Evaluating Similarity Network Construction in Biomedical Data: Implications for Community Detection Performance

Aidan Marnane

Doctor of Philosophy

THE UNIVERSITY OF EDINBURGH

2024

Abstract

Similarity network construction is a fundamental step in many approaches to community detection in biomedical analysis. It is used both in the creation of network structures from non-relational data and as a processing step in clustering pipelines. The foundation of any network analysis hinges on the quality of the underlying network. With the rising popularity of network learning and network-based clustering, the importance of correctly constructing these networks is vital. However, the implications of key choices in similarity network construction —— specifically in sparsification methods and multi-modal integration —— remain poorly explored.

Similarity network construction involves several critical stages: computing pairwise similarities using an appropriate metric, sparsifying these similarities to define edges, and, in the case of multi-modal data, integrating the modalities. This thesis evaluates two key components within this pipeline —— similarity sparsification and multi-modal integration — by measuring their impact on community detection performance in the final network. To this end, I developed a flexible network generation framework and used it to create a suite of simulated datasets with known embedded cluster structures. These networks, with ground-truth communities, were evaluated using a novel analytic framework focused on the community detection performance of diverse clustering algorithms —— Stochastic Block Modelling, Leiden clustering, and Spectral clustering. A comprehensive set of metrics, including ground-truth cluster modularity, Adjusted Rand Index (ARI), Adjusted Mutual Information (AMI), and network statistics such as density, were employed to evaluate the quality of the constructed networks.

Firstly, I assess the quality of single-modality networks generated using common sparsification methods by evaluating the community detection performance of the clustering algorithms. The key sparsification approaches studied include K-Nearest Neighbour and ε-Thresholding. The analysis reveals a critical limitation of ε-Thresholding, which fails to account for variations in cluster density, resulting in networks of poor quality.

The thesis then extends the analysis to evaluate the effectiveness of popular multi-modal similarity integration techniques, such as Similarity Network Fusion (SNF) and Neighborhood-based Multi-Omics clustering (NEMO), across various multi-modal data scenarios. By applying transformations to ground-truth clusters, a range of modalities with differing embedded cluster information and noise levels were generated to stress-test the integration techniques. These scenarios included adjustments such as merging ground-truth clusters, increasing the presence of outliers and noise, and adding uninformative modalities. Notably, SNF and NEMO fail to outperform simpler techniques, such as mean similarity aggregation, when incorporating

modalities with inconsistently embedded clusters. I demonstrate how integration methods can be used to incorporate partial modalities — datasets where not all individuals have a full set of measurements in all modalities. SNF shows significant sensitivity to incomplete modalities while NEMO and mean aggregation are more resilient.

Finally, I validate the findings of our synthetic data scenarios using two biomedical datasets; one for discerning cancer subtypes using data from The Cancer Genome Atlas (TCGA) and the second for differentiating individuals with Autism Spectrum Disorder (ASD) using data from the Simons Simplex Collection (SSC). Both datasets exemplify common challenges encountered with biomedical data; high dimensionality, unbalanced class membership and partial modalities.

Lay Summary

This thesis explores how we can create networks to study biomedical data and understand connections between different elements. Creating these networks is like building a map of relationships, and is crucial for accurate analysis in biomedical research. To make a network we have two key steps, measuring how similar individuals are and selecting which individuals to connect together.

In this thesis, I develop a method to test different ways of selecting connections and study their impact on the how well we can identify groups (communities) within data. One of the common ways of doing this is to pick a cutoff value where we connect people above this value and do not if it is below. I show this approach has serious limitations which may lead to less accurate results.

The study also looks at situations where we have many different types of data available from an entity, a general example would be having both image and text describing an object. Combining and making use of different data sources (known as multi-modal data) is a critical challenge in biomedical research. For example, a person's biological information can be measured by many different components including genes and proteins. A number of approaches have been developed to make use of this data but they have not been compared in a structured manner. Interestingly, I show advanced techniques do not always perform better than simpler methods in handling these challenging datasets.

To confirm these findings, I apply these methods to real-world datasets related to cancer and autism. These datasets reflect common issues in biomedical research, like incomplete information and high complexity. I confirm my findings and introduce an method to understand and compare the communities identified.

In summary, the thesis provides insights into improving the methods scientists use to build networks for studying biomedical data, ensuring more accurate and reliable results in identifying patterns and relationships.

Acknowledgements

Firstly, I would like to express my immense gratitude to my supervisor Ian. While his guidance and advice throughout this project was invaluable, it was his kindness, compassion and unwavering support that allowed me to complete this journey. This PhD took many twists and turns, both within the research and outside it. From lockdown to unexpected illness, Ian's belief and patience in me never faltered.

Thank you to Antoine, Barry and Magda for creating a research group filled not only with brilliant ideas but fantastic friendship. To the wider Bioinformatics group, thank you for the weekly meetings that introduced me to such varied topics and were such a comforting constant during those lockdown months.

To all my fellow Data Science CDT cohort (& Co), thank you for the shared knowledge, the thoughtful discussions, the delightfully distracting lunch breaks and, most importantly, the friendship.

I would like to thank my family for their love and support. They provided invaluable advice and encouragement every step of the way.

Lastly, but most vitally, I must express my gratitude and love to my fiancée Lorna. Her belief in me was the driving force behind every achievement, and her support sustained me through every challenge. Without her, I would never have started. Without her, I would never have finished.

This work was supported in part by the EPSRC Centre for Doctoral Training in Data Science, funded by the UK Engineering and Physical Sciences Research Council (grant EP/L016427/1) and the University of Edinburgh.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

Aidan Marnane

Contents

Figures and Tables

Figures

Tables

Chapter 1

Introduction

A network is a collection of individuals or entities (nodes) joined by connections (edges) representing relationships or interactions. Networks are data structures used to represent and capture the complex relationships between a set of entities. Networks have been analysed and constructed across many domains and applications including energy networks, transportation networks, internet and communication networks, financial networks and citation and collaboration networks Barabási and Márton (2016); M. Newman (2018). Typical analysis can include studying dynamics within the network e.g. epidemiological spread of disease Keeling and Eames (2005), examining the robustness of networks to sudden changes or loss of connections Callaway, Newman, Strogatz, and Watts (2000) and identifying communities embedded within the structure Khan and Niazi (2017). Recently another field of study has emerged; network learning, which has exploded in popularity W. L. Hamilton, Ying, and Leskovec (2017). Machine learning techniques are used to leverage the relationships embedded within the network to make predictions on node, edge and graph properties Wu et al. (2021); Zhou et al. (2020). With the breadth of analysis available, naturally the question arises, how does one create a network in order to leverage these techniques?

In traditional networks, the relationships between entities have been inherent in the application. For example, energy networks can be defined by the flow of energy from power plant to home through physical wires Pagani and Aiello (2013). Internet networks can be defined by explicit weblinks between webpages Broder et al. (2000). Citation networks can be defined by the set of citations within each paper Radicchi, Fortunato, and Vespignani (2011). A relationship between entities is clearly defined within the context of the problem and is typically binary. A wire physically connects two houses or it doesn't, a hyperlink is included in a webpage or it is absent. The relationship is defined by the presence or absence of an explicit link between entities. Similarly, social networks can be defined by who interacts with who e.g. who follows who on Instagram Manikonda, Hu, and Kambhampati (2014). All these networks encapsulate a set of sparse relationships where the presence or absence of a relationship can be clearly identified. However, in many contexts a relationship between entities clearly exists but its representation is far less obvious.

Imagine we have a group of patients at a hospital, we can suppose that there is a relationship in the way their set of diseases/illnesses express themselves. Some individuals will have conditions very alike. Others will differ significantly, both in progression and expression of symptoms. We could create a network by adding edges between the most similar individuals. Yet this raises two questions, **How do we estimate the similarity between two entities and what level of similarity defines a relationship?** As another example, consider the construction of a social network based on interactions between a set of individuals. What defines a social relationship? Consider a typical group of individuals, regular interactions, such as in a workplace, clearly define a social connection but a once off interaction when purchasing a coffee in a cafe might want to be ignored. In a particular set of individuals, each individual will be more similar to some people and less similar to others. By counting the number and frequency of interactions between individuals, we obtain a method of estimating their similarity. However, we again encounter the question - **How similar do a pair of entities need to be in order to define a relationship and how do we identify this cutoff?**

Networks constructed from measurements of similarity are known as *similarity networks*. Similarity networks have been used extensively in biomedical applications. For example, the analysis of gene expression through expression networks Ruan, Dean, and Zhang (2010) and analysis of disease through patient networks Pai and Bader (2018). There are two fundamental steps in the construction of a similarity network; 1) the calculation of pairwise similarity between all entities and 2) the addition of edges to the network based on the value of their similarity[1]. In the context of community detection, a good similarity metric needs to assign high similarity between individuals within the same community and low similarly between individuals in different communities. While this estimation of similarity between individuals is a challenging task, the suitability of a particular metric is highly dependent on the particular application and the underlying data Dozmorov (2018); Huang, Luo, Li, Wu, and Wang (2021). In contrast, the process of constructing a network from a set of pairwise similarity measurements should be largely application independent provided an accurate and "good" similarity measure is selected. This edge selection process is vital. The key information contained within a network is its edge structure. The edges of a network define the communities, the flow of information and the dynamics within the network. It is the singular most important component of a network. As a result, the selection process of edges from similarity scores is essential to the network representation.

1. A pairwise similarity matrix can also be considered a weighted fully connected network where all nodes are to connected to all other nodes. (2) can also be considered a sparsification process where low similarity edges are removed.

What are the approaches to network construction from pairwise values? As discussed by Von Luxburg in von Luxburg (2007), perhaps the simplest approach to similarity network construction is thresholding. A threshold similarity value ε is selected and connections below this cutoff are removed while edges above the threshold are retained in the network. A highly popular technique, threshold similarity networks have been used to analyse gene expression networks Allen, Xie, Chen, Girard, and Xiao (2012); Langfelder and Horvath (2008), chemical networks Scalfani, Patel, and Fernandez (2022) and financial networks Nie and Song (2018). Equally popular is the process of K-Nearest Neighbour (KNN) network construction where the K most similar edges for each node are retained while all other connections are removed Pai and Bader (2018); S. Islam et al. (2021); B. Wang et al. (2014). These are the two most common approaches of transforming a pairwise similarity matrix into a sparse network, a process I refer to in this thesis as sparsification. Both methods have highly influential hyperparameters - choice of threshold ε and choice of K that heavily dictate the edge structure of the resulting network. Yet, the influence of sparsification choice and hyperparameter choice is rarely discussed. One of the few papers discussing the different approaches is von Luxburg (2007). Others have examined the effect of choice of threshold on network statistics such as clustering coefficient Zahoránszky-Kőhalmi, Bologa, and Oprea (2016) or compared the effect of hyperparameter selection on connected components in both threshold and KNN networks Ruan et al. (2010). The thesis seeks to investigate the effect of choice of sparsification on network structure but optimal network structure may depend on the particular task. For example node classification might have different structural requirements to community detection. I focus on the task of community detection. More specifically, **what is the optimal choice of sparsification method when performing community detection?**

Biomedical data is multifaceted. Datasets can be comprised of several modalities with vastly different numbers of features and varying distributions — each offering unique challenges. Examples include patient data comprised of health records and medical imaging Acosta, Falcone, Rajpurkar, and Topol (2022) and multi-omic data i.e. data from the genome, proteome, transcriptome, epigenome, metabolome, or microbiome Santiago-Rodriguez and Hollister (2021). For a particular set of entities, each modality offers a unique insight into their relationships. Disease subtyping and clustering are common analysis tasks. While the large number of features can offer challenges, network based approaches have shown success at unlocking the community structure within. A key step in constructing similarity networks from multi-modal data is integration. Should different modalities be processed independently or combined? Should separate networks be constructed or should pairwise similarity be combined before constructing a network? Complex methods such as similarity network fusion (SNF) B. Wang et al. (2014), which constructs a network by diffusing similarity across modalities, has shown success in cancer subtype detection. Yet the structure of networks resulting from its complex mechanisms are poorly understood. Simpler integration methods

such as NEighborhood based Multi-Omics clustering (NEMO) Rappoport and Shamir (2019) have shown similar clustering performance through use of a far simpler integration process (mean relative similarity of KNN networks). One study found that a simple mean similarity across modalities consistently outperformed SNF Mitra, Saha, and Hasanuzzaman (2020). In this this, I examine the effect of integration method on network structure. **How does choice of integration method affect community detection performance**? **Does SNF outperform simpler integration methods?**

Biomedical data is typically incomplete Molenberghs and Kenward (2007); Voillet, Besse, Liaubet, San Cristobal, and González (2016). While the effect of *item non-response* (values unavailable for an individual in a a particular feature or set of features) is well studied Arslanturk, Siadat, Ogunyemi, Killinger, and Diokno (2016); Rubin (2018); Stiglic, Kocbek, Fijacko, Sheikh, and Pajnkihar (2019); Wells, Chagin, Nowacki, and Kattan (2013), a unique challenge in multi-modal data is *unit non-response* (no values available for an individual in any features within a modality) and partially complete modalities. Individuals often have an incomplete set of measurements and the number of observations within each modality can vary. While understandable factors such as difficulties with funding or differences in data measurement tools across clinical locations contribute to the difficulty of data collection Hall, Kea, and Wang (2019); Piantadosi (2005); Santiago-Rodriguez and Hollister (2021), there are dangers *unit non-response* may be a result of factors related to the disease or feature of interest Nakagawa and Freckleton (2008). A common solution to partially complete modalities is removal of individuals absent from modalities or the removal of entire modalities with low observation counts from analysis. This can lead to significant data wastage. Furthermore, within each modality when we ignore samples, we reduce the coverage of our feature distributions and reduce the quality of our representations within each modality. The problem of partial data is well recognised within the field of multi-view learning J. Wen et al. (2022); S.-Y. Li, Jiang, and Zhou (2014). For example, NEMO integration was developed to tackle the integration of partial modalities Rappoport and Shamir (2019). Within these works, the absence of an individual's observations from modalities are typically assumed to occur at random. Yet a fundamental concept in the study of *item non-response* is that the process of missingness can depend both on observed and unobserved variables. Data entries can be missing completely at random (missing independent of observed and unobserved variables), missing at random (missing dependent on observed variables) and missing not at random (missing dependent on unobserved variables) Nabi, Bhattacharya, Shpitser, and Robins (2022). Similarly, *unit non response* can occur due to observed and unobserved variables. In this thesis, I explore the effect of partial data, both random and non-random, on network structure. **Can partially**

complete data be incorporated into network construction? What is the effect of non random partial data? How does partial data affect network structure and do certain methods incorporate partial data better than others?

The next sections give an overview of the key research topics in the construction of similarity networks from uni-modal and multi-modal data. I start by introducing some fundamental network analysis concepts before introducing the key research topics this thesis seeks to address; sparsification, multi-modal integration and partially complete data. I discuss approaches to unsupervised clustering, also referred to as community detection, as well as introducing the metrics and quality scores that will be used to evaluate the networks constructed. I finish by outlining the structure and main contributions of this thesis.

1.1 Networks

Networks are also commonly referred to as graphs and in general the two terms are used interchangeably. It is important to note that there is a subtle distinction between the two. A network is a real-world system with practical applications, while a graph is the mathematical abstraction with vertices and edges that is used to represent these relationships. The same graph can be used to represent several networks. For example, nodes can correspond to different real world entities such as authors in citation network or proteins in a protein interaction network.

A graph can be formally defined as $G = (V, E)$ where V represents a set of vertices (nodes) and E represents a set of edges (links) connecting pairs of vertices. Let N be the number of vertices $|V|$. Let m be the number of edges $|E|$. A graph can be fully specified by its adjacency matrix A, which is a $N \times N$ matrix where each element a_{ij} is defined by

$$a_{ij} = \begin{cases} 1 & \text{if an edge exists between node } i \text{ and node } j \\ 0 & \text{if no edge exists between node } i \text{ and node } j \end{cases} \tag{1.1}$$

In undirected networks, A is symmetric with $a_{ij} = a_{ji}$. The degree of a node is the count of the number of edges connected to it. In an undirected network, the degree of node i is given by

$$k_i = \sum_{j=1}^{N} a_{ij} = \sum_{j=1}^{N} a_{ji}. \tag{1.2}$$

The total number of edges in a network can be expressed as a sum of node degrees

$$m = \frac{1}{2} \sum_{i=1}^{N} k_i. \tag{1.3}$$

In most real world networks, the degree of their nodes are not homogeneous and instead are distributed across a range of values. A networks degree distribution is the probability p_k that a randomly selected node in the network has degree k. The degree distribution is the normalised histogram given by

$$p_k = \frac{N_k}{N} \tag{1.4}$$

where N_k is the number of nodes of degree k.

Two of the most studied degree distributions are Poisson and scale-free degree distributions. Randomly generated networks such as Erdos-Renyi networks where N nodes are generated and m edges are placed at random Erdös and Rényi (1959) have degree distributions well approximated by the Poisson distribution and is characterised by a bell-shaped curve. Real world networks such as social networks or the internet exhibit scale-free distributions where the majority of nodes have low degree but a select number of nodes have very high degree. Such networks are typically power-law distributions where the probability that a random node has degree k is

$$p_k \propto k^{-\gamma} \tag{1.5}$$

for some constant γ. High degree nodes in scale-free networks are typically referred to as hubs. These highly connected nodes are central to the flow of information in the network and critical to network robustness. Scale-free networks are highly robust to random node removal but targeted removal of central hubs cause a significant collapse in information flow and connectivity Barabási and Márton (2016); M. Newman (2018).

Degree assortativity coefficient r is the Pearson correlation coefficient of degree between pairs of linked nodes. It is a measure of the tendency of nodes to connect to others of a similar degree. Positive assortativity indicates nodes with high degrees connect to nodes of high degree while nodes of low degree connect to nodes of low degree. Conversely in networks with negative assortativity high degree nodes connect to low degree nodes. Protein interaction networks typically display positive assortativity while the internet web page network is an example of a negatively assortative network.

Several metrics can describe the structure of a network beyond its degree distribution. Path length is a common measure of describing the interconnectivity and flow of information in a network. The shortest path between two nodes i and j is the path travelled along the edges of the network with the fewest links between i and j. The diameter of a network is the longest shortest path between all pairs of nodes in the network. The average path length of a network is average shortest path between all pairs of nodes in the network.

The clustering coefficient measures the tendency of nodes to cluster together. The local clustering coefficient for a node i is given by

$$C_i = \frac{2m_i}{k_i(k_i-1)} \tag{1.6}$$

where k_i is the degree of node i and m_i is the number of links between the k_i neighbours of node i. It is a measure of the density of the neighbourhood around a node and the tendency of a node's neighbours to connect together. The average clustering coefficient C is measures the degree of clustering within the entire network and defined as

$$C = \frac{1}{N}\sum_{i=1}^{N} C_i \tag{1.7}$$

It is important to note that degree distributions do not fully specify the edge structure of network. Two networks with identical degree distributions can have vastly different underlying structure. A typical example would be a random network versus a network with embedded clusters or communities. The community network will have groups of nodes that are more densely connected internally to themselves with few connections to other external nodes in the network. In contrast, the random network has edges distributed randomly across all nodes. Random networks have lower clustering coefficients unlike the community networks. The average path length in a random network will be lower. Communities can increase the total average path length as there are fewer inter community edges but within a community the path lengths will be lower due to the high density.

In this dissertation, I focus on undirected unweighted networks. Directed networks have non-symmetric relationships between entities where an edge between nodes i and j does not automatically indicate a corresponding edge exists between nodes j and i ($a_{ij} \neq a_{ji}$). An example of a directed network is a network of the internet constructed from hyperlinks Broder et al. (2000). A blog post might contain a link to a large website such as *wikipedia.org* but *wikipedia.org* is unlikely to contain a link to a smaller website such as a blog post. Weighted networks are networks where edges have different weights associated to them ($a_{ij} = w_{ij}, w_{ij} \in \mathbb{R}$). Similarity networks can be easily represented as a weighted network simply by setting the edge weight equal to the pairwise similarity between nodes.

