on the same network, underscoring a significant disparity in efficiency between the algorithms.

The computational complexity of the OH-PIN algorithm is heavily influenced by the number of initial $M_{clusters}$ prior to hierarchical aggregation. Initially, the Clustering Coefficient Value (CCV) is calculated for every pair of clusters to identify the most interconnected ones for merging. This step, and the subsequent recalculations of CCV after each merge, grow quadratically with the number of initial clusters, rendering the complexity of this phase $O(C^2)$.

Furthermore, each edge in the network con-

tributes to forming an initial B-cluster, which includes the edge vertices u and v along with their common neighbors. The initial number of clusters created is directly proportional to the number of edges, assuming each edge could potentially form a distinct cluster. As the initial set of $M_{clusters}$ is derived from non-highly overlapping $B_{clusters}$, the number of clusters subjected to hierarchical agglomeration is linearly dependent on the number of edges, leading to an approximate cost for this phase of $O(E^2)$.

This quadratic complexity in relation to the number of edges means that OH-PIN does not scale well on dense networks. This scaling issue was a decisive factor in limiting our tests to the E. coli network. The computational demands of both OH-PIN and Girvan-Newman (which has a complexity of $O(NE^2)$ in sparse graphs and can approach $O(N^3)$ in dense graphs), exacerbated by inefficiencies in our current implementations, restrict their practical application to larger or denser networks without substantial optimization.

In conclusion, while OH-PIN and Girvan-Newman offer high-quality clustering results, their computational inefficiencies highlight significant challenges, particularly for larger datasets. The Louvain algorithm presents a more computationally feasible option, suggesting that the choice of algorithm for PPI network analysis should consider both the quality of clustering and the practicality of implementation across various network sizes and complexities.

4 Conclusion and discussion

This study assessed the OH-PIN algorithm's performance against traditional clustering methods like Louvain and Girvan-Newman in clustering PPI networks. Our findings confirm that OH-PIN's novel approach to managing overlapping clusters substantially enhances the detection and identification of protein complexes, aligning with the results reported by Wang et al. This capability significantly improves the precision and accuracy of biological functionality exploration within the networks, surpassing other hierarchical and modularity-based clustering methods. OH-PIN's ability to analyze clusters at various resolutions is particularly beneficial, providing insights into complex structures from large, well-defined groups to smaller, more subtle formations. This flexibility is essential for thorough biological analyses but comes at the expense of computational efficiency. The extensive running times and limited scalability of OH-PIN, as observed in our tests on the *E. Coli* PPI network, restrict its use to smaller or less dense networks. The Girvan-Newman algorithm faced similar computational challenges, although optimizations such as the parallelization have somewhat alleviated these issues. Another limitation of this study stems from the relatively small number of experimentally determined protein complexes known for *E. Coli*. From the Complex Portal, we retrieved only 288 protein complexes out of the 3,043 nodes in the network, limiting our ability to comprehensively assess the algorithm's full potential in identifying complexes. This constrained set of known complexes may also contribute to the exceptionally good results observed for the OH-PIN algorithm, as the limited benchmark could potentially skew the performance metrics favorably. Expanding the analysis to other organisms, such as *S. Cerevisiae*—which has 629 known complexes across 5,925 proteins—could provide a more comprehensive and reliable assessment, offering a broader base for benchmarking the algorithms. Initially, we aimed to test the algorithm across a broader range of networks, but computational inefficiencies curtailed these plans. Consequently, we confined our evaluation to the *E. Coli* network, rather than expanding our tests to more and larger networks than what initially done by the OH-PIN authors. To fully exploit the potential of incorporating overlapping in hierarchical clustering algorithms, which accurately reflects the complex and layered structures of modules in PPI networks, future efforts should focus on enhancing computational performance. Enhancing algorithm efficiency would facilitate more extensive testing and broader application, enabling researchers to apply these methods to more complex and larger networks, thus broadening our understanding of biological interactions at the molecular level.

References

[1] Bhowmick, S. S., and Seah, B. S. "Clustering and summarizing protein-protein interaction networks: A survey." IEEE Transactions on

Knowledge and Data Engineering 28.3 (2015): 638-658.

[2] Szklarczyk, D., Gable, A. L., Lyon, D., Junge, A., Wyder, S., Huerta-Cepas, J., ... and von Mering, C. "STRING v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets." Nucleic Acids Research 47.D1 (2019): D607-D613.

[3] Meng, X., Li, W., Peng, X., Li, Y., and Li, M. "Protein interaction networks: centrality, modularity, dynamics, and applications." Frontiers of Computer Science 15 (2021): 1-17.

[4] Wang, J., Ren, J., Li, M., and Wu, F. X. "Identification of hierarchical and overlapping functional modules in PPI networks." IEEE Transactions on Nanobioscience 11.4 (2012): 386-393.

[5] Meldal, B. H. M., Bye-A-Jee, H., Gajdoš, L., Hammerová, Z., Horáčková, A., and Melicher, F. "Complex Portal 2018: extended content and enhanced visualization tools for macromolecular complexes." Nucleic Acids Research 47.D1 (2019): D550-D558.

[6] Wu, Z., Lin, Q., and Liu, B. "A comprehensive review and evaluation of computational methods for identifying protein complexes from protein–protein interaction networks." Briefings in Bioinformatics 21.5 (2020): 1531-1548.