Another common type of network are heterogeneous networks. Heterogeneous networks are networks with multiple types of nodes and edges. Knowledge graphs e.g. Wikidata Vrandečić and Krötzsch (2014) are examples of heterogeneous networks where different sets of entities are represented in the same network. For example, a knowledge graph of a university might contain different node types such as *department*, *lecturer*, *student*, *course* and different relations such as *teaches*, *member of*, *enrolled in*. Another common form of heterogeneous network are ontologies such as the Gene Ontology Gene Ontology Consortium (2004) or Human Phenotype Ontology Köhler et al. (2014) which are directed networks with formal logical rules governing the relations between entities in a particular domain. I focus on homogeneous networks with a single node and edge type. Not only is analysis such as community detection simpler on homogeneous network, the majority of clustering techniques and similarity network applications focus on homogeneous unweighted undirected network representations.

1.2 Similarity Networks

Similarity networks facilitate the construction of a network object from non-relational data. Similarity networks allow the application of network analysis approaches such as community detection or link prediction to non traditional network data. The two key steps in similarity network construction are the calculation of pairwise similarity between all nodes and the construction of a sparse network by adding or removing edges based on this similarity value.

The estimation of similarity is a highly challenging task and the choice of metric is highly dependant both on the particular application and the type of relationships that we seek to represent in the network. For example, when constructing an airport network, one might initially connect the closest airports based on geographical distance e.g. Gatwick and Heathrow airports should have a relationship as both are airports based in London. A far more informative measurement of similarity would be measuring the number of flights or the number of passengers flying between two airports. While simple metrics such as Euclidean distance or Pearson correlation are typically used to calculate similarity Dai, Zhu, and Liu (2020); Kim et al. (2019); Langfelder and Horvath (2008), numerous application specific metrics have been developed. Examples include measurements of disease similarity based on shared gene ontology terms P. Ni et al. (2020), similarity of drugs based on molecular structure Huang et al. (2021) or the fold similarity of proteins Sun, Zou, Guan, and Jin (2006). Due to the high dependency on the particular application of focus, we are less interested in exploring the choice of metric used to calculate similarity and far more interested in the process of constructing a network from pairwise similarity values.

To construct a similarity network, we first compute pairwise similarity scores between all entities (nodes). Typically, similarity score functions are structured around a particular distance metric, which is then converted into a similarity score. For example, the Gaussian similarity kernel $s(x_i,x_j) = exp(-||x_i - x_j||^2/2\sigma)$ makes use of the euclidean distance $d(x_i,x_j) = ||x_i - x_j||^2$ between two nodes x_i and x_j with hyperparameter σ. As shown in Figure 1.1, the pairwise similarity matrix S, where $S_{ij} = s(x_i,x_j)$ for similarity function s, can be considered a weighted, fully connected network where each node is connected to all other nodes (weighted by their corresponding pairwise similarity score). The process of creating a network from S can be considered in two ways: sparsifying the fully connected network by removing uninformative or dissimilar connections (edges), or constructing the network by starting with the set of entities and adding only the most similar edges between nodes. In this work, the process of creating unweighted undirected networks from a pairwise distance/similarity matrix is referred to as ***sparsification***.

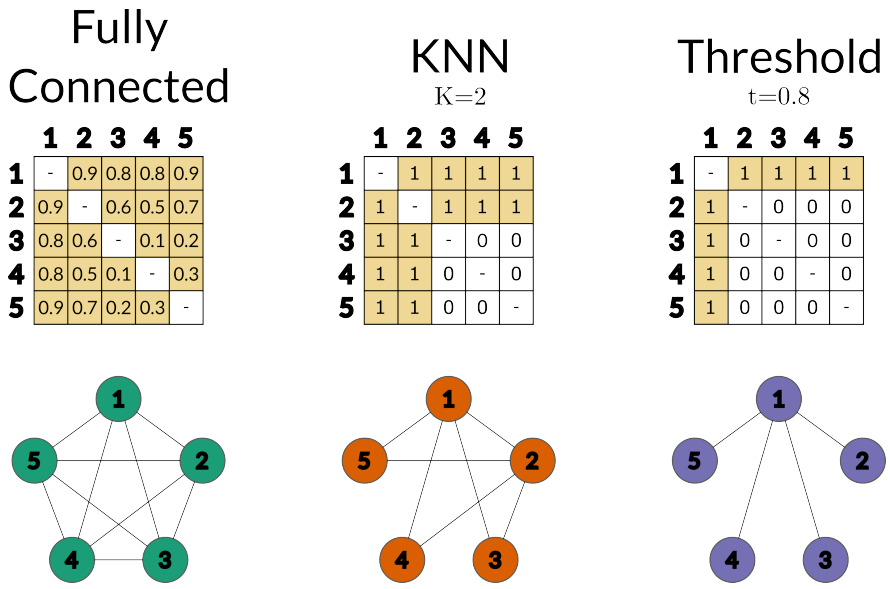

Figure 1.1: Illustration of the Sparsification of Pairwise Matrices. This figure demonstrates how pairwise similarity matrices can be represented as fully connected networks, where sparsification reduces the network to a sparse yet informative edge structure. The figure provides a simple example of two widely used sparsification methods: thresholding, where edges below a certain similarity threshold are removed, and K-Nearest Neighbour (KNN) selection, where each node retains connections only to its K most similar neighbours.

The motivations for the sparsification process are three-fold; firstly, naturally arising networks are typically sparse and the methods developed for network analysis were designed for sparse networks where the absence of a edge between individuals can be as informative as the presence of an edge. Secondly, many methods have computation complexity proportional to the number of edges $\mathcal{O}(|E|)$ in the network, computation over dense networks can quickly

become intractable, yet identical computations on sparse networks with a much larger number of nodes remain feasible. Thirdly, our similarity metric provides a ranking or an estimation of how similar individuals are. A fully connected network contains edges with weak evidence/justification and the sparsification process allows us to remove uninformative edges. The challenge of sparsification is the identification of these unlikely edges and the retention of edges that will be informative for our task.

As stated above, the process of sparsification converts our weighted similarity matrix to a unweighted undirected network G. The only criteria of interest in our similarity metric is the ranking or distribution of the pairwise distances between data points. The particular value assigned by a similarity function s is less important as this value will not be used once the unweighted network is constructed. For example, the Gaussian kernel $s(x_i,x_j) = exp(-||x_i - x_j||^2/2\sigma)$ and the euclidean distance $d(x_i,x_j) = ||x_i - x_j||^2$ can be considered equivalent for fixed σ as the order/ranking of edges calculated using both functions will be identical. The subsequent edges added to the network will be the same for both similarity functions. d is a distance/dissimilarity but can be converted to a similarity by simply reversing the order $s_d(x_i,x_j) = -d(x_i,x_j)$. In the Gaussian kernel, large euclidean distances are shrunk to 0, where as, similar objects with small distance values will be kept close to 1. σ is a key hyperparameter needs that needs tuning to identify the neighbourhood or set of values where the similarity should not be shrunk. In this work, such tuning is not required as we only consider the ordering or percentiles within the pairwise similarity not the specific value that is outputted.

As introduced earlier, thresholding and the selection of K-Nearest Neighbors (KNN) are among the most popular approaches to sparsification. While threshold networks are theoretically simple to construct, selecting an appropriate threshold ε in practice poses significant challenges. A threshold set too high can lead to disconnected components and isolated nodes, whereas a threshold set too low results in a highly dense network, potentially obscuring community structures and rendering the network uninformative Ruan et al. (2010). It is important to note that using a single global threshold to filter edges is an overly simplistic approach that fails to account for local variations in the data space, which can lead to the aforementioned issues.

To address these shortcomings, a more sophisticated approach would involve examining local density and adopting a dynamic threshold that accounts for local similarities between nodes. However, this raises further challenges: How should we determine an appropriate local threshold for each node? What criteria should guide the selection of ε_i for individual nodes?

K-Nearest Neighbours, a popular alternative to thresholding, offers a solution to these challenges by providing a mechanism for dynamic thresholding. In a KNN-based network, the threshold ε_i for each node i is implicitly defined as the similarity to its Kth nearest neighbor. Instead of analysing the entire distribution of similarity values and selecting a global cutoff ε, edges are retained based on local connections, with each node connected to its K most similar neighbours. This approach mitigates the risk of disconnected components and isolated nodes, which are less common in KNN networks.

Although choosing an appropriate K remains a challenge, it is typically selected to maintain low network density. One notable advantage of KNN networks over threshold networks is their scalability through the use of approximate KNN methods J. Chen, Fang, and Saad (2009). Despite the significant differences in network structure induced by these methods, their effects are rarely discussed. In Chapter 2, I will explore the impact of the choice of sparsification method on network structure, particularly focusing on how it influences community detection performance.

1.3 Multi-Modal Integration

There are number of different ways to approach the challenge of dealing with multi-modal/multi-omic data. The field of study is typically termed multi-view learning Y. Yang and H. Wang (2018). Multi-view learning encompasses a multitude of different approaches to tackling multi-modal data; dimensionality reduction Mitra et al. (2020), matrix factorisation Serra et al. (2015), deep learning approaches Zhao, Ding, and Fu (2017) and network based methods B. Wang et al. (2014). Multi-view/multi-modal datasets are datasets comprised of collections of modalities with distinct features, properties and statistics collected for each individual/entity in the dataset. Each modality captures different aspects of an individual/entity. A common example would be a dataset of animals containing both images and text descriptions. A biological example would be a multi-omic cancer dataset containing genomic and transcriptomic data. Common applications for multi-view learning include supervised prediction tasks (e.g. labelling of entities based on image and text) or unsupervised clustering (e.g. cancer subtyping).

In this thesis, I focus on network-based approaches to incorporating multi-modal data for community detection. My specific focus is on the construction of similarity networks from non-network data i.e. data where the relationships between objects are not explicitly defined. A prototypical example of a multi-view technique for similarity network construction is Similarity Network Fusion (SNF) B. Wang et al. (2014), a method of incorporating multi-omic data to

perform cancer subtyping. Network construction has the advantage over other multi-view techniques in that they output a data structure that can be easily adapted to perform several types of analysis Y. Yang and H. Wang (2018).

Multi-modal similarity integration methods for network construction can be broken into three categories as shown in Figure 1.2; early, intermediate and late integration. Let X_i be the data feature matrix, S_i the pairwise similarity matrix and G_i be the similarity network for each modality i.

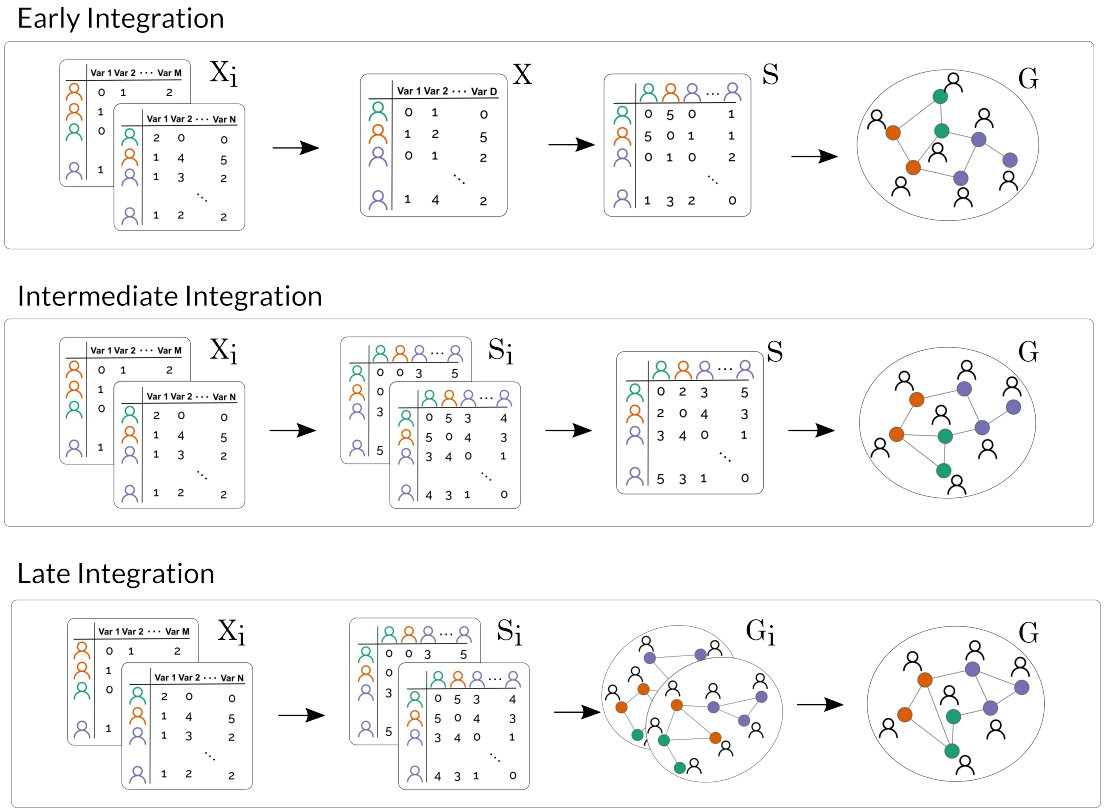

Figure 1.2: Approaches to Similarity Integration in Multi-Modal Network Construction. Methods can be classified as early, intermediate or late integration techniques where one of the modality's i) data features X_i, ii) pairwise similarities S_i or iii) individual networks G_i are integrated together in order to construct a similarity network G for the dataset.

Early integration combines the data features of the modalities. A typical approach to handle data from multiple sources is to simply merge or join the data for each entity into a single dataset and analyse a single data feature matrix. The preprocessing of the data prior to merging can be quite complex but for the purposes of network construction a typical early integration is to simply combine the individual modalities (after preprocessing). The pipeline is comprised of a single combined data feature matrix X, a single pairwise similarity matrix S

and final network G;

$$X_i -> X -> S -> G \tag{1.8}$$

Intermediate integration seeks to combine the pairwise similarities of each modality before constructing a network. For example in multi-omic data, it is often preferable to consider each modality individually. The number of features in each modality can vary significantly, with some modalities having an order of magnitude more features. When merging into a single feature matrix, the (potentially) distinct information contained in each modality can be dominated by the high number of features in certain modalities. An alternative approach is to compute similarity on each data source individually and integrate later. A benefit to such an approach is that different similarity measures can be applied to each modality individually. A common example of intermediate integration is to simply take an average of the pairwise similarity matrices $S = \frac{1}{M}\sum_{i=1}^{M} S_i$ (where M is the number of modalities) before constructing a final network.

$$X_i -> S_i -> S -> G \tag{1.9}$$

Late integration seeks to construct an network for each individual modality and combine the individual networks in some fashion. Similar to intermediate integration, late integration benefits from processing each modality individually. By creating a network for each modality, only the most important relationship and highly similar edges present in each modality are retained. The sparsification process required in the construction of networks helps distil the key relationships and pertinent information from each modality. An example of late integration is Similarity Network Fusion (SNF) B. Wang et al. (2014) which combines KNN networks from each modality in an non-linear process.

$$X_i -> S_i -> G_i -> G \tag{1.10}$$

Challenges in Evaluating Integration Methods

Late integration methods, such as SNF B. Wang et al. (2014), have seen extensive use in biomedical subtyping, including areas such as COVID-19 Ahern et al. (2022), Alzheimer's Tong, Gray, Gao, Chen, and Rueckert (2017) and paediatric brain tumours Cavalli et al. (2017). The original assessment of the method's performance was conducted on cancer data from the Cancer Genome Atlas (TCGA) Tomczak, Czerwińska, and Wiznerowicz (2015). Due to a lack of ground truth data, the validity of the method was verified by comparing survival curves and the number of significant genes within each cluster. Extensions to the method, such as NEighborhood based Multi-Omics clustering (NEMO) Rappoport and Shamir (2019), have also struggled with evaluation, and similarly relied on survival curve comparison and significant gene count.

One of the few studies that conducted a comparison between SNF and a simpler method, such as mean similarity (average pairwise similarity across modalities), using ground truth clusters found that the normalised mutual information (NMI) (see Eq. 1.5.2) performance of SNF was consistently worse than mean similarity across a number of datasets Mitra et al. (2020). It must be noted that this particular paper evaluated on clustering NMI performance on embeddings produced from each integration method and did not examine network quality or network based clustering. For example, instead of constructing a network from the mean pairwise similarity matrix, they produced a low dimensional embedding. Another concern was the choice of ground truth clusters for evaluation. While the ground truth subtypes of TCGA-BRCA and TCGA-GBM datasets are well verified, the use of cancer stage as a ground truth is questionable in datasets lacking accepted molecular subtypes. More concrete evidence is required to show that the relative increase in complexity of SNF compared to simpler produces a tangible benefit. A focused evaluation of integration methods on data with known ground truths is needed. Furthermore, to our knowledge no study has evaluated the effect of SNF's diffusion process on the underlying structure of the network produced.

1.4 Partial Data

The phenomenon of incomplete data is well studied in biomedical data. Missing data is commonplace Arslanturk et al. (2016); Molenberghs and Kenward (2007) and a plethora of strategies and imputation techniques have been developed to handle incomplete data Sterne et al. (2009); Wells et al. (2013), ranging from simple methods such as mean value imputation Graham (2009) to more complex methods such as multiple imputation Rubin (2018). The focus of research efforts has primarily been been on *item non-response* or missing values within features. While *unit non-response*, the absence of any features from a particular individual, is recognised as a problem, data was typically uni-modal and such individuals would not be included in any analysis. Care was needed to identify why an individual had no recorded values Hall et al. (2019) but without any observed values such individuals could not be considered for analysis. With the increasing availability of multi-modal datasets, *unit non-response* and methods to handle individuals with a partially complete set of measurements has increased in importance.

Typically, real world multi-modal data is only partially complete. Measurements are often missing for entities in one or more modalities Flores et al. (2023); Voillet et al. (2016). In this thesis, I refer to a dataset containing modalities with an unequal number of individuals as *partial data*. Whether to include partially complete data is a challenging question. Data analysis methods are typically developed with the expectation of complete data as an input and while imputation methods can ensure a feature's replaced missing values do not distort

analyses, the imputation of an entire individual's measurements is avoided in nearly all cases. As a result, when analysing multi-modal data, datasets are restricted to the subset of entities with a complete set of measurements in all modalities[2]. In general, the set of entities with complete measurements is a fraction of the total set of entities included in a dataset. An alternative approach is to restrict the set of modalities included in an analysis in order to maximise the number of entities that can be included. In either case, data wastage is common and large portions of a dataset, whose collection maybe have incurred a significant cost, is not utilised.

There are many possible reasons for unequal data collection across modalities Hall et al. (2019). There could be external factors such as budget constraints at the time of data collection (funding is only available to sequence X individuals). Partial data could be due to patient/individual time constraints (individual y can only attend 2 of the 3 evaluations). It could be due human error and issues in data recording. Another common restriction are location factors. For example, specialised equipment may be required to measure one modality and only a subset of the data recording locations have that equipment.

Partial data can also be due to the particular characteristics of a patient/individual. For example, only certain patients may be eligible for/undergo a particular test e.g. in an EHR dataset, a patient with a broken foot will not typically be sent for an echocardiogram (ECG) and, as a result, ECG data is unlikely to be available for patients without suspected heart issues. Moreover, unequal data can be by design, different tests are often applied based on age or severity of a condition. Many diagnostic tests for psychiatric conditions require a specific level of language ability and different tests are used to diagnose depending on the age of the individual Gotham, Risi, Pickles, and Lord (2007).

Another potential factor for unequal data collection may be socioeconomic. Availability for testing may depend on ability to take/obtain time absent from work or ability to afford childcare. Additionally the availability of specialised equipment may depend on the wealth of the neighbourhood/city/region/country. For disease analysis in particular, this possibility is highly significant as different racial/wealth groups may have different incident rates/subtypes of a particular disease and the proportion of individuals with partially complete data may be higher than wealthier individuals. This could lead to bias and imbalanced datasets Hall et al. (2019); Nakagawa and Freckleton (2008).

2. It should be noted that a complete set of measurements might still include many occurrences of *item non-response* and require traditional imputation techniques.

Many papers have explored the inclusion of partially incomplete multi-modal data from a methodological perspective S.-Y. Li et al. (2014); Xu et al. (2022), however, it is rare to see incomplete data included in formal disease analysis. If partially complete data is to be included in the analysis of a disease or condition, the reasons behind an individual's incompleteness are important to understand. Figure 1.3 provides an illustration of the types of partial data that can occur.

If the data is partial at random, then it is unlikely that restricting analysis to fully complete data will result in a biased dataset. On the other hand, including the partial data (assuming our methods do not suffer a significant drop in performance) should not greatly affect the conclusions of a clustering analysis but can increase the statistical power of a study. If the data is not missing at random there are two possible scenarios i) the factors dictating if an individuals data *is complete or not* are related to the clusters or subtypes within the data or ii) the factors are unrelated to the cluster distribution. For example, suppose we have two modalities and the only individuals missing from modality A are men and only individuals absent from modality B are women. In scenario i) each subtype is comprised of individuals of the same sex, in scenario ii) the subtypes will have a 50/50 split of sex.

In other words, i) the cluster distribution of data with partial information is significantly divergent from the complete set of data or ii) the cluster distribution is identical to the complete set. An analysis that restricts to the complete data in i) will add significant bias and fail to include critical individuals. An analysis that restricts to the complete data in ii) will not diverge from the true population.

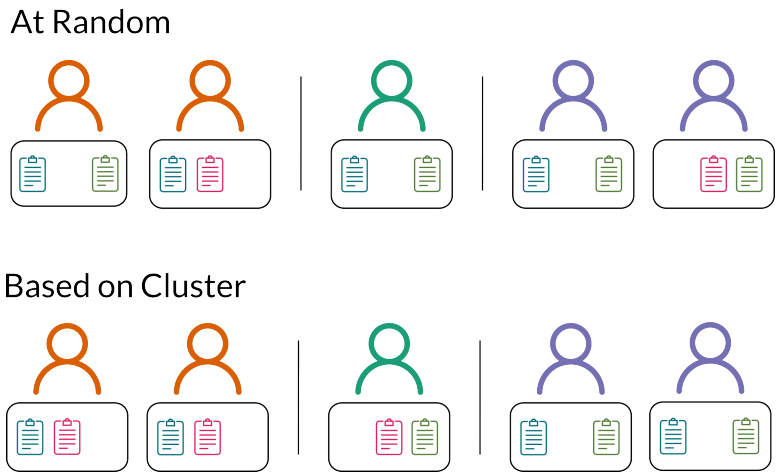

Figure 1.3: Types of Partial Data in Multi-Modal Datasets This figure illustrates two scenarios of partial data in multi-modal datasets: missing data either at random or based on cluster membership. When measurement are missing based on cluster, only individuals from cluster 1 (orange) do not have measurements in modality 3 (light green). In data partial at random, there is no link between the cluster label and the partial data.

In this thesis, I explore the effects of the including partial data on network structure. In Chapter 3, I evaluate the ability of integration methods to incorporate increasing levels of incomplete data on synthetic generated data with known ground truth labels. I assess data both partial at random and partial depending on cluster membership. In Chapter 4, I compare the clustering performance of partial and complete versions of real world datasets: multi-omic cancer tumours and a multi-modal dataset of individuals with autism spectrum disorder.

1.5 Clustering

In this work, I consider the creation of similarity networks with the aim of unsupervised detection of clusters within a dataset. In network science, this is commonly referred to as community detection. While the particular problem can vary from application to application, the most common setting in community detection is one where, *a priori*, both the number and size of the clusters are unknown. To reflect this, I require adaptable methods that do not require knowledge of the number of clusters but can detect both the number and size of each cluster.

In this thesis, by clustering I refer to disjoint or non fuzzy clustering where clusters do not overlap and a node can only belong to a single cluster. Our interest is further limited to focus on non-hierarchical methods. Hierarchical clusterings are rich in information but the evaluation and comparison of different hierarchical clusterings is challenging. Hierarchical clustering produces a tree representation. This tree is comprised of a number of levels where each level contains different partitions of the nodes. This tree representation is typically called a dendrogram. Dendrograms provide additional information over simple partitions — the similarity between different clusters can be measured using their distance in the tree. However, it is not simple to compare two different dendrograms. Measures typically used to asses hierarchical clusterings are the cophenetic correlation coefficient and the Fowlkes-Mallows index Fowlkes and Mallows (1983); Sokal and Rohlf (1962).

Cophenetic correlation is an internal measure that assesses the correlation between the dendrogram distance of the clusters and the mean dissimilarity between the nodes in the clusters. It does not facilitate external comparison to a ground truth dendrogram or direct comparison between two dendrograms. Indirect comparison can be still be performed by comparing the correlation coefficients of two dendrograms, yet this measure does not evaluate the agreement between both dendrograms. The Fowlkes-Mallows index does enable the comparison of two dendrograms. However, the index facilitates the selection of the fairest partitions to compare the two clusterings. It does not compare the overall agreement between the two hierarchical clusterings.

To calculate the Fowlkes-Mallows index, both dendrograms are cut to obtain k clusters for $k = 2,...,n-1$. For each k, the geometric mean of the precision and recall between the two partitions (B_k) is computed. The optimal k identified using B_k allows us to select cuts of the two dendrograms which has the highest agreement. Selecting the cuts to obtain k for both dendrograms in fair manner is difficult (cutting at each level in the dendrogram will not necessarily produce the correct sequence of number of clusters $k = 2,...,n-1$). Furthermore, while plotting B_k vs k can be visually informative, comparing the agreement of more than two dendrograms is not simple — what if each pairwise comparison identifies a different k as optimal?

Neither of the metrics introduced here facilitate direct comparison to a ground truth dendrogram. While comparison could be performed using "optimal" cuts of the dendrogram, i) selecting partitions for fair comparison is not trivial as discussed and ii) simply comparing partitions does not compare their hierarchical information. In this work, I want to assess sparsification methods for community detection. Without a clear method of evaluating dendrograms, hierarchical clustering methods are not suitable for this task.

Numerous approaches have been developed to perform unsupervised clustering. Non network based clustering methods such as K-means, Gaussian mixture models or DBSCAN are typical choices Ester, Kriegel, Sander, and Xu (1996); MacQueen (1967); Reynolds (2009). While in many applications these methods can be as accurate or even more accurate than network based methods Murugesan, Cho, and Tortora (2021), my aim in this work is to evaluate similarity network construction for community detection. Such methods do not accept networks as input and cannot be used to evaluate the quality of the networks produced by different sparsification methods.

A number of approaches have been developed for network based community detection including optimisation, spectral, probabilistic and dynamic based methods Fortunato and Hric (2016); Fortunato and Newman (2022); Lancichinetti and Fortunato (2009); Schaub, Delvenne, Rosvall, and Lambiotte (2017). Perhaps the most studied approaches are optimisation algorithms that aim to detect communities by optimising a criteria that measures how community-like sets of nodes are. One of the most popular criteria is modularity maximisation Clauset, Newman, and Moore (2004); M. E. J. Newman (2006a); Traag, Waltman, and van Eck (2019) but other methods use criteria such as edge betweeness to define communities M. E. J. Newman and Girvan (2004). The optimisation of such criteria in a network is a NP-hard problem and many different approaches have been proposed to identify approximate solutions using heuristics M. Chen, Kuzmin, and Szymanski (2014).

A highly popular set of algorithms use the dynamics of random walks on the network to identify communities. A key assumption in community detection is that clusters are more connected internally than externally. A random walk on the network starting in a cluster is more likely to visit nodes within a community than external nodes. The Walktrap algorithm uses short random walks to identify such patterns Pons and Latapy (2006). Infomap is another dynamics based method that assumes an infinitely long random walk on the network and uses the map equation to identify an information criterion based solution Rosvall and Bergstrom (2008).

Probabilistic approaches assume a probabilistic model to describe the community structure in the network and aim to infer community structure by fitting a generative model to the data (network). The Stochastic Block Model (SBM) is by far the most used generative model. There are several variants of the stochastic block model. The standard SBM introduced by Holland *et al* Holland, Laskey, and Leinhardt (1983) and its degree corrected variant proposed by Karrer and Newman Karrer and Newman (2011) both require the number of clusters or blocks to be known *a priori*. One of the most significant enhancements was proposed by Peixoto Peixoto (2019) with the microcanonical formulation that facilitates the selection of number of blocks (B) by applying the principle of minimum description length to infer the optimal choice of B.

Finally, spectral based methods identify communities by performing a spectral decomposition. There are several choices of matrices derived from the network; adjacency matrix, modularity matrix M. E. J. Newman (2006b), Laplacian matrix von Luxburg (2007) and Bethe-hessian Saade, Krzakala, and Zdeborová (2014). Following decomposition into the spectral space, a non network based clustering algorithm is used to identify the clusters. A typical choice is K-means clustering.

1.5.1 Clustering Methods

To evaluate our sparsified networks, a selection of different methods that take different approaches to community detection is required. A good similarity network should enable the discovery of the underlying communities for a variety of clustering approaches. To reiterate, in order to reflect real world scenarios, our community detection methods should identify the number of clusters as well as the particular node partitions. The methods used in this work are

- **SBM** — Minimum description length stochastic block model as implemented in `graph-tool`. We use a non-nested model with degree correction.
- **Leiden** — Leiden modularity maximisation algorithm. The resolution parameter is selected using event sampling and modularity maximisation.

- **Spectral** — Spectral decomposition of the random walk Laplacian $L_{rw} = I - D^{-1}A$ followed by K mean clustering using cosine similarity. The number of clusters are selected using eigengap ratio heuristic.

Stochastic Block Model

We use the degree corrected microcanonical SBM proposed by Peixoto in Peixoto (2019). A key assumption in probabilistic community detection is that the clusters or blocks define the generation process of the network. The aim in stochastic block modelling is to decompose the network into its "building blocks" where N nodes are partitioned into B blocks. As shown in Peixoto (2018), the microcanonical formulation of the SBM is given by

$$P(b|A) = \frac{P(A|b)P(b)}{P(A)} = \frac{P(A|e,b)P(e|b)P(b)}{P(A)} \tag{1.11}$$

where $A = \{A_{ij}\}$ is the adjacency matrix, b is the group membership vector of node i with $b_i \in \{1,\dots,B\}$ and $e = \{e_{rs}\}$ is the matrix of edge counts between groups. The difference in this microcanonical model to earlier formulations is the hard constraint that edge counts between groups are exactly e_{rs}. Previous formulations constrained the average number of edges between groups where fluctuations could occur between different samples.

A key advantage of this formulation is the ability to frame the posterior in accordance with information theory and automatically detect the number of clusters or blocks B. The description length Σ of the microcanonical model is given by

$$\Sigma = -\log_2 P(A,e,b) \tag{1.12}$$

$$= -\log_2 P(A|e,b) - \log_2 P(e,b). \tag{1.13}$$

Σ assesses the asymptotic amount of information required to encode data A together with model parameters e and b. The two terms can be thought of as i) the model evidence and ii) a complexity penalty. These two terms together help prevent overfitting and allow the identification of the number of blocks B. If the evidence for a particular block structure is high, the posterior probability $P(A|e,b)$ will increase. Any increase in model complexity due to increasing the number of blocks must be accompanied by a sufficient increase in the fit to the data. This penalty ensure the model does not become overly complex and overfit.

The microcanonical formulation can be extended to include a degree correction. Most networks have heterogeneous degree distributions and the degree corrected variant reduces the chance of trivial decomposition of our network into blocks of nodes with similar degrees. As Peixoto discusses in Peixoto (2019), degree corrected stochastic block models (DC-SBM) generally provide a better fit but do introduce an additional set of parameters which increase

the model complexity. The updated posterior is given by

$$P(A|b) = P(A|k,e,b)P(k|e,b)P(e|b) \tag{1.14}$$

where $k = \{k_i\} = \{\sum_j A_{ij}\}$ is the degree of node i. The model is fit using Markov Chain Monte Carlo where model selection is performed by selecting the model with the minimum description length. We make use of the python implementation in the `graph-tool`[3] package Peixoto (2014).

Leiden Modularity Maximisation

Modularity maximisation is one of the most widespread approaches to community detection on networks. For a given grouping of nodes into clusters C, modularity is a measure of the difference between the fraction of edges that exist within the groups compared to the fraction of edges that would be expected to exist under an appropriate null model. Modularity Q is given by

$$Q = \frac{1}{2m} \sum_{i,j} [A_{ij} - \gamma P_{ij}] \, \delta(C_i, C_j) \tag{1.15}$$

where A is the adjacency matrix, m is the number of edges in the network, γ is the resolution hyperparameter, C_i is the community node i belongs to, $\delta(C_i, C_j) = 1$ if nodes i and j belong to the same cluster and P is the expected adjacency matrix under a null model. A typical choice for null model is the configuration model where node degrees are assumed fixed and edges are placed at random, resulting in expected number of edges $P_{ij} = \frac{k_i k_j}{2*m}$ under the null model.

The modularity metric was introduced by Newman and Girvan in M. E. J. Newman and Girvan (2004) to measure the quality of the clusters identified by their edge betweeness clustering algorithm. The original definition of modularity was defined with γ=1. Methods to maximise modularity quickly emerged. For example, a greedy optimisation approach Clauset et al. (2004), a spectral approach using the leading eigenvectors of the modularity matrix M. E. J. Newman (2006b) and approaches adapted to larger networks that use multiple levels of scale of the network by merging communities Blondel, Guillaume, Lambiotte, and Lefebvre (2008) (the so-called Louvain method due to the location of its authors).

The resolution parameter was introduced as a solution to the resolution limit of modularity Fortunato and Barthélemy (2007). In larger networks, modularity optimisation has been shown to be unable to detect more than $\sqrt{2m}$ communities. This arises as a result of the number of edges expected under the null model. The expected number of edges between two nodes

3. v2.45

under the null model $\frac{k_i k_j}{2m}$ depends on the number of edges globally in the network m. Clusters are local phenomena in large networks but the null model implicitly assumes that each node can attach to any other node in the network. As the number of edges in a network increases, the expected number of edges between nodes and, more importantly, between small sets of nodes becomes vanishingly small. A single edge between clusters is seen as strong evidence that two sets of nodes form a single cluster. As the number of edges increase, the size of the set of nodes for which one single connecting edge constitutes strong evidence increases. As a result, clusters below a certain size will not be separated. The modularity of joining the sets of nodes together is higher than keeping the two separate Kumpula, Saramäki, Kaski, and Kertész (2007). To overcome this limitation, the resolution parameter was introduced. This allows the model to split such clusters by artificially increasing the number of edges expected between node sets under the null model.

The resolution parameter controls the number and size of the predicted clusters. A lower resolution parameter results in a fewer number of large clusters. A higher resolution parameter causes a larger number of smaller clusters. One challenge with the resolution parameter is an inability to handle clusters that vary significantly in size Peixoto (2021). Another challenge is the selection of an appropriate resolution hyperparameter. The optimal parameter changes from network to network and from application to application. The most popular approach to resolution parameter selection is a simple hyperparameter search for the parameter than results in a labelling with the highest computed modularity. There are difficulties, however, in selecting a representative set or range of parameter values for the different cluster scales.

The resolution parameter is not linear in terms of cluster scale. Qualitatively, the possible choices range from resolution parameter value which results in a single joined cluster containing all nodes to a choice that results in a set of singletons with each node classified as its own cluster. Yet, as shown in Jeub, Sporns, and Fortunato (2018), there is a very large range of values that result in a large number of singletons and this range of values varies from network to network. The qualitative properties do not change linearly with the resolution parameter. There is the danger that a selected set of possible sample values will not cover qualitatively different cluster ranges. Modularity maximisation across this range will result in a poor choice of gamma as different cluster scales will not be evaluated. One solution would be to evaluate a large sample of resolution scales. There are two disadvantages to such an approach; high computation complexity and a significant possibility of overfitting. Jeub et al. (2018) show an alternative process, which they term event sampling, that can be used to gain a set of parameter samples that are qualitatively distinct without increasing the number of gamma samples drastically.

The key component of their approach is the use of the relative fraction of node pairs with less edges than expected for any particular value of gamma as measure of cluster scale:

$$\beta(\gamma) = \frac{\sum_{(i,j)\in E^-(\gamma)} |A_{ij} - \gamma P_{ij}|}{\sum_{(i,j),i\neq j} |A_{ij} - \gamma P_{ij}|} \tag{1.16}$$

where $E^-(\gamma) = \{(i,j)|i \neq j, A_{ij} - \gamma P_{ij} < 0\}$ is the number of pairs of nodes with less edges than expected under the null model. $\beta(\gamma)$ is monotonically increasing for the γ values of interest and we can sample γ by inverting β. Equally spaced β values should produce a set of γ values that cover a representative set of qualitatively different cluster scales. They compare their approach to both linear and logarithmic γ sampling and show that event sampling produces a far more representative range of values. One issue with the approach, as proposed in their paper, is the number of possible events grows significantly in larger networks and the inversion calculation increases significantly in computational complexity. We use a simple subsampling process to estimate the beta curve in the inversion step rather than requiring a complete calculation. There is a loss of accuracy in the estimation of the beta curve in range of values where clusters split into singletons but, in general, the labellings produced by this range of hyperparameter values are of little interest.

Modularity maximisation has suffered criticism and a number of issues with the clusterings produced by maximisation methods have been identified. There is degeneracy in high modularity partitions. Partitions that are qualitatively quite different in terms of size and number of clusters have been shown to have similar modularity scores across a range of networks Good, de Montjoye, and Clauset (2010). Indeed, high scoring modularity partitions offer no guarantee of detection of true community structure. High scoring modularity partitions can be found in random networks generated using the Erdos-Renyi process which contains no implanted community structure Peixoto (2021). In the majority of networks, there is a plateau of high scoring partitions where multiple instances of the same algorithm on the same network return very different results Good et al. (2010). Coupled with the resolution limit, there are significant concerns on the quality of modularity maximisation clustering. In spite of such concerns, modularity maximisation has been shown to outperform other methods Z. Yang, Algesheimer, and Tessone (2016) and have been used to great effect in a variety of applications.

In this work, we make use of the Leiden algorithm Traag et al. (2019), it improves upon the Louvain algorithm. The Louvain algorithm has a tendency to produce badly connected or disconnected communities. The Leiden algorithm guarantees locally well-connected partitions. It is very computationally efficient and can handle large networks. We make use of the python implementation provided by the `igraph`[4] package Csardi and Nepusz (2006).

4. v0.10.3

Spectral Clustering

In spectral clustering, we perform a spectral decomposition of a matrix derived from the network to reduce the dimensionality of our data and then cluster in the lower dimensional space defined by a chosen number of eigenvectors ranked by their eigenvalues. The key parameters involved in spectral clustering are the choice of dimensionality k (number of eigenvectors to use) and the network affiliated matrix to be decomposed. Following decomposition and dimensionality reduction, any vector based clustering method can be used to identify the clusters but a common choice is K-Means clustering.

The key advantage of spectral clustering over other dimensionality reduction methods is the eigengap heuristic. The eigengap heuristic allows us to automatically identify the number of clusters (and eigenvectors) k. As described in von Luxburg (2007), if k clusters are well separated from one another then the ratio of the $k+1$th and kth eigenvalue will be larger than all other ratios. By selecting the largest eigengap, we should automatically detect the correct number of clusters K. There are several choices of commonly used matrices in spectral decomposition

- Pairwise Similarity (Affinity) matrix S
- Adjacency Matrix A
- Laplacian $L = D - A$
- Random Walk (left normalised) Laplacian $L_{rw} = I - D^{-1}A$
- Symmetric (normalised) Laplacian $L_{\text{sym}} = I - D^{-1/2}AD^{-1/2}$

In this work, we perform spectral clustering on the random walk normalised Laplacian L_{rw}. As discussed in von Luxburg (2007) the only difference between the two normalised Laplacians is a numerical factor in the eigenvectors. Yet this difference can lead to numerical artifacts in L_{sym}.

We use the eigengap ratio heuristic to select the number of dimensions and clusters K. In general this is quite effective, however, if the clusters are not well separated i.e. the clustering problem is quite noisy, then the eigengap heuristic is far less accurate Afzalan and Jazizadeh (2019). No eigenvalue ratio will be significantly larger than the others and the k corresponding to the true number of clusters may not be selected. To perform spectral clustering, we use the python implementation of spectral clustering provided by the package `spectralclusterer`[5] Q. Wang, Downey, Wan, Mansfield, and Moreno (2018).

5. v0.2.16

1.5.2 Clustering Evaluation

It is a difficult task to assess the performance of clustering algorithms. Particularly when, *a priori*, the number of clusters N_c is unknown. There are two possible errors that a good cluster evaluation score must account for; i) errors in the number of predicted clusters — is a prediction that underestimates N_c better than one that overestimates N_c? and ii) more intuitive errors such as assigning a node to the incorrect cluster. A challenge in this task is that cluster labels between the ground truth and predicted cluster labellings have no guarantee of matching. For example, cluster 1 in y may be labelled as cluster 2 in $\hat{y}$. As a result, a simple measure such as accuracy is not available for cluster evaluation. There are two approaches to overcoming these differences in cluster labels: i) compare all pairs of entities and count pairs that match in both cluster labellings and differ in both and ii) use an information theory approach by evaluating individual and joint entropies.

In this section, I will describe the most common measures of cluster evaluation, how they are computed and finally give an illustrative example showing how they evaluate in toy scenarios.

- **ARI** — Adjusted Rand Index,
- **AMI** — Adjusted Mutual Information,
- **V-Measure** — Harmonic mean of Homogeneity and Completeness of two clustering labels.

Adjusted Rand Index

The Rand Index (RI) was introduced in Rand (1971) and evaluates two labellings by comparing pairs of entities in both labellings. The (unadjusted) Rand Index for labellings y and $\hat{y}$ is given by

$$RI(y,\hat{y}) = \frac{a+d}{\binom{N}{2}} = \frac{a+d}{a+b+c+d} \tag{1.17}$$

where a, b, c and d denote the number of pairs in agreement or disagreement between the two labellings as outlined in Table 1.1. The Rand Index is proportional to the number of samples whose label agree in both or disagree in both y and $\hat{y}$.

The Rand Index accurately reflects the level of agreement between two labellings in scenarios with few classes. However, as the number of classes increase the number of elements who are in different classes in both y and $\hat{y}$ (d) will increase. Consider two completely random labellings, the random labelling with a larger number of classes will have a higher RI between it and the ground truth than random labellings with fewer classes. The baseline RI i.e. $RI(y,\hat{y})$ where $\hat{y}$ is a random labelling, changes based on the number of clusters in both y and $\hat{y}$. The increase results from the number of pairs in different clusters in both labellings d. The

Partition y	$\hat{y}$	
	Pair in same cluster	Pair in different cluster
Pair in same cluster	a	b
Pair in different cluster	c	d

Table 1.1: Contingency table for comparing pairs of nodes in partitions y and $\hat{y}$.

more clusters there are in both the labellings the higher the chance that two entities will be in different clusters in both labellings. To correct for this natural increase in RI and establish a comparable baseline score for all labellings i.e. score 0 for random labellings, the RI is corrected for chance. Typically a permutation model is assumed; the number and size of clusters are assumed fixed and labels are randomly assigned by shuffling elements between clusters. The ARI was introduced in Hubert and Arabie (1985) and is given by

$$ARI = \frac{RI - E(RI)}{\max RI - E(RI)} \tag{1.18}$$

$$= \frac{\binom{N}{2}(a+d) - [(a+b)(a+c)+(c+d)(b+d)]}{\binom{N}{2}^2 - [(a+b)(a+c)+(c+d)(b+d)]} \tag{1.19}$$

This form was first presented in Steinley (2004). The Adjusted Rand Index ranges from [-1, 1] but a random labelling scores 0 and the score does not increase when the number of clusters increase.

Adjusted Mutual Information

The adjusted mutual information (AMI) is an information theory based measure based on the joint and individual entropies of the cluster labellings. The mutual information (MI) measures the agreement between two labellings. A common scoring function used in literature is the normalised version (NMI) but more recently the adjusted mutual information has emerged due its to correct for chance Vinh, Epps, and Bailey (2009).

The entropy for a labelling y is given by

$$H(y) = -\sum_{r=1}^{n} p(y{=}r) \log p(y{=}r) \tag{1.20}$$

where n is the number of classes in y and $p(y{=}r) = |y{=}r|/N$ is probability that an element in y picked at random is class r.

The mutual information for labellings y and $\hat{y}$ is given by

$$MI(y,\hat{y}) = \sum_{r=1}^{n}\sum_{s=1}^{m} p(y{=}r,\hat{y}{=}s)\log\left(\frac{p(y{=}r,\hat{y}{=}s)}{p(y{=}r)p(\hat{y}{=}s)}\right)$$

where m is the number of classes in $\hat{y}$

As discussed above, typically the mutual information is normalised to facilitate comparison. It is given by

$$NMI(y,\hat{y}) = \frac{MI(y,\hat{y})}{f_{\text{mean}}(H(y),H(\hat{y}))}$$

where f_{mean} is a generalised mean function of the entropies of each labelling. Common choices are the arithmetic or geometric mean but different selections are made depending on the particular application. Similar to the rand index both mutual information and the normalised mutual information are not adjusted for chance and the values increase as the number of classes increase. Similar to ARI, a permutation model is assumed in the correction for chance. The AMI is given by

$$AMI = \frac{MI - E(MI)}{f_{\text{mean}}(H(y),H(\hat{y})) - E(MI)} \tag{1.21}$$

where $E(MI)$ is the expected mutual information assuming a hypergeometric distribution and permutation model (full derivation can be found in Vinh et al. (2009)). In this work, we use the arithmetic mean $f_{\text{mean}}(H(y),H(\hat{y})) = (H(y)+H(\hat{y}))/2$ but as described above other choices are commonly encountered depending on the application at hand.

V-Measure

The V-measure is a conditional entropy based cluster evaluation score Rosenberg and Hirschberg (2007). It is based around two metrics of desirable properties in any cluster labelling

- Homogeneity (H) — $h(y,\hat{y}) = 1 - \frac{H(y|\hat{y})}{H(y)}$ is 1 if all $\hat{y}$ clusters contain only data points which are members of a single y class.
- Completeness (C) — $h(y,\hat{y}) = 1 - \frac{H(\hat{y}|y)}{H(\hat{y})}$ is 1 if all members of any given y class are data points in the same cluster in $\hat{y}$.

The V-measure is the harmonic mean of the H and C. Typically, an additional parameter β allows the weighting of the V-measure towards homogeneity or completeness depending on which property is desired to constitute a "good" labelling.

$$V(y,\hat{y}) = \frac{(1-\beta)*homogeneity*completeness}{\beta*homogeneity+completeness} \tag{1.22}$$

For $\beta = 1$, the V-measure is identical to the normalised mutual information (normalised with the arithmetic average of two entropies). As described in Section 1.5.2, the normalised mutual information does not account for the natural increase in mutual information that occurs when more clusters are included in a labelling. As an example, assume y has 3 clusters. A labelling $\hat{y}$ containing one single cluster e.g $[1,1,...,1]$ has a completeness score of 1 and homogeneity 0. A labelling $\hat{y}$ containing N individual clusters $[1,2,..,N]$ has homogeneity 1. However, unlike the labelling with one single cluster, it does not have completeness 0 and so does not have V-measure 0. This is due to the increase in entropy that arises from the increased number of clusters.

As a result of this natural increase with increased number of clusters, the V-measure is a poorer choice for evaluation than adjusted measures such as AMI and ARI. The conditional entropies do still have use. The H and C quantities assist greatly with explainability. They provide qualitative insight when comparing two clustering with similar ARI. Differences in homogeneity or completeness help understand the behaviour in any particular pair of labellings.

Clustering Score properties

As discussed in Section 1.5.2, there are two possible errors that can arise in cluster evaluation; incorrect prediction of cluster labels and incorrect prediction of the number of clusters. The cluster scoring functions take very different approaches to evaluate labellings but it not clear how the different types of errors affect the different cluster scoring functions. To provide more intuition on how different scores are affected by such errors, we conduct a simple experiment. We generate a toy example of 150 nodes split into 3 equal clusters. Labellings with the two error types are generated by i) swapping cluster labels to obtain incorrect classification and ii) splitting clusters while keeping the nodes in each sub-cluster homogeneous i.e. all nodes in each sub-cluster come from the same cluster in the original labelling.

Figure 1.4 shows the effect of two error types on ARI, AMI, Homogeneity (H) and Completeness (C). Nodes are grouped by their predicted cluster in $\hat{y}$ and coloured by their original cluster in y. A *KNN* network ($K = 5$) has been generated for visualisation purposes. As we can see, ARI and AMI respond differently to different errors types. Figure 1.4A shows the effect of introducing 6%, 10%, 22% and 30% of incorrectly labelled nodes. We can see AMI, ARI, H and C all experience equivalent drops in performance. It is notable that 6% incorrectly labelled nodes result in a ARI of 0.83 compared to an accuracy of 94%.

There is a noticeable divergence in behaviour in Figure 1.4B. Figure 1.4B shows the effect of splitting clusters into 4, 5, 9 and 15 sub-clusters respectively. The ARI decreases far more rapidly than AMI. Homogeneous but incorrectly split clusters are scored far worse by ARI than AMI. This drop is likely due to the correction for chance in ARI. It punishes differences in the number of clusters far more than the entropy based measure. The AMI is likely not as affected due to the high homogeneity within the clusters.

Evaluating performance per cluster

In the case of clusters of different sizes, problems in cluster evaluation arise similar to those that occur when assessing unbalanced labels in supervised prediction problems. The majority of nodes are contained in the largest clusters. Suppose a method is very good at detecting large clusters but very poor at detecting smaller clusters. This method will consistently score highly when evaluated using the metrics proposed in Section 1.5.2. Conversely a method that accurately identifies the smaller clusters but splits or fails to detect the larger clusters will consistently score lower using these metrics than methods that detects large clusters. In settings such as disease subtyping, it is common to encounter large homogeneous cohorts i.e. large clusters, mixed with smaller subtypes that are less studied and the targets of interest. A method that accurately identifies the known larger group is of less use within subtyping applications than methods that can accurately detect smaller subgroups.

How to obtain a measure invariant to class imbalances (differing cluster sizes)? One alternative method of assessing the quality of the clustering labels (when ground truth labels are known) is to identify the best matching predicted cluster for each ground truth cluster. The prediction of each ground cluster can then be considered to be independent binary prediction problems. Measures that account for class imbalances such as the F1-score or balanced accuracy can be used to evaluate how well each individual ground truth cluster is predicted. The mean score over all ground truth clusters provides a metric that accounts for the differences in cluster size. Algorithm 1 outlines the process of calculating a per cluster score s for a proposed set of clusters $\hat{y}$.

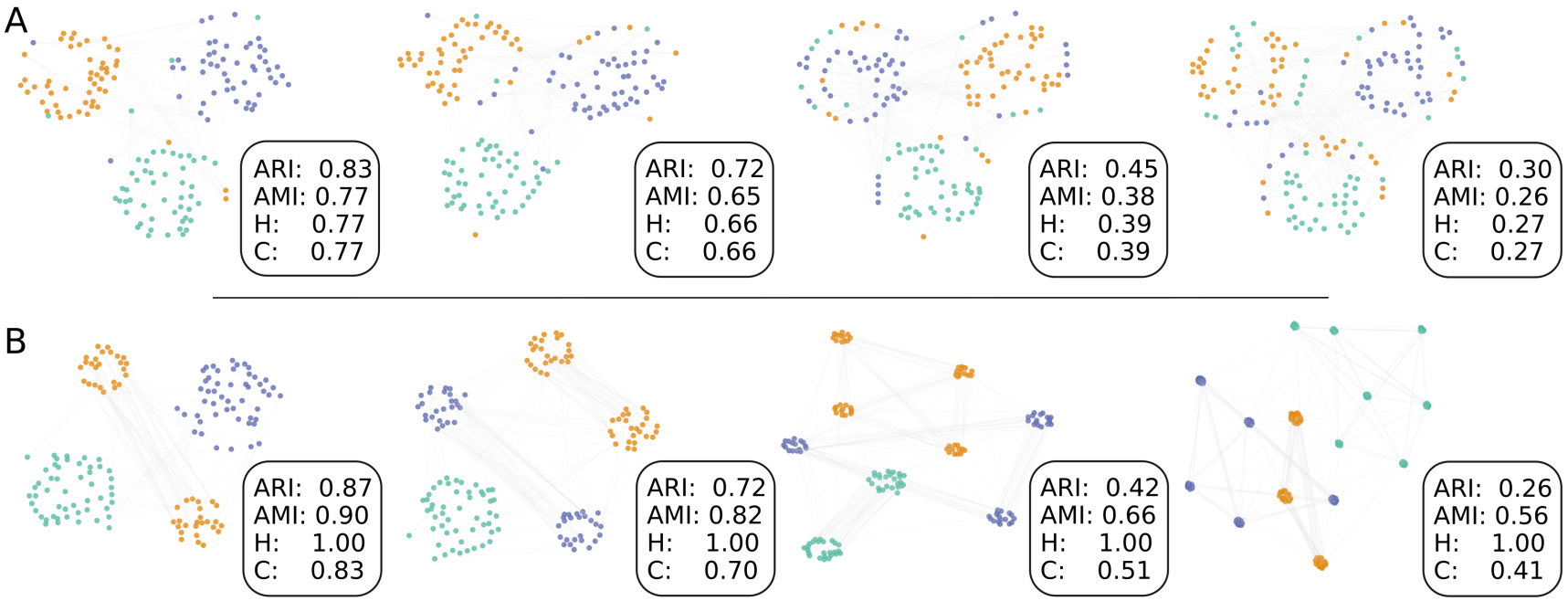

Figure 1.4: Impact of Mislabelled Data vs. Incorrect Number of Clusters on Clustering Scores. This figure examines the effects of mislabelled data and incorrect cluster numbers on clustering performance metrics. Each panel shows the clusters grouped by predicted cluster in $\hat{y}$ and coloured by true cluster in y alongside the ARI, AMI, Homogeneity and Completeness scores between $\hat{y}$ and y. Panel **A** shows the impact of mislabelling on clustering scores, where 6%, 10%, 22%, and 30% of nodes are incorrectly labelled. Panel **B** illustrates the effect of incorrect numbers of homogeneous clusters, where the original 3 clusters are split into 4, 5, 9, and 15 sub-clusters, each containing only the same y classes. We can see two distinct types of behaviour. The ARI and AMI decrease equivalently for mislabelled nodes. ARI decreases more than the true accuracy, for example ARI 0.83 for 94% correctly labelled nodes. There is a deviation in behaviour between the two scores in **B** when number of clusters is predicted incorrectly. The ARI decreases more rapidly and scores labellings with homogeneous but incorrectly split clusters worse than AMI.

Algorithm 1 Per Cluster Score for scoring function S

for y_i in **y do** ▷ Loop over labels in y
  $b_i \leftarrow$ CreateBinaryVector(y, l_i) ▷ Create binary vector based on target label l_i
  $\hat{y}_j \leftarrow \arg\max_j y_i \cap \hat{y}_j$ ▷ Find best matching cluster in $\hat{y}$
  $\hat{b}_j \leftarrow$ CreateBinaryVector$(\hat{y}, \hat{l}_j)$ ▷ Create binary vector based on label of best matching cluster $\hat{l}_j$
  $s_i \leftarrow S(\hat{b}_j, b_i)$
end for
$s \leftarrow \frac{1}{nc} \sum_{i=1}^{nc} s_i$ ▷ nc is total number of clusters in y

1.5.3 Cluster Quality

A central facet to all of the above scoring functions is the assumption that ground truth labels are available. However, a far more common scenario encountered when performing community detection is an unsupervised setting where true cluster labels are unavailable. The metrics proposed in Section 1.5.2 cannot be computed without ground truth labels. Alternative approaches are required. A number of different metrics or heuristics have been developed to assess the internal quality of proposed clusters without requiring comparison to external ground truth labels. Both network and non-network based metrics can be used. Examples of non network metrics commonly encountered are the Silhouette Coefficient Rousseeuw (1987) or Davies–Bouldin index Davies and Bouldin (1979) which make use of inter cluster distances in the feature space to evaluate the quality of a cluster. These methods rely on the accuracy of the selected distance metric and are not easily interpreted.

In this thesis, I am interested in network based cluster quality scores. As described in Section 1.5.1, a central assumption of community detection in networks is that the ground truth communities have higher intra-cluster connectivity than inter cluster connectivity. The majority of internal cluster metrics evaluate how distinct clusters (sets of nodes) are from the surrounding neighbourhoods in the network. A number of measures have been proposed and evaluated for their use in the identification of communities.

In J. Yang and Leskovec (2012), Yang and Leskovec conducted an extensive review of network cluster scoring functions that do not require knowledge of ground truth labels. They evaluate different internal cluster functions on an extensive set of naturally occurring networks with known ground truth communities. They define a set of community goodness metrics that measure the essence of what good communities are considered to be — well separated from the wider network, compact and internally well connected. Cluster scoring functions are ranked based on their community goodness metrics and how they evaluate perturbations in

the connectivity of ground truth communities. A subset of representative scoring functions are identified from correlations in cluster scores. Unlike Yang and Leskovec, we are not evaluating clusters on naturally occurring networks. We want to evaluate both the performance of clustering algorithms and the edge structure introduced by different similarity network construction processes. Both the quality of community goodness metrics and the cluster scoring functions of ground truth clusters and predicted clusters are likely to change. In this thesis, I make use of the three best performing cluster scoring functions; modularity, conductance and triad participation ratio (TPR) along with three key community goodness metrics; separability, density and clustering coefficient to evaluate cluster labellings.

The three cluster scoring functions $f(S)$ and three community goodness metrics $g(S)$ for a set of nodes S in an undirected graph $G(V,E)$ with $n = |V|$ nodes and $m = |E|$ edges are given by

- **Modularity** — as shown in (1.15) with $\gamma = 1$.

- **Conductance** — $f(S) = \frac{c_s}{2m_s+c_s}$ is the fraction of total edge volume that points outside the cluster. Lower is values imply a more community like set S.

- **Triad Participation Ratio** (TPR) — The TPR

$$f(S) = \frac{|\{u : u \in S, \{(v,w) : v,w \in S, (u,v) \in E, (u,w) \in E, (v,w) \in E\} \neq \emptyset\}|}{n_s}$$

  is the fraction of nodes in S that belong in a triad.

- **Separability** — $g(S) = \frac{m_s}{c_s}$ is the ratio of internal and external edges. It assesses how well separated a community is from the rest of the network.

- **Clustering Coefficient** (CC) — The CC for vertex v is

$$C_v = \frac{2|\{(u,w) : u,w \in N(v), u,w \in S, (u,w) \in E\}|}{k_v(k_v-1)}.$$

  It measures the number of links between vertices in a node's neighbourhood divided by the total possible links that could exist between them. The average CC for a set S is

$$g(S) = \frac{1}{n_S} \sum_{v \in S} C_v.$$

- **Density** — $g(S) = \frac{2m_s}{n_s(n_s-1)}$ is the ratio between the number of edges in the subgraph S and the maximum number of edges that could exist.

where $c_S = |\{(v,w) : v \in S, w \notin S\}|$ the number of edges from S to the rest of the graph G, n_S is the number of nodes in S, m_S is the number of edges in S, k_v is the degree of vertex v and $N(v) = \{w : w \in G, (v,w) \in E\}$ is the set of neighbours of vertex v.

1.5.4 Consensus

An alternative for assessing the internal quality of a clustering, especially in the absence of ground truth labels, involves examining the consistency and agreement across diverse clustering algorithms. Each clustering algorithm optimises for distinct community qualities within a network. Conversely, a network with a well-embedded community structure should be identifiable by all methods. Assuming that a set of clustering methods is equally accurate (which may not necessarily be true), a strong consensus should exist among the labels produced by these algorithms. A set of clustering algorithms with high consensus is less likely to contain random noise and be overfitted. We can be confident that the algorithms have detected the true embedded network structure.

Consensus can also be used to evaluate the quality of networks produced by sparsification methods. When comparing two network sparsification methods, the level of agreement between algorithms can serve as a decisive criterion. Strong agreement implies a clearly defined community structure and a lower signal-to-noise ratio in the network. A network construction method detectable by multiple clustering algorithms is more adaptable and valuable than one optimised for a specific approach.

To evaluate the agreement between algorithms, the methods proposed in Section 1.5.2 are all suitable. Adjusted Rand Index (ARI), Adjusted Mutual Information (AMI), and V-measure (Normalized Mutual Information, NMI) are symmetric and do not depend on either of the input labellings being true labellings. An exception exists for the measures of homogeneity and completeness. Each is the antithesis of the other: $H(y,\hat{y}) = C(\hat{y},y)$. A significant portion of their interpretability is lost when not comparing a predicted labelling to a ground truth. If one of the methods is fixed and considered the 'ground-truth' for purposes of comparison, the measures of completeness and homogeneity can still be used to compare the type of agreement between the two algorithms.

It must be noted that here we define clustering consensus as the agreement between cluster labellings produced by different algorithms applied to a single network. The term 'clustering consensus' is also used in the context of creating a consensus matrix or consensus network from a set of cluster labellings and performing a final clustering on the consensus matrix/entity

Lancichinetti and Fortunato (2012); Monti, Tamayo, Mesirov, and Golub (2003). The final labelling is often more robust to noise, more stable, and has been shown to be more accurate in certain scenarios. Indeed, this form of consensus clustering has been utilised extensively in single-cell RNA cell type detection Kiselev et al. (2017) as well as cancer subtype detection Brannon et al. (2010); C. Wang, Machiraju, and Huang (2014). In this work, it is of more interest to identify alternative signals for community detectability that do not require a ground truth rather than producing cluster predictions with higher stability.

1.6 Thesis Overview

The aim of this thesis is to evaluate similarity network construction in both uni-modal and multi-modal datasets, focusing on the impact of components of the similarity network construction pipeline on community detection performance. Specifically we seek to answer the following research questions:

- Defining Network Structure from Pairwise Similarity
  - What criteria or threshold should be used to determine when a similarity is strong enough to represent a relationship (or edge) in a network?
  - How does the choice of sparsification technique, such as thresholding or K-nearest neighbors (KNN), influence the structure of the resulting network?
  - Which sparsification method is optimal for constructing networks specifically for the task of community detection?
- Integrating Multi-Modal Data
  - How should different modalities (e.g., genomic, proteomic data) be integrated when constructing similarity networks?
  - Do complex multi-modal integration methods such as Similarity Network Fusion offer benefits over simpler approaches (e.g., mean similarity) in terms of community detection performance?
  - How does the choice of integration method affect the structure and quality of the resulting network?
- Incorporating Partially Complete Data
  - Can partially complete data (both random and non-random) be incorporated into the network construction process?
  - What is the effect of partial data, particularly non-random missingness, on the structure and quality of the network?
  - Do certain network construction methods perform better than others when dealing with incomplete data?

A key challenge in the evaluation of similarity network construction is the lack of datasets with embedded communities and ground truth labels. To facilitate accurate evaluation of the impact of network structure, I develop a data generation framework with a suite of various data distributions, cluster compositions, and relationships across modalities. The structure of the thesis is as follows:

- In Chapter 2, I evaluate common sparsification methods, such as threshold and k-nearest neighbour networks, on a representative set of network-based clustering algorithms. I use multiple instances of a set of generated data, comprised of representative cluster and data distributions, to measure clustering performance. I examine the sensitivity of sparsification methods to key hyperparameters in both performance and changes in network structure.
- In Chapter 3, I extend my assessment to similarity integration methods on multi-modal data. I evaluate a variety of early, intermediate, and late integration approaches, such as mean similarity, SNF, and NEMO. I explore different modality configurations to identify the strengths and weaknesses of various methods. Additionally, I examine the sensitivity of methods to an increasing number of modalities. The ability of methods to handle partially complete data is evaluated in two scenarios: when individuals are absent at random from a modality and when targeted clusters are absent in particular modalities.
- In Chapter 4, I reinforce my findings on two biomedical datasets. I explore cancer subtyping on multi-omic data from the Cancer Genome Atlas (TCGA) Tomczak et al. (2015) and differentiate individuals with Autism Spectrum Disorder (ASD) from non-ASD siblings within the Simons Simplex Collection (SSC) Fischbach and Lord (2010). Both datasets are representative of the challenges in biomedical multi-modal data with partially complete modalities, high variance in dimensionality across modalities, and unique distributions. I evaluate similarity integration methods on complete and partial versions of the TCGA and SSC data. Furthermore, I evaluate the predictability of the discovered clusters and illustrate factors critical to cluster membership.

I conclude with a discussion of the most important theoretical and empirical contributions of the thesis and point out directions for future research.

Chapter 2

Similarity Network Sparsification

2.1 Introduction

Network science and the analysis of network data has grown significantly in popularity in recent years. Two developments have been notable. Firstly, there has been a substantial increase in the availability and utilisation of biological, chemical, and biomedical data. These datasets often exhibit complex relational structures, making them well-suited for traditional network science tasks such as community detection and network robustness analysis. Secondly, there has been a breakthrough in the development of graph representational learning techniques, network embeddings, and graph neural networks (GNNs). These advancements have revolutionised the analysis of network data by enabling the application of machine learning algorithms to networks, demonstrating remarkable capabilities in tasks such as node classification, link prediction, and graph-level prediction W. Hamilton, Ying, and Leskovec (2017); M. M. Li, Huang, and Zitnik (2022). In many applications, data is inherently relational and a network representation occurs quite naturally; social networks, citation networks and protein-protein interaction networks. More commonly, however, the construction of a network representation from properties of the data is not obvious and network analysis methods are unavailable. Fortunately, there are ways of constructing networks in such settings and by far the most commonly used method of creating a network representation from non-relational data is through a similarity network.

A similarity network is a network where a set of individuals/entities (referred to as nodes or vertices) are connected to other nodes based on their shared pairwise similarity (calculated using some metric). Connections (known as edges) are included or excluded using some criteria; for example, a threshold value or desired number of neighbours. Similarity networks are constructed through a two step process. Firstly, the pairwise similarity between all nodes is calculated using a similarity measure. Secondly, a network is created by adding edges between nodes with the highest similarity using a particular selection strategy. The most common methods for selecting edges are using a threshold i.e. edges with a similarity value above the threshold are added or by creating a K-Nearest Neighbour (KNN) graph i.e. add the top K highest similarity edges for each node. It is important to note we can also consider similarity network construction through the lens of network sparsification where edges are

removed according to their pairwise similarity so only the highest similarity edges are retained. Many different approaches can be taken when estimating similarity or affinity between entities and are often domain specific, for example, calculating protein similarity based on their structural alignment Valavanis, Spyrou, and Nikita (2010) or estimating drug similarity based on their molecular structure or induced side effects Huang et al. (2021). The selection of metric/kernel/similarity function is essential to ensuring that the resulting network accurately captures relationships within the data.

Similarity networks have seen significant use across many biomedical applications. For example, the detection of cancer subtypes in multi-omic data B. Wang et al. (2014) and cell type discovery in Single Cell RNA seq data Hao et al. (2021); Kiselev, Andrews, and Hemberg (2019). They have also been used in to better understand disease pathways Y. Chen, Zhang, Zhang, and Xu (2015), novel subtype discovery with patient networks Pai et al. (2019) and in smartphone sensing and activity recognition applications Lane et al. (2014). While similarity networks are consistently created and utilised across a wide variety of applications, the process of constructing or sparsifying a network from pairwise similarity scores is not well understood von Luxburg (2007).

The effect of network construction approach is significant. Different approaches result in networks with significant divergences in network structure. In traditional networks, the structure is reflective of the data and the analysis of what type of network arises; scale free, small world, etc informs us about relationships within the data. By contrast with similarity networks, significantly divergent networks can arise from identical input data. Such similarity networks are still insightful but the choice of construction method is not inherent. A user must make decisions on the density, the number of edges and the process for adding edges.

Such challenges do not exist in naturally arising networks. For example, a citation network only has an edge if one paper cites another and there is very little ambiguity on whether such a citation exists. In similarity construction, an adjustment to the similarity threshold can add or remove an edge from the network. Von Luxburg discusses the most common approaches taken when constructing a network in von Luxburg (2007) but highlights that there are no guiding principles for the choice of graph hyperparameters when selecting how sparse a network should be. In Zahoránszky-Kőhalmi et al. (2016), Zahoránszky-Kőhalmi et al. show the effect of choice of threshold on a graph property — local clustering coefficient but to my knowledge no body of work has explored the effect of sparsification on common network problems such as community detection.

In this chapter, I demonstrate the effect of choice of sparsification method on community detection performance. A challenge in the evaluation of network construction is the lack of datasets with known ground truth community structure and consistent data properties. To overcome this, I use multiple instances of synthetic data to allow both a graph hyperparameter search and improved estimation of the effect of five common sparsification methods. I evaluate several network community detection methods across a range of different cluster settings. Section 2.2 describes the network sparsification methods evaluated. Section 2.3 describes the different synthetic data distributions generated and in Section 2.4 I discuss the experiments undertaken and the arising results.

2.2 Similarity Networks

In this chapter, I want to evaluate the quality of the network produced by different sparsification methods, specifically when applied to community detection. A key consideration in similarity network construction is the selection of metric and the calculation of similarity between individuals. A discussion on the importance of similarity metric selection can be found in Section 1.2. In the discussion of sparsification methods provided here, I am assuming for any application we can identify a similarity metric that ranks relationships between individuals accurately i.e. high similarity scores indicate a strong relationship between individuals that should be included in the network and low similarity scores indicate a weak relationship between individuals that should be excluded. Our focus is on evaluating the edge selection process — sparsification.

Different sparsification methods prioritise different types of connections. We want to evaluate a range of approaches. The two most popular approaches to sparsification are thresholding and K-Nearest Neighbour (KNN) sparsification. Both have seen extensive use across numerous applications. As discussed in Section 1.2, a key improvement on ε-thresholding is the use of adaptive and dynamic thresholds for each node i. KNN can be considered a method of identifying dynamic thresholds by setting each ε_i as the similarity to node i's K nearest neighbour. However, an ideal approach to dynamic thresholding would take in account the local density in a nodes neighbourhood. We propose two methods of selecting a dynamic threshold by adapting the K assigned to each node in KNN sparsification based on its local average distance using a linear map (Linear Skewed KNN) and logarithmic map (Log-Skewed KNN).

These different approaches result in networks with very different characteristics. My aim in this chapter is to evaluate and identify which characteristics are most beneficial in the specific context of community detection. I consider five sparsification methods

- **Threshold** — Select a percentile value *t*. Only pairwise similarity scores above this value are retained as edges.
- **K-Nearest Neighbour** — Select a number of neighbours *K*. For each entity the top K pairwise similarity scores are retained as edges.
- **Combined** — A combination of the prior two methods; Select *K* and *t* as above. Typically lower values for both are chosen.
- **Linear-Skewed KNN** — Select the max number of neighbours *K*. Scale the number of neighbours based on local density.
- **Log-Skewed KNN** — Similar to above. Select the max number of neighbours *K*. Scale the number of neighbours based on local density using a log scale (see Section 2.2.4).

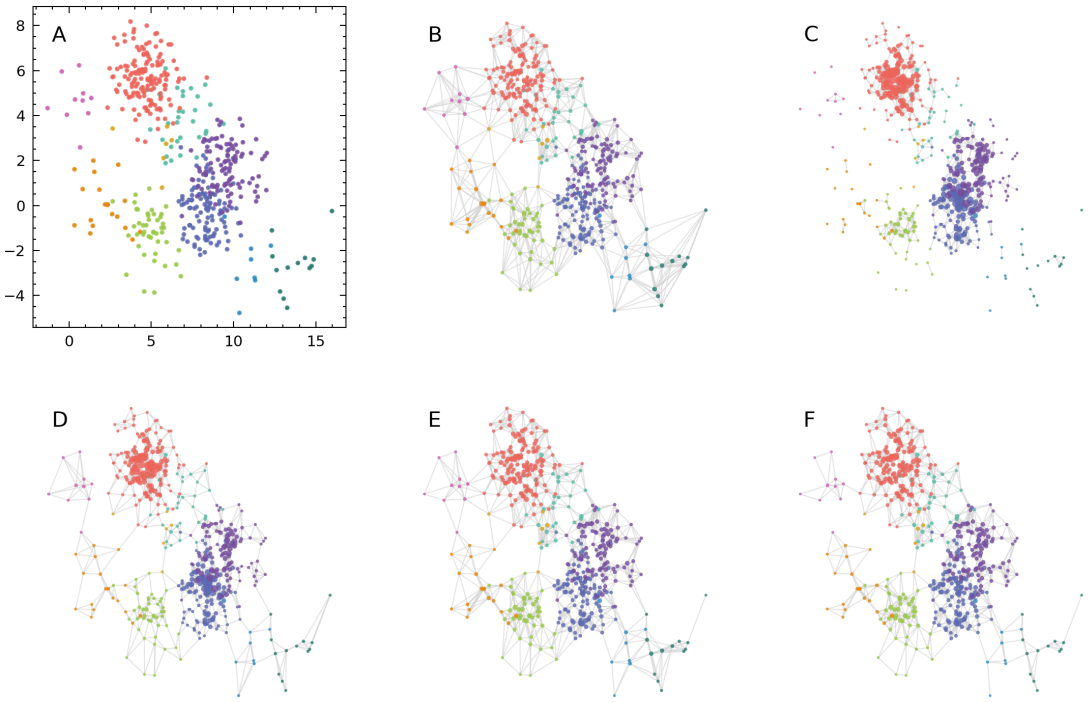

Figure 2.1: Example Sparsification Methods on a Two-Dimensional Mixture of Gaussians; **A**-Data, **B**-KNN, **C**-Threshold, **D**-Combined, **E**-Linear Skewed KNN, **F**-Log Skewed KNN. All networks have a density of 0.02, nodes are coloured by cluster membership and node size is scaled by node degree (number of edges). We can see in the Threshold networks (**C** & **D**) the large clusters are far denser. **C** highlights the issue of isolated nodes while **B** highlights the significant increase in edges less dense areas of the feature space receive using a KNN.

2.2.1 Threshold Network

Perhaps the most intuitive method for selecting edges to retain in the network is to make use of a cutoff and remove all edges below a certain similarity value. It is common to apply this criteria when a metric is interpretable. For example, a common filtration criteria is to remove edges below a Pearson correlation value (0.7 is a common cutoff). This approach is less suited to unnormalised and less interpretable metrics — a sensible threshold value for euclidean distance is far less obvious and will depend entirely on the dataset. A metric invariant approach is to consider the distribution of pairwise distance values in the data and retain only the top x% closest values as edges e.g. find the 99th percentile value and remove all edges with a similarity below this cutoff. While selecting a percentile can affect comparisons between different applications/datasets (the 99% correlation value could be 0.7 in one dataset and 0.8 in another) it provides an interpretable value for metrics like the euclidean distance that can vary significantly from one dataset to another.

The threshold approach to sparsification is a global evaluator. The entire distribution of pairwise distances is examined and only the closest connections within a dataset are retained. A key aim in similarity network construction is the removal of uninformative or dissimilar connections. Assuming our metric is well chosen, the computed similarity will correspond to our expected understanding of similarity within a particular application i.e. it will rank uninformative connections as far apart and informative connections as close together. For example in a patient network, a good similarity metric will always evaluate the similarity between two individuals with the same condition as "closer" than two individuals without. It should be noted that assuming the we have selected a good metric is a significant assumption. The quality or suitability of the metric depends on the distributions within the underlying data. If only features that do not differentiate between conditions are included, it not possible for similarity metric to compute higher similarity between individuals of the same condition.

An essential decision in network construction is the choice of metric but assume for the moment that our chosen metric is accurate and ranks similarity between nodes as expected for a particular application. An issue still arises when there are clusters of different densities within the data. Returning to our patient similarity example, suppose we have two conditions; i) A homogeneous condition with highly similar individuals (condition A) and ii) a more heterogeneous condition where the similarity between individuals with the heterogeneous condition is not high as the similarity between individuals with the homogeneous condition (condition B). Furthermore, assume as a group, individuals with condition B are consistently more similar to one another than to the individuals with condition A. The measured similarity within group B will be low compared to the similarity within group A. Yet the within group B similarity will be higher than the similarity between group A and B. If a global ranking approach

such as thresholding is applied, there are a number of cutoffs for which no connections within group B (the heterogeneous group) will be included as a result of its lower density. Group A (the homogeneous condition) will form a more dense, highly connected group with high similarity values between individuals. While this is an extreme example, the problem of including clusters with differing densities in the same graph is commonly encountered.

Figure 2.1 shows an example of a Threshold network. In Figure 2.1C, there are several dense clusters surrounded by isolated nodes. Many of the nodes in the less dense portions of the data have one or even no edges. These isolated nodes introduce a significant problem — there is no information in the network on how they relate to other nodes. In this two dimensional example, it is clear visually which clusters the isolated nodes are closest to but any network based clustering method will not have access to this information following sparsification and most methods will struggle to correctly classify these nodes. In order to add connections to these outlying nodes and include them in a larger connected component, the selected threshold would have to be increased and edges with weaker justification would be included in the network. It may be that the increase in density required to include isolated nodes will increase the level of noise in the network and render the previously detectable clusters indistinguishable from one another before the outlying nodes are included in the largest component of the network.

2.2.2 K-Nearest Neighbour Network

In contrast to the global threshold approach, K-Nearest Neighbour (KNN) networks can be considered a local approach to network construction. Knowledge of the entire pairwise similarity distribution is not needed to sparsify edges for a particular node. For each node, we select the top K most similar nodes and remove connections to all other nodes. As a result, each node will be of degree K at minimum. This guarantees areas of lower densities will have local information included in the network. Nodes are guaranteed K connections and so problems associated with threshold networks such as isolated nodes cannot occur. A common criteria used to detect communities in networks is cluster assortativity; connections within a clusters are more likely than connections outside a cluster. The KNN network guarantees that all nodes in a network receive a minimum number of connections. If true communities exist, the KNN should result in an assortative pattern. Figure 2.1B highlights the effect of the KNN approach. All nodes, even those in the less dense areas of the network, receive connections.

There are two possible problems that can still arise. We might under sample in dense regions of the data space where there are a large number of highly similar nodes (the node degree in the larger clusters is much smaller for Figure 2.1B vs 2.1C) or we might over sample less dense regions and include spurious or outlier connections that do not reflect the true community structure. The inclusion of connections to outlier nodes is of particular concern. Outliers are nodes that are truly remote and unlike all others, for example a misdiagnosed patient in a collection of patients with a particular condition. This individual might be very dissimilar to all other individuals in a cohort but a KNN network will guarantee it receives at least K connections. By contrast, in a threshold network we might identify this individual as an outlier (the node will have degree 0 and be dissimilar to all other nodes). The trade-off for a guarantee of at least K edges for all nodes is the inclusion of spurious edges or edges with weak evidence. An example of this type of outlier node is visible in Figure 2.1. The right most node in all panels is isolated and remote. It would be quite reasonable to label this node as an outlier but in the KNN network it is guaranteed K connections drawing it into the network. A potential benefit of including outliers is more complete coverage of the possible feature space. With the collection of more data, we might observe more entities in this outlier region of the feature space. One potential risk in similarity network construction is that only the most similar patients have edges in the network.

2.2.3 Combined Network

As discussed above, the threshold network and the K-nearest neighbour networks can be thought to encapsulate global and local information respectively. Assuming our metric measures similarity effectively enough (i.e. is optimal for the particular application) then these two sources of information should be complementary; the closest connections locally should accurately capture the community structure in less dense areas while the threshold network will prioritise the most similar connections in the network as a whole. Assuming a well defined metric, the global similarity should be prioritised as these connections correspond to the strongest connections in the network. In practice, howeverk, identifying a sufficiently accurate metric that accounts for difference in local density when scoring similarity is highly challenging. The inclusion of local similarity through the addition of KNN connections should help alleviate the downsides of the global information: isolated notes and poor retention of information in less dense areas. In this work, we term this inclusion of KNN and Threshold approaches a combined network. To create it, we use low values of K and high threshold t i.e. ($K < 20$ and the top 1-2.5% most similar edges).

2.2.4 Skewed K-Nearest Neighbour Network

One of the potential drawbacks of the KNN network is that it does not adapt the number of neighbours assigned to any node. It overvalues connections in less dense areas i.e. it includes connections that are too dissimilar. An alternative to assigning each node an identical K would be to skew the value or adjust the value K based on an estimation of it local density. This has two benefits; nodes with a large number of similar neighbours will not be under-represented and nodes that are in less dense areas that are unalike to other nodes will not be over represented in the network. The question then is how to estimate local density? Here we consider the distribution of the mean distance to a nodes' K_1 nearest neighbours. We select K_1 to be small to ensure we consider a small enough radius of connections but large enough that isolated nodes or communities have an increased value.

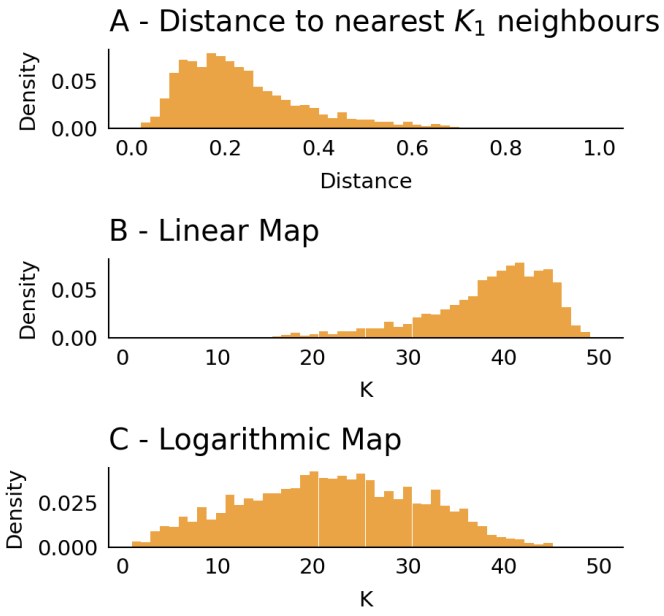

Figure 2.2: Mapping of Local Density Distribution to Number of Nearest Neighbours. This figure demonstrates how the number of nearest neighbours assigned to a node can be adapted based on local density in a dataset of mixed Gaussians. **A** shows the distribution of local density for all nodes estimated by the mean distance to their top K_1 = 10 nearest neighbours. For each node, we map from its local density to its assigned number of neighbours K. **B** & **C** show the distribution of neighbours K each node is assigned using a linear and logarithm map respectively from the local density to $[1, 2, .., K_{max} = 50]$. The Logarithmic map creates a larger number of nodes with low K.

Figure 2.2A shows an example distribution for a mixed multi Gaussian of ten clusters of mixed sizes. We consider two methods of mapping our density distribution to the distribution of neighbours $[1, 2, .., K]$; B) a linear map, and C) a logarithmic map. The linear map simply reflects the distribution of the density estimation; in this particular case a right skewed distribution with

a significant concentration of highly dense node i.e nodes with a small distance to its local neighbours. The logarithmic map maps significantly more nodes to lower values. Figure 2.3 highlights this effect. Panels 2.3A and 2.3B show nodes coloured by their assigned number of neighbours for the linear and logarithm map respectively. In the linear map, the majority of nodes are assigned K close to $K_{\max}$ (=10 in this example). By contrast, the logarithmic map assign significantly lower K values for nodes outside the dense clusters. This is reflected in the resulting networks. The Log Skewed KNN network (Figure 2.3D) has far fewer edges in the less dense areas of the feature space than the Linear Skewed KNN network (Figure 2.3C).

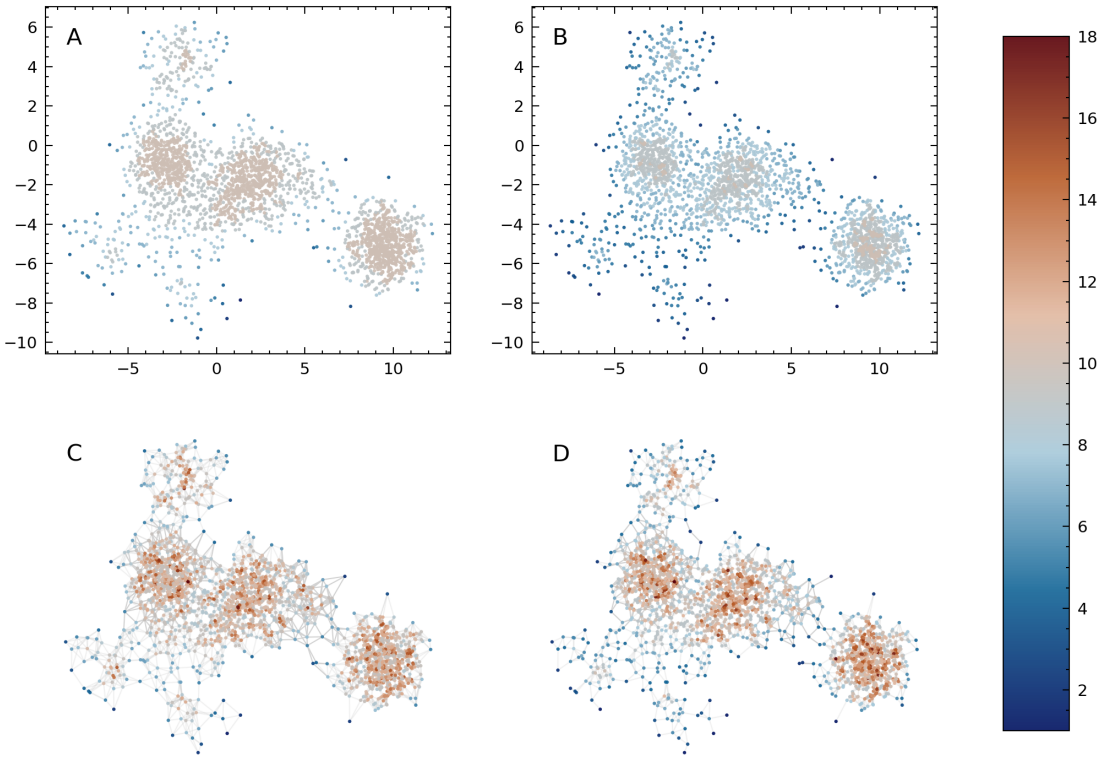

Figure 2.3: Creating a Nearest Neighbour Network with Adaptive Number of Neighbours. A & **B** show the number of neighbours assigned to each node using linear and logarithmic mapping respectively. **C** & **D** show the corresponding generated networks. In **A** & **B** points are coloured by their assigned number of neighbours K. In **C** & **D** nodes are coloured by their degree. The same colour gradient is used for all panels. The logarithm mapping assigns low density nodes lower K than the linear mapping greatly reducing the density at the peripheries of the network.

2.3 Synthetic Data Generation

As discussed above, one of the most common applications of similarity networks is community detection, in biomedical applications these include disease subtyping or single cell classification. They are frequently used as part of pipelines within detection methods for example, K-Nearest Neighbour networks are commonly used as a processing step within Laplacian based spectral clustering von Luxburg (2007) or as part of more involved analysis such as single cell clustering Hao et al. (2021). Again the effect of network parameter choices are not discussed extensively. One of the challenges with assessing the effect of parameter choices and construction methods is that ground truth labels are typically unavailable. Another challenge in evaluating similarity construction methods is that assumptions are made on the distribution of communities in the data space, more specifically, different communities have different distributions but these assumptions can only be verified through the accuracy or apparent success of methods after the construction of the network and the clustering method. Again in most applications, we will only have pointwise estimates of the model performance and assessing the impact of network choice is challenging. It is quite difficult to separate the effect of network choice from the choice of clustering algorithm with limited data instances.

In this work, I propose assessing network sparsification performance using synthetic data. Synthetic data is commonly used to evaluate network clustering algorithms. There are a number of benchmark networks with embedded community structure such as the Lancichinetti-Fortunato-Radicchi (LFR) Lancichinetti, Fortunato, and Radicchi (2008), the Girvan-Newman (GN) Girvan and Newman (2002) or the Artificial Benchmark for Community Detection with outliers (ABCD+o) Kamiński, Prałat, and Théberge (2023) benchmarks. These algorithms generate networks with realistic community structures, which are desirable. However, they produce fully formed, sparsified networks and lack the raw data and feature sets necessary for estimating similarity. Consequently, they are not suitable for evaluating sparsification methods.

To assess similarity networks, we require data generators that embed community structure in a set of features. We need data with two key characteristics — known ground truth labels and data where each cluster arises from a separate distribution. Synthetic data can provide both. With synthetic data, we can control both the size and number of clusters in our data. It is also possible to control the noise and difficulty of our cluster problems. Finally, we know with synthetic data that the different data instances arise from identical distributions. Changes in performance caused by different networks can be assessed with multiple instances, providing a more accurate measurement than single point-wise estimates.

We must acknowledge that there are significant drawbacks to synthetic data. We have no guarantee that these data settings are reflective of real world scenarios, in particular, the data scenarios where similarity networks are most typically used — gene expression data (RNA seq) and patient medical data. We have no guarantee that the noise profile and distribution is as challenging as real world scenarios. In developing this synthetic data framework, we make a number of decisions designed to reduce the simplicity and separability within the data and attempt to reflect more realistic real world scenarios. While the specific data distributions might not reflect real world distributions, the core assumptions behind each generation method do reflect assumptions commonly made when similarity networks are utilised.

We consider three different types of synthetic data distributions

- **Mixture of Gaussians** — each cluster assigned a separate cluster center and samples drawn from a d-dimensional Gaussian distribution with identity covariance,
- **Mixture of Student's-t distributions** — similar to the mixture of Gaussians but samples are drawn from a Student's-t distribution with 2 degrees of freedom,
- **Categorical Features** — d categorical features with n_l possible values. Samples from each cluster are drawn according to their own independent probability distribution.

Each of the three distributions assess very different characteristics. The mixture of Gaussian reflects an idealised setting where each cluster is quite distinct, each feature is informative and each data point is consistent with its particular cluster. The generation of mixture of Student's-ts is similar to the mixture of Gaussians but the samples for each cluster are drawn from a noisier distribution. The key difference introduced by this change in cluster distribution is lower density close to the center of the cluster and fatter tails. The higher spread ensures overlap and difficulty in distinguishing between different clusters. In contrast, the categorical data offers a significantly different problem setting. We have a mixture of informative and uninformative features. The low resolution of each feature is a challenge that both heightens similarities and differences between clusters. When the number of possible values within categorical features is fixed, the difficulty of the clustering problem increases significantly with the number of clusters for this data distribution. The categorical dataset additionally allows us to assess a setting where the euclidean distance metric is unlikely to be optimal in ranking the similarity/dissimilarity between different entities accurately.

2.3.1 Mixed Gaussian and Student's-t Distributions

Perhaps the simplest form of mixed cluster data, or at least the most commonly analysed cluster data, is mixed multi-dimensional Gaussian data. In this setting, points from each cluster are generated from separate Gaussian distributions. We consider the simplest setting: square Gaussians with unit standard deviation and identity covariance. In this setting, the only difference between each cluster's distribution is the center from which they are generated. A key challenge in creating realistic or challenging mixed Gaussian data is selecting the cluster centers. We want centers that are far enough apart that the clusters can be detected but close enough that the task is not trivial.

To allow flexibility generating arbitrary numbers of clusters, we require automatic generation of cluster centers. We generate the set of cluster centers sequentially. We start from a set of one initial point X_0. We then select a center at random from the existing cluster centers (initially just X_0). Proposal points are generated by sampling from a hypercube centered around X_0 of diameter U. Points too close (within a radius L) to all existing centers are rejected. Proposal points are generated until a center is accepted. We repeat until the required number of centers have been generated — randomly changing the existing cluster to center the hypercube on. Figure 2.4 shows an example of this process in two dimensions.

Biomedical data is typically high dimensional. Our generation process does not depend on the dimensionality of the Gaussian distribution. We can adapt this process to higher dimensional Gaussian clusters providing a more realistic set of data. For a fixed number of samples n, as the number of dimensions increases the likelihood of overlap between clusters decreases. In order to retain a challenging community detection problem, we adjust the diameter of the proposal region U and the radius of the rejection region around the cluster centers L relative to the dimensions of the multivariate Gaussians. Empirically, we found that scaling the rejection and sampling radius with $\frac{1}{\sqrt{d}}$ works well for $d <= 50$. Figure 2.5 illustrates how the clusters generated with this approach scale with dimensionality on a number of example cluster settings — — *Equal 3*, *Equal 10*, *Equal 30*, *Single Large* and *Mixed Sizes* (detailed description provided in Section 2.4.1). Figure 2.5A shows examples of data generated with two-dimensional multivariate Gaussians and Figure 2.5B shows examples of data from fifty-dimensional multivariate Gaussians projected to two dimensions using principal component analysis (PCA) Wold, Esbensen, and Geladi (1987). We can see in Figure 2.5B the clusters are not trivially separable in the two-dimensional PCA projection and that the properties of the data at lower dimensions are well retained as we scale the dimensionality.

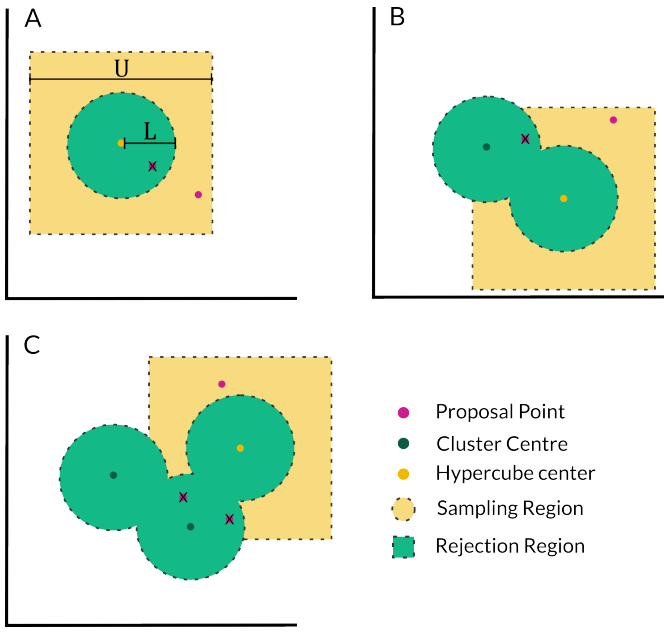

Figure 2.4: Process of Generating Cluster Centers for Mixed Gaussian and Student's-t Distributions. We generate cluster centers in a sequential manner. Panels **A**-**C** show a two-dimensional example of the iterative process. Two parameters control the behaviour of process; U the diameter of the possible sampling region and L the minimum radius around each center where we reject proposal points. By adjusting U and L, we control the level of overlap between clusters and the difficulty of the clustering problem.

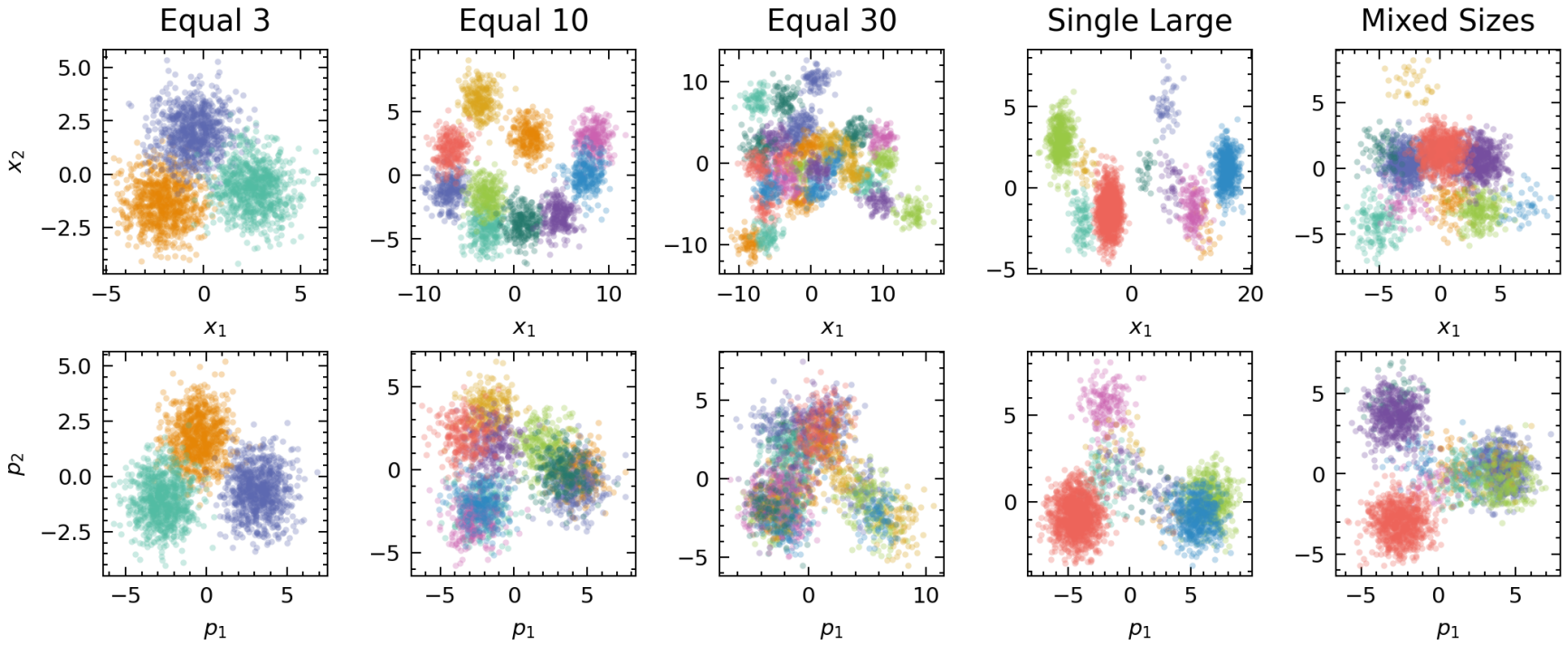

Figure 2.5: Cluster Properties in Mixed Gaussian Data with Increasing Dimensions. Data generated from **A** two-dimensional Gaussians and **B** fifty-dimensional Gaussians projected to two-dimensions using PCA. Five settings of different numbers and sizes of clusters are visualised — *Equal 3*, *Equal 10*, *Equal 30*, *Single Large* and *Mixed Sizes* (detailed description provided in Section 2.4.1). By scaling the diameter of the cluster center proposal region U and rejection radius L with $1/\sqrt{d}$, we ensure a similar level of overlap between the clusters and retain a challenging community detection problem.

Our procedure for placing cluster centers is designed to create datasets with dense clusters that exhibit overlap, ensuring a balance between cluster detectability and complexity. The placement algorithm positions cluster centers in a way that results in some samples being situated between clusters, leading to overlapping regions. This overlap creates a non-trivial clustering problem, where clusters are detectable but not easily separable. A more challenging community detection task can easily be obtained by replacing the distribution we draw our samples from. Using a Student's-t distribution with identity covariance and 2 degrees of freedom, we can add random noise to the data space and create a far more challenging detection task while keeping many of the properties of our mixed Gaussian data. The Student's-t distribution adds two key effects; significantly higher noise with greater overlap between clusters and a higher number of outliers with several points completely distinct from other members of the same cluster. Figure 2.6 shows the differences between Student's-t and Gaussian distributed clusters. Figure 2.6A shows example two-dimensional Gaussian data, 2.6B shows example Student's-t data generated with the same parameters and 2.6C contains the same data as 2.6B but limited to the area where the majority of samples lie. Figure 2.6B highlights the increased number of outlier points and Figure 2.6C shows increased spread and cluster overlap. Using Student's-t distributed clusters offer a noisier and more challenging cluster problem while retaining many of the characteristics of the mixed Gaussian distributed data.

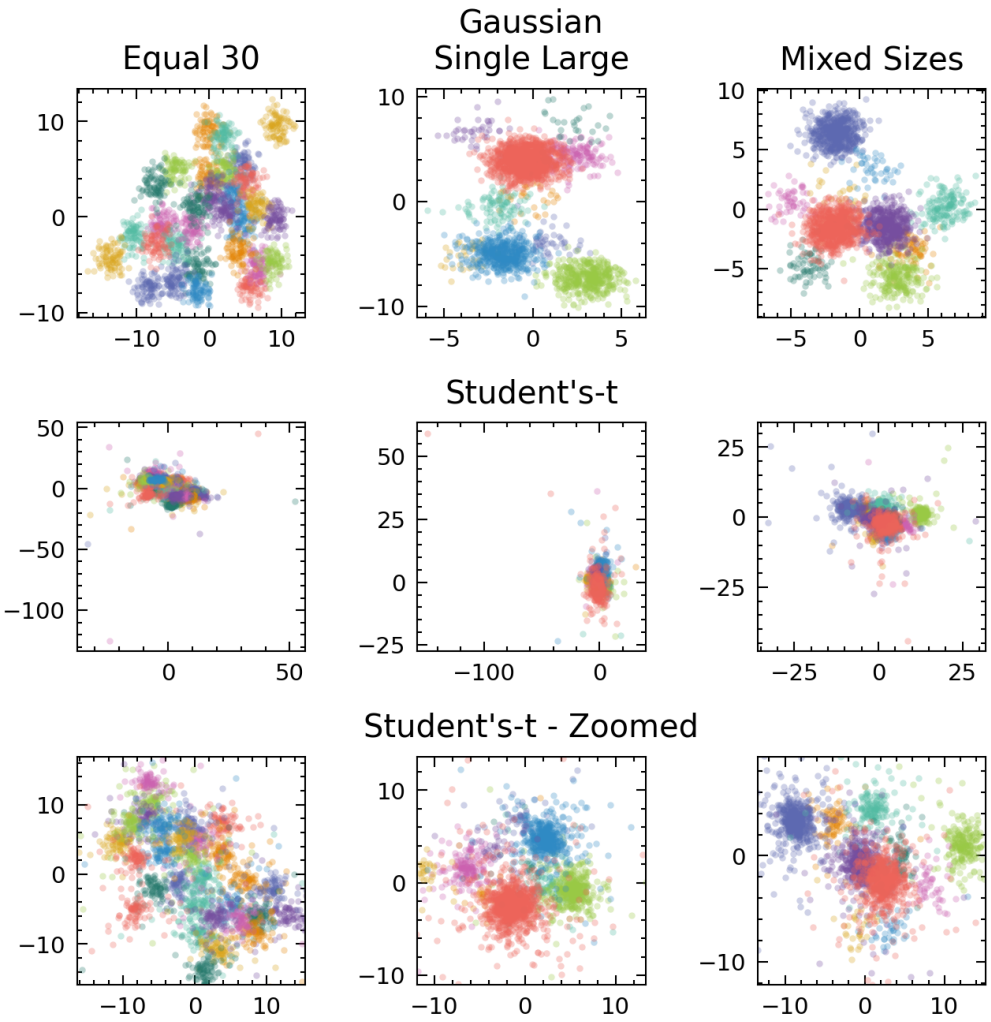

Figure 2.6: Comparison of Data Properties Between Gaussian and Student's-t Distributions. This figure demonstrates how mixed Student's-t distributions create more challenging clustering problems due to higher noise levels and the presence of outliers. We show examples of two-dimensional mixed cluster data generated with **A** Gaussian and **B** Student's-t distributions. The details on the size and number of clusters in the *Equal 30*, *Single Large* and *Mixed Sizes* data can be found in Section 2.4.1. **C** shows the Student's-t data restricted to the area where majority of samples lie. The heavier tail of the Student's-t introduces a significantly higher number of outlier points and increased overlap between clusters. This is a far more challenging clustering scenario that will evaluate the performance of different clustering and sparsification methods in a more noise intensive setting.

2.3.2 Categorical Data

Categorical and ordinal data are ubiquitous in medical settings. Medical questionnaires, and survey data in general, heavily utilise questions with scales of increasing severity/intensity to gather information about a patient or particular topic. Categorical data can simplify and streamline the set of possible responses as well as the subsequent statistical analysis. A careful component of survey design is the choice of closed i.e. categorical/ordinal questions vs open i.e. numerical/text based questions. Selecting open ended questions can introduce complicated data cleanup. As a simple example, suppose you ask at what age an individual started speaking; answers could be given in months (18 months), years and months (1 year 6 months) or even decimal years (1.5 years). Not to mention there may be several distinct ways of writing years/months (yr, yrs, mo., m, y, ms, ys) (assuming years/months are given in english). On the other hand, categorical data can introduce problems of their own such as lack of granularity, the merging of responses or limited ranges introducing problems such as the ceiling effect where a significant range of individuals are compressed into a single category. Disease analysis and the analysis of EHR data often encounter significant number of ordinal or categorical variables yet the majority of clustering and analysis techniques are designed for high dimensional numerical data.

In this work, we want to evaluate our sparsification methods in settings where the euclidean distance is unlikely to be optimal. The mixtures of Gaussians/Student's-t are constructed through a spatial based procedure. We want a data generation procedure that is built using assumptions typically made when analysing real-world categorical/ordinal data. As with numerical data, our core assumption is that each subgroup arises from its own unique distribution. For our categorical distribution, rather than generating individual clusters from multivariate distributions, we generate individual features where each clusters has its own probability mass function (PMF) over the possible data values. Within categorical data, there may often be significant overlap between observed values within each group e.g. if two groups are flipping a coin both groups will have an observed count of heads and a count of tails. However, our assumption is the generating distribution for our groups is different, imagine one group has a biased coin then the observed proportion of heads will not be equal to the tails.

To generate a probability mass function (PMF) across m categories, we sequentially sample m times from m uniform random variables to get a set of probabilities p_k where the limits on each random variable are adjusted so that $\sum_{k=1}^{m} p_k = 1$. Our PMF is then given by $[p_1, p_2, \ldots, p_m]$. We have two requirements — the total mass must equal 1 and all values must be non-negative. We add a parameter $xmin$ that allows us to ensure at least one category will have $xmin$ probability mass, $\exists k; p_k \geq xmin$. Our process proceeds as follows; $xmax_1 = 1$. i) if first element: $p_1 \sim U[xmin, xmax_1]$. else: $p_k \sim U[0, xmax_k]$ ii) reduce remaining mass by sampled

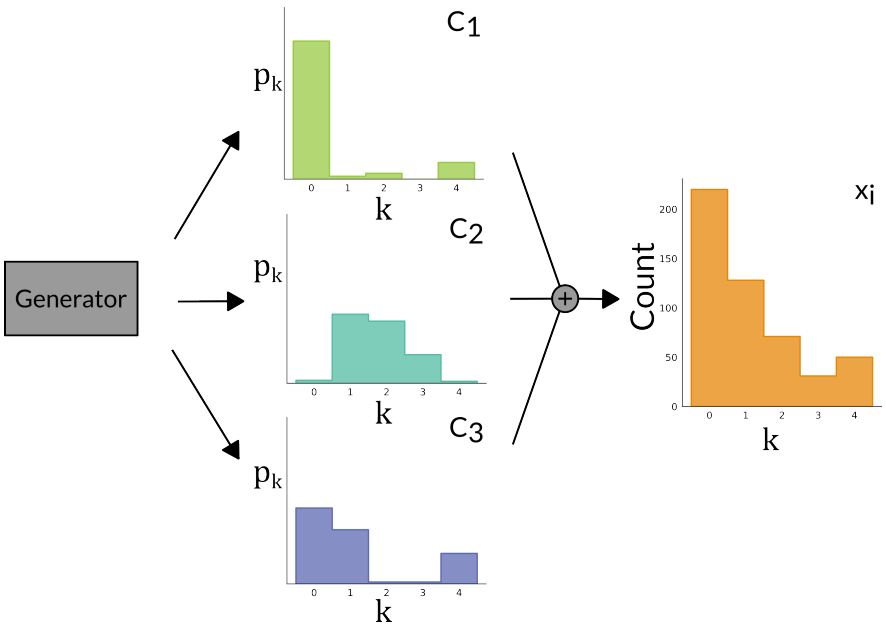

Figure 2.7: Generating Categorical Features Using Independent Probability Distributions for Clusters. To generate a categorical feature, we first generate independent distributions for each cluster across the m possible category values (in this example there are 5). Observations x_i are then sampled according to these distributions for each data point within the cluster. For instance, if the categories represent levels of language proficiency (e.g., $k = 0$: few words, $k = 4$: fluent), individuals in cluster C_1 are predominantly assigned $k = 0$, indicating minimal proficiency, while individuals in cluster C_2 are more likely to be assigned $k = 1$ or $k = 2$, reflecting intermediate levels of proficiency.

value: $xmax_{k+1} = xmax_k - p_k$. iii) repeat $m-1$ times, iv) set remaining mass as final value : $p_m = 1 - \sum_{k=1}^{m-1} p_k$. iv) shuffle elements to ensure that the category with $xmin$ guaranteed mass can occur in any of the m categories. The parameter $xmin$ allows us to control how detectable a cluster is by concentrating mass in a single category.

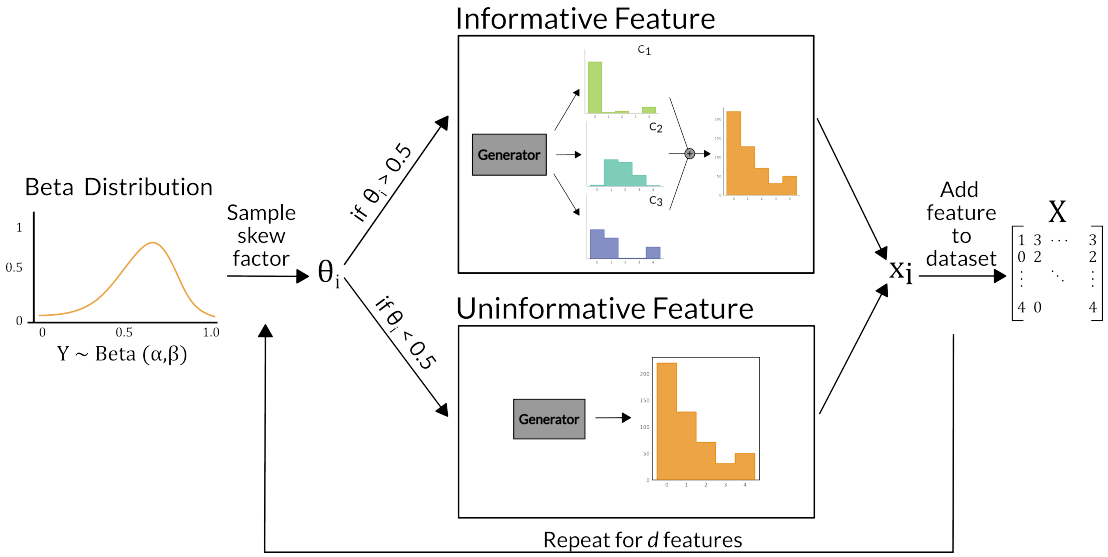

Figure 2.8: Controlling Feature Generation with Beta Distribution for Categorical Data. This figure illustrates how a Beta distribution is used to adjust the informativeness of features in a categorical data generator. Features are generated with and without cluster information ($>$ or <0.5) according to the value of a skew factor θ_i sampled from a beta distribution. The tendency of each probability mass function (PMF), sampled from the generator, to concentrate in a particular category is controlled by θ_i. Values closer to 1 result in clearly defined clusters with more samples from each cluster receiving the same value. In this way, the difficulty of clustering the categorical dataset can be controlled. Our dataset is parameterised by the number and size of the clusters, number of features d and parameters of a beta distribution Beta(α,β).

To generate a feature with detectable clusters, we generate separate PMFs, $P_c = [p_1^c, p_2^c, \dots p_m^c]$ for each cluster c where p_k^c is the probability of a member of cluster c having category k in the feature. We then sample values for each node based on the PMF of the cluster it belongs to, $x_i \sim P_{c_i}$ where c_i is the cluster to which node i belongs to. Finally, we merge our feature as shown in Figure 2.7. Each feature is effectively the sum of independent random variables. The skewness of each cluster's PMF indirectly measures the informativeness of the feature and the ease with which the subclusters can be identified.

In realistic settings, not all features are informative. As a result, we want to include a mix of informative and uninformative features. Partially to reflect real world datasets but also to increase the difficulty of the clustering problem. When generating a set of d features, we use a Beta distribution to control the choice of informative and uninformative features. As

shown in Figure 2.8, to generate a feature we first sample our skew factor θ_i from a Beta distribution, if $\theta_i < 0.5$ our feature will be "uninformative", it will not be generated through a sum of independent subtype distributions. Instead, we generate a single PMF for all nodes, $x_i \sim P$, regardless of their cluster. If $\theta_i >= 0.5$, we generate as normal with each cluster receiving it's own distribution. We use the skew factor to control how skewed the underlying PMFs are. θ_i is scaled from $[0.5, 1]$ or $[0.5, 0]$ to $[0, 1]$ and controls the minimum probability mass of the first uniform random variable. Our dataset is constructed by concatenating the d features together. In this way, the parameters of the Beta distribution Beta(α, β) control how challenging the generated dataset is. Figures 2.9 and 2.10 show examples of data generated using different sets of values for α and β.

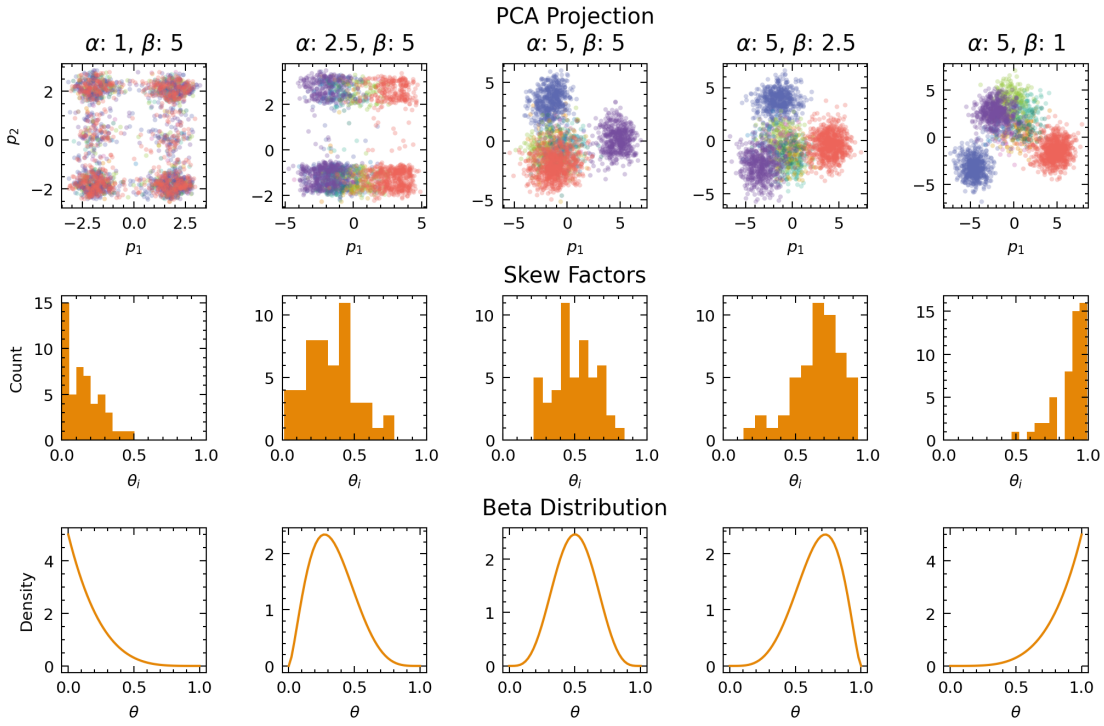

Figure 2.9: Impact of Beta Distribution Parameters on Clustering Difficulty for Categorical Data Data with 50 categorical features are generated for a number of pairs of different α and β values. The two-dimensional PCA projection of the data, the distribution of sampled skew factors θ_i and true Beta(α, β) density function are shown for each pair of (α,β) values. We can see the more informative features that are included in the data the more distinct the clusters are and the easier the clustering problem.

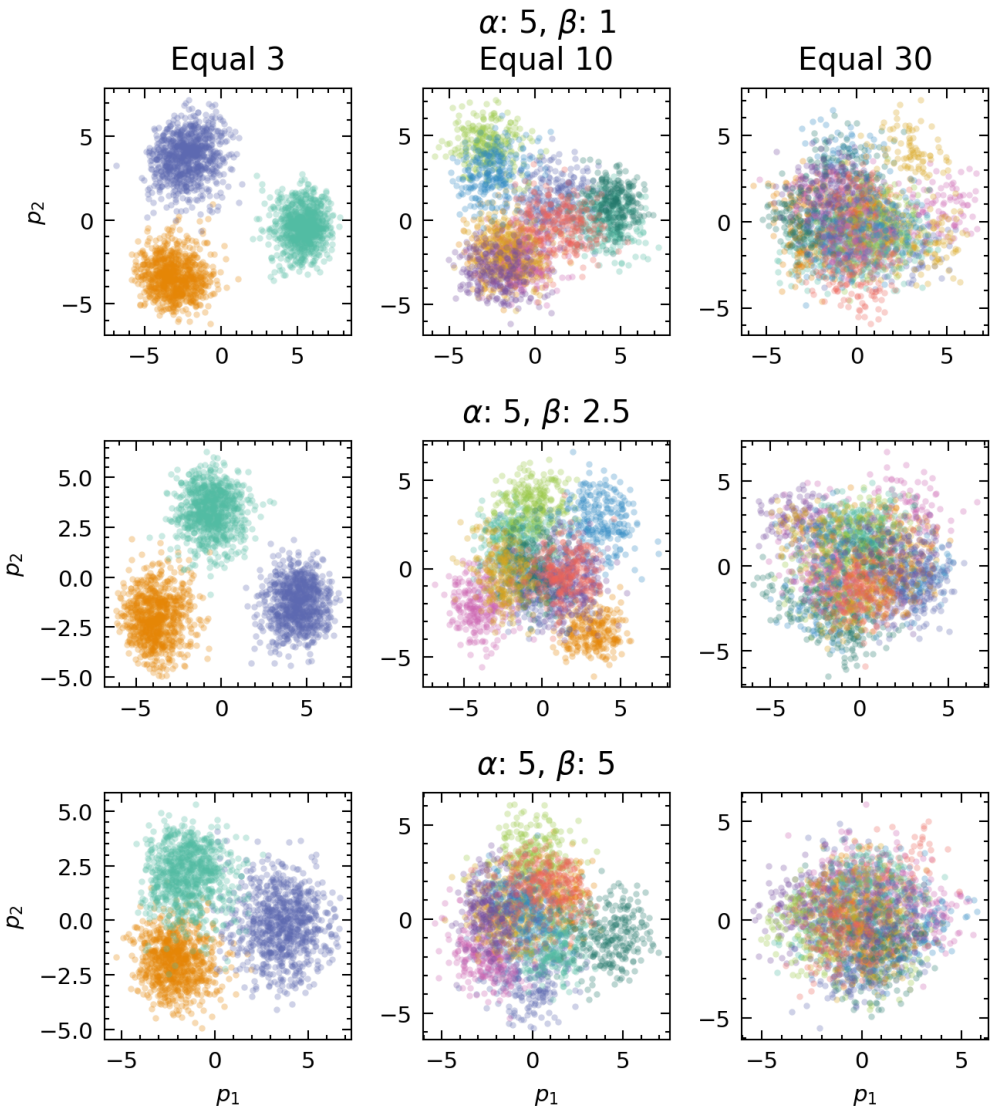

Figure 2.10: PCA Projections of Categorical Data with Different Beta Parameters and Number of Clusters. This figure shows two-dimensional PCA projections of categorical data, generated with fifty features and five categories per feature. The dataset consists of 2500 samples divided into 3, 10, and 30 clusters, with different pairs of α and β values applied. The projections illustrate how clusters become less distinct as the number of clusters increases. For the parameter setting $\alpha : 5, \beta : 1$, representing an "easy" problem, the three clusters are well-separated. In contrast, with 30 clusters, only one or two clusters remain clearly visible, demonstrating the increased difficulty of distinguishing a higher number of clusters in categorical data.

2.4 Experiment Setup

To evaluate the performance of the sparsification methods, I generate datasets of 2500 samples with 50 features These values were selected as they provide a realistic level of complexity both in terms of number of samples and feature dimensions. We generate datasets for each of the distributions described in Section 2.3

- **Mixture of Gaussians**,
- **Mixture of Student's-t distributions**,
- **Categorical Features**.

A key advantage of these synthetic data generators is the ability to adapt both the number and size of the embedded clusters. The specific cluster settings used in the experiments in this work are detailed in Section 2.4.1. In this experiment, I limit each categorical feature to 5 possible levels/categories.

2.4.1 Cluster Settings

A key consideration in evaluating clustering algorithms is the choice of cluster scenarios. In this study, I examine five distinct cluster settings:

- **Equal 3/10/30** — 3/10/30 equally sized clusters.
- **Single Large** — 10 clusters; 1 large cluster containing >50% of nodes, 7 small clusters (1-5%) and 2 medium clusters (10%, 20%).
- **Mixed Sizes** — 10 clusters of mixed sizes; 3 larger clusters (20-30% of nodes), 2 medium clusters (5-10%) and 5 smaller clusters (1-5%)

The motivation behind selecting these cluster settings is twofold: (i) to test the ability of different algorithms to adapt to variations in the number and size of clusters within the network, and (ii) to evaluate how well sparsification methods adjust to changes in the size and density of the underlying data space. Our data generation framework allows for the creation of clusters with arbitrary numbers and sizes, providing flexibility in the design of cluster settings. The chosen scenarios introduce significant variety in the number of clusters to be detected and reflect several realistic situations encountered in biomedical datasets.

The settings with equally sized but increasing number of clusters reduce the density of the clusters in the feature space. If we simply count the size of the clusters in each setting, we range from 833 samples per cluster in *Equal 3*, to 250 in *Equal 10* to only 83 samples per cluster in *Equal 30*. For data generated using the Gaussian and Student's-t distributions in particular, the density of each cluster increases with the number of samples. The chance of observing a strong tightly knit core is less likely with fewer samples per cluster. Coupled with the increased number of different centers, the increase in number of clusters provide a

significantly more challenging problem. These settings, however, do not reflect cluster counts typically encountered in real world settings. In real world settings, we are unlikely to see uniform clusters of similar density in the data space. We, therefore, consider two settings with a more realistic cluster distribution. A setting with one large cluster that surrounded by smaller satellite clusters and a setting with a variety of different cluster sizes. It is important to note the number of samples from each Gaussian distribution corresponds to the relative density in the data space. Variation of cluster size leads to a range of different densities in the feature space. In particular, this creates data where the local distances to its nearest neighbours vary from cluster to cluster. These settings are more challenging and allow us to evaluate the ability of sparsification methods to adapt to areas of different density and the ability of clustering algorithms to handle clusters of different sizes.

2.4.2 Sparsification Hyperparameters

A key challenge in evaluating similarity network construction methods is an inability to fairly compare the performance of different sparsification methods. The difficulty in obtaining fair comparisons arises as a result of an inability to compare different datasets and the difficulty in isolating the effect of a particular sparsification method rather than the effect of choice of clustering algorithm. Furthermore, typically comparisons are performed using pointwise estimates for clustering performance rather than several assessments from multiple data instances. Occasionally cross validated estimates of performance are evaluated. An additional factor that complicates comparison is the dependency on particular choice of graph hyperparameter.

As described in Section 2.2, the different approaches to creating a network from pairwise similarity scores result in networks with significantly different qualities. The choice of graph hyperparameter e.g. number of neighbours K or percentile similarity threshold t, greatly affects the connectivity, density and diameter of each network. Fair comparison between these networks is difficult. It is not immediately obvious how to select equivalent similarity networks from the sparsification methods. A common choice is the number of edges in the network or graph density. However, two networks of similar density may have very different local structure and the "optimal" choice of parameter (in terms of clustering performance) for two different sparsification methods may result in very different levels of graph density. While a hyperparameter evaluation can be conducted on any particular dataset, there is the danger of overfitting to that particular data instance and arriving at conclusions that generalise poorly. Furthermore, the need to conduct both a hyperparameter search and cross validation before arriving at an estimate of clustering performance naturally adds further scepticism to the general conclusions one can draw on sparsification results from a single dataset.

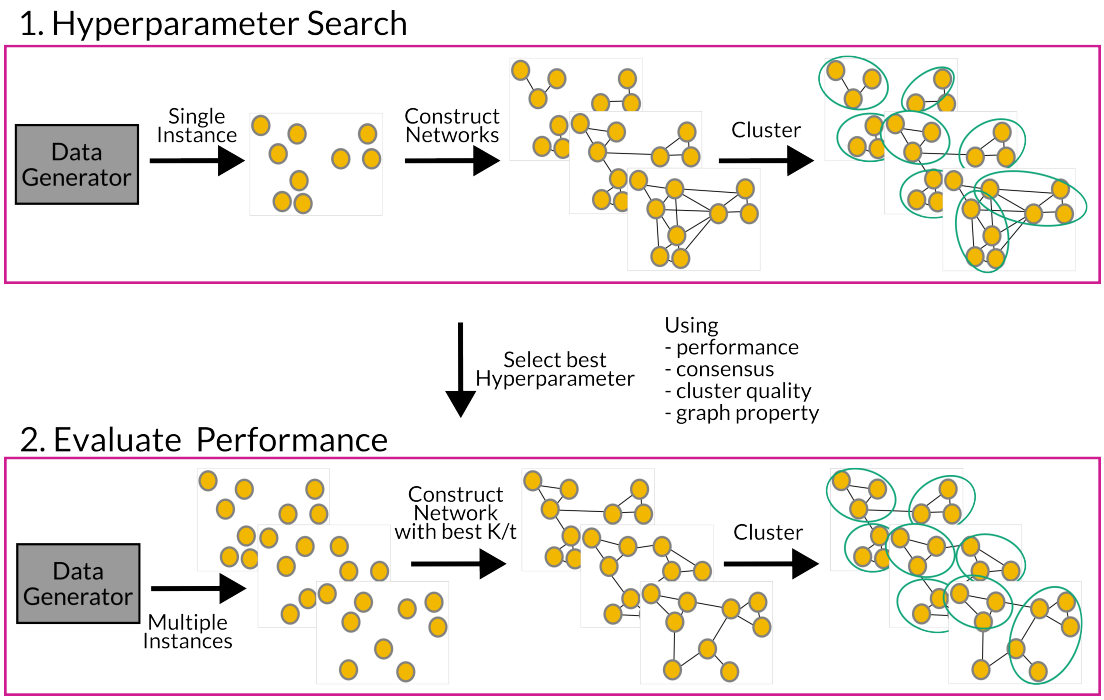

Figure 2.11: Workflow for Evaluating Network Sparsification Methods Using Synthetic Data This figure illustrates the workflow for evaluating network sparsification methods using multiple instances of synthetic data. A hyperparameter search is conducted for all sparsification methods on a single data instance. Various approaches can be used to select hyperparameters — clustering performance of an algorithm, clustering performance of several algorithms, consensus between algorithms or metrics of cluster quality such as mean modularity. Once the "optimal" parameter is selected using one of these criteria, the effectiveness of each sparsification method is evaluated by measuring clustering performance on 10 additional data instances.

Synthetic data offers the ability to estimate the effect of sparsification method without the danger of overfitting to any specific data instance. It is possible to generate several instances of cluster data from the same known underlying distribution. Comparison of performance across several real world datasets is challenging due to differences in data distribution, cluster distribution and problem type that offer no guarantee of equivalence. Through the synthetic data generation, I have a guarantee that the performance of different sparsification methods should be equivalent across instances. Additionally the equivalence in problem setting ensures that differences in performance of clustering algorithms are a result of differences in the sparsification networks not differences in the clustering problem or data distributions.

To evaluate my sparsification methods, I follow the procedure shown in Figure 2.11. First, on a single instance I sample and evaluate 25 different hyperparameters settings for each sparsification method. I use uniform sampling across a range of hyperparameters.

- KNN — $K \in [1, \sqrt{N}]$
- Threshold — $t \in [0.01, 0.10]$ for
- Combined — $K \in [1, 10]$ and $t \in [0.01, 0.03]$
- Linear Skewed KNN — $K \in [1, 2\sqrt{N}]$
- Log Skewed KNN — $K \in [1, 4\sqrt{N}]$

Using the metrics introduced in Section 1.5.2, I evaluate the clustering performance and qualities of the True and Predicted clusters for three distinct clustering algorithms on each sparsification method and each hyperparameter sample. Using this performance, I select my hyperparameters. There are several possible ways to define an optimal hyperparameter

- **Algorithm Performance** — for each algorithm, select the parameter that results in highest ARI/AMI/V-measure.
- **Mean Performance** — Select the parameter with highest average ARI/AMI/V-measure across all algorithms.
- **Consensus** — Select the parameter that results in highest pairwise agreement (pairwise ARI/AMI/V-measure) between the clustering algorithms.
- **Modularity** — Select the parameter with highest mean modularity for all clustering methods.

For any given clustering algorithm, the natural choice of hyperparameter is the one that yields the highest clustering performance. However, when evaluating the quality of network sparsification, a more robust measure might be the parameter that enables effective cluster detection across multiple clustering algorithms. This is particularly relevant in real-world scenarios where ground truth clusters are unavailable, and the optimal clustering algorithm cannot be determined a priori. Moreover, in such cases, traditional performance metrics like ARI, AMI,

or V-measure cannot be used, as the true cluster labels are unknown. Instead, alternative indirect metrics, such as consensus clustering or mean modularity, might serve as better criteria for selecting hyperparameters, providing a more reliable assessment of network quality in the absence of ground truth.

In this work, I focus on ground truth ARI of each clustering algorithm. Although mean clustering performance is crucial, Spectral clustering was found to be quite noisy, leading to significant variability in subsequent parameter selection. For each optimal parameter, I then evaluate their performance on 10 other instances of the synthetic data, providing a robust estimate of sparsification performance. From this, I obtain an estimate of sparsification performance. I perform this evaluation — 25 parameter evaluations and 10 instance evaluations, on each synthetic data distribution and on all 5 clustering problems. This number of parameter evaluations and instance evaluations was chosen to balance the need for a reliable estimate of sparsification performance with the constraints of computational complexity.

The benefit of this approach is twofold. Firstly, it provides an estimation of the distribution of sparsification performance on equivalent networks. Typically, only pointwise estimates of the effect of sparsification choices are feasible for any particular problem due to a scarcity of data. Secondly, it mitigates the risk of overfitting to a particular data instance. The practice of selecting hyperparameters on one instance and evaluating on others can be likened to the practice of using a validation set to identify parameter values in machine learning settings. The key advantage of this approach lies in its ability to quantify the variance in the sparsification process. While each clustering method contributes to the variance in performance, the properties of the data and the clusters remain identical for each instance of a specific cluster setting and distribution. As a result, this approach facilitates a fair comparison, ensuring consistent conditions across different instances and enabling a more robust assessment of the impact of sparsification on network clustering performance.

2.5 Results

2.5.1 Sparsification methods

In Figure 2.12, we can see the ARI score of the SBM clustering prediction and the true cluster membership across all five cluster settings for all five sparsification methods. Figure 2.12A shows the change in performance across different hyperparameter settings and Figure 2.12B shows the ARI performance across 10 instances with the optimal hyperparameter. For each cluster setting, the hyperparameter was selected using the parameter with the highest ARI of the SBM algorithm on that particular problem. We order the different parameter settings

using density of the networks to allow comparison between the different parameters types — K and t. The *Threshold* network is consistently the worst performing. The higher density is necessary to ensure a connected network with limited isolated nodes. For all networks the performance of SBM decreases as density increases. The *Log-skewed KNN* performance does not drop as significantly as other KNN methods. In addition in Figure 2.12B, we can see the performance more consistent across different settings. *Linear-Skewed KNN* is more noisy but has higher mean ARI across the 50 evaluations.

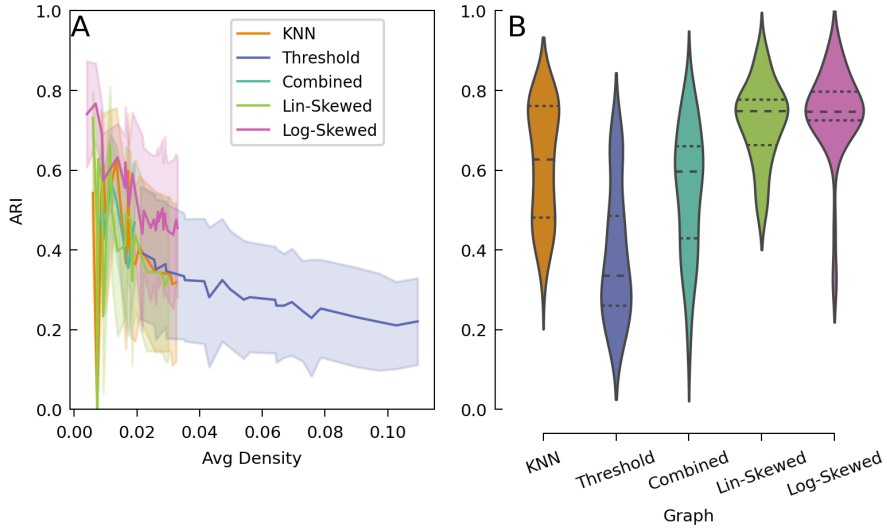

Figure 2.12: Hyperparameter Search and Performance Evaluation of Sparsification Methods using SBM clustering ARI The ARI SBM clustering score of the five sparsification methods across all five cluster settings of mixed Gaussian data is shown using euclidean distance as a metric. Panel **A** shows the results of hyperparameter evaluation, showing how ARI changes with varying hyperparameters. To fairly compare the different parameters (K & ε), we plot ARI vs graph density. To account for the high number of isolated nodes and subcomponents at low densities, *Threshold* networks are evaluated over a broader range of densities. Panel **B** displays the distribution of ARI SBM scores across 10 instances, with hyperparameters optimised for the highest ARI performance. The *Threshold* network consistently performs worse than all other methods, with a significant difference ($p < 1 \times 10^{-12}$)

Figure 2.13 shows the ARI performance of (A) SBM, (B) Leiden, and (C) Spectral clustering methods on all five cluster problems evaluated on ten instances of mixed Gaussian data. For each problem and each clustering algorithm the highest ARI performing graph hyperparameter is selected. The *Threshold* method is the worst performing method across all algorithms and all cluster problems. The behaviour of the three clustering algorithms are quite distinct. The Leiden algorithm is consistent across all sparsification methods and cluster problems. While the *Threshold* method has a drop in performance relative to the other sparsification methods it is not as significant as with the SBM and Spectral algorithms. The SBM algorithm

on *Log-Skewed* and *Linear-Skewed KNN* outperform the traditional *KNN* in settings with larger clusters — *Equal 3*, *Single Large* and *Mixed Sizes*. Their performance is equivalent in settings with a larger number of clusters. The Spectral algorithm does not perform as well as the other algorithms in nearly all settings although it is notable that the *KNN* method has a higher performance than the *Linear-Skewed KNN* on the *Mixed Sizes* cluster problem.

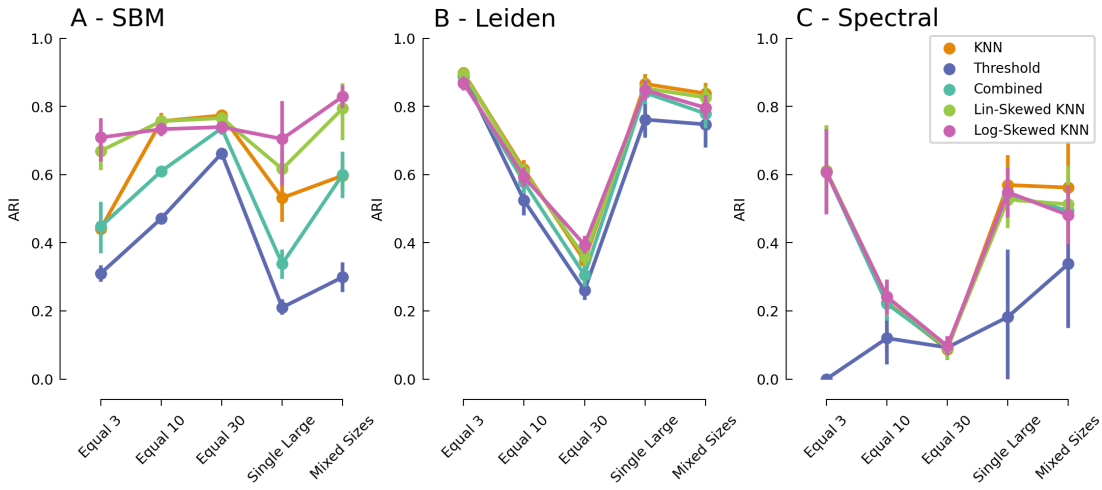

Figure 2.13: ARI Performance of Sparsification Methods Across Different Clustering Algorithms on Mixed Gaussian Data. This figure shows the mean ARI performance of various sparsification methods across three clustering algorithms: **A** SBM, **B** Leiden, and **C** Spectral, evaluated across 10 instances of mixed Gaussian data. Each data point represents the mean ARI on networks using the hyperparameter found to havemaximum ARI for each clustering algorithm and sparsification method respectively. 95% confidence intervals across the 10 instances are indicated. The *Threshold* method consistently performs the worst across all algorithms and cluster settings. Conversely, *Log-Skewed KNN* enhances the performance of the SBM algorithm, particularly in problems involving large clusters such as *Equal 3* and *Single Large*.

While there are noticeable difference in clustering performance for the sparsification methods, there are also notable differences between the three clustering algorithms. The performance of the SBM and Leiden algorithms on *KNN* and *Threshold* networks across the cluster problems are inverted (Figure 2.13A vs.2.13B). The Leiden algorithm has greater performance in settings with large clusters while the SBM algorithm is superior in settings with a higher number of equally sized clusters — *Equal 10* and *Equal 30*. The Spectral algorithm is far more variable in on all networks and perform particularly poorly in the settings without large clusters.

The disparity between KNN methods and the *Threshold* approach is even more significant in high noise settings as shown in Figure 2.14. Figure 2.14 shows the ARI performance of (A) SBM, (B) Leiden and (C) Spectral algorithms on 10 instances of mixed Student's-t data. Both the SBM and Spectral algorithms fail to converge to a solution on the *Threshold* network. The labels produced are almost random. While the difference in performance of the Leiden algorithm was not too significant in the low noise setting with the addition of noisy clusters the *Threshold* network performs far worse compared to KNN methods. The combined sparsification method does improve that of the threshold network but simply using a KNN method is still more optimal. It is notable that the *Log-Skewed KNN* no longer provides an improvement compared to the *KNN* even in settings with large clusters. The *Linear-Skewed KNN* does show some differences but there difference cannot be explained by cluster size.

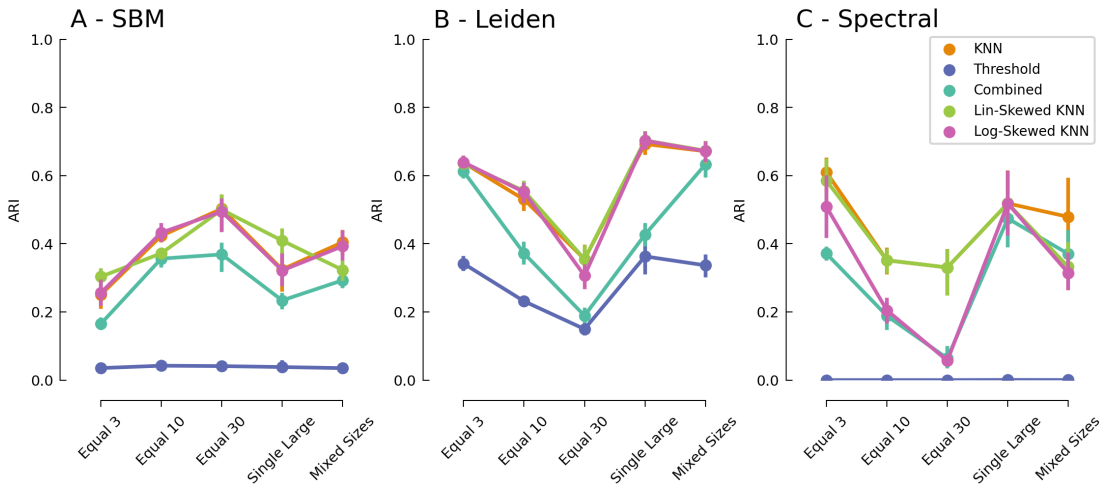

Figure 2.14: ARI Performance of Sparsification Methods Across Clustering Algorithms with Mixed Student's-t Data This figure presents the ARI performance of various sparsification methods across three clustering algorithms: **A** SBM, **B** Leiden, and **C** Spectral, evaluated over 10 instances of mixed Student's-t data. The *Threshold* network shows a significant drop in performance the in high noise settings. It performs noticeably worse with Leiden clustering compared to its performance with a mixture of Gaussians (see Figure 2.13B), and its performance is almost random for Spectral and SBM clustering.

Differences in performance of the clustering algorithms can be seen in the high noise setting. SBM and Leiden performance on the *KNN* network show similar trends across the cluster problems as those seen in the Gaussian data (Figure 2.13). SBM is best with a higher number of smaller clusters (*Equal 30*) and the Leiden is best with large clusters (*Equal 3*/*Mixed Sizes*/*Single Large*). However, the SBM algorithm shows a consistent drop in ARI performance. While SBM still detects larger number of clusters more accurately the Leiden algorithm is more robust to noise and does not show as significant a drop in performance between the high and low noise distributions. Unlike the SBM and Leiden, the Spectral algorithm