[7] Kazem, J. "Girvan-Newman Algorithm Implementation." GitHub. https://github.com/kjahan/community

[8] Jeantet, I., Miklós, Z., and Gross-Amblard, D. "Overlapping hierarchical clustering (OHC)." Advances in Intelligent Data Analysis XVIII: 18th International Symposium on Intelligent Data Analysis, IDA 2020, Konstanz, Germany, April 27–29, 2020, Proceedings 18. Springer International Publishing, 2020.

[9] Rivera, C. G., Vakil, R., and Bader, J. S. "NeMo: network module identification in Cytoscape." BMC Bioinformatics 11 (2010): 1-9.

[10] Luo, F., Yang, Y., Chen, C. F., Chang, R., and Zhou, J. "Modular organization of protein interaction networks." Bioinformatics 23.2 (2007): 207-214.

Supplementary Materials

Machine Specifications:

- Processor: Intel(R) Xeon(R) Gold 6136 CPU @ 3.00GHz
- RAM: 250GB DDR4
- Operating System: Ubuntu 20.04 LTS

Software and Versions:

- Python: 3.10.13
- NumPy: 1.26.4
- Pandas: 2.2.1
- Networkx: 3.3
- Matplotlib: 3.8.4

Algorithms and Implementation

OH-PIN: The OH-PIN algorithm was implemented following the pseudocode and explanation provided in the original paper by Wang et al. It takes as input a graph formatted as an edge list, with edges separated by tabs in a '.txt' file compressed in '.gz' format. The algorithm, in a single run, outputs JSON files for different lambda values, ranging from the smallest lambda and lowest resolution to the largest. The lambda values are not input parameters in the script but can be modified within the script itself.

Girvan-Newman: We used an available implementation in Python, which can be found at https://github.com/kjahan/community. This implementation was modified to take as input the '.txt.gz' files, the standard format for STRING PPI networks. Additionally, instead of outputting just the best modularity cut as originally intended, our modification outputs several cuts at different numbers of clusters (Table 3). This approach leverages the hierarchical divisive nature of the algorithm, where the number of components decreases with the removal of edges. As the number of components surpasses the lowest value in the list of target clusters, that partition is outputted. Thus, a single run of the algorithm provides clustering for all the specified cuts.

Louvain: For the Louvain algorithm, we fine-tuned the resolution parameters (Table 2) by applying the algorithm with a wide range of values. We saved the clustering results that had a number of clusters similar to the target clusters outputted by OH-PIN. In this case, each run of the algorithm was performed for a different resolution parameter.

Reproducibility

All the script used, along with all the input data, output data, and a notebook to reproduce the analysis and figures are available on GitHub.

Further Results

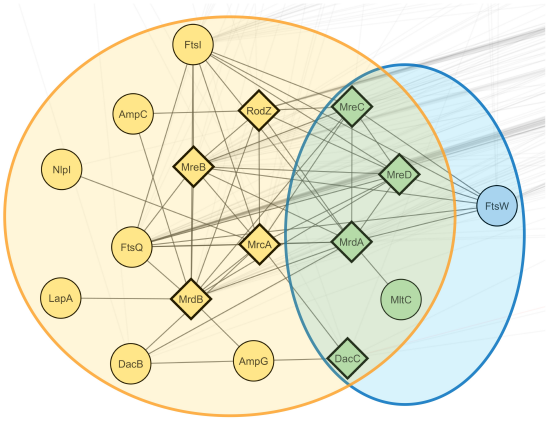

Figure 7: **Example of Clustering:** The Elongasome Complex at 1000 Clusters Resolution. The OH-PIN algorithm (yellow) fully recovers all known components of the Elongasome complex (square nodes), whereas the Louvain algorithm (blue) only partially captures these components. Green nodes indicate proteins identified by both algorithms. Additional nodes in the OH-PIN cluster may represent undiscovered complex components or functionally related proteins.

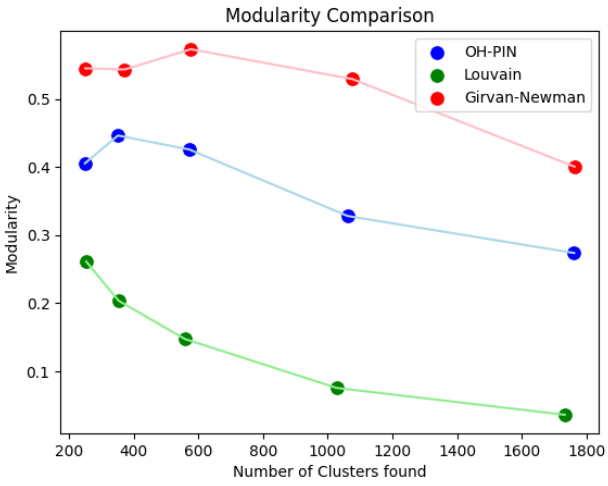

Figure 8: **Modularity** trends across varying cluster counts for OH-PIN, Louvain, and Girvan-Newman algorithms. This plot illustrates the modularity values achieved by each algorithm as a function of the number of clusters identified. Higher modularity values indicate better-defined community structures within the network. The Girvan-Newman algorithm consistently achieves the highest modularity, particularly at lower cluster counts, followed by OH-PIN, which maintains stable performance across different cluster counts. The Louvain algorithm shows lower modularity values overall, with a significant decline as the number of clusters increases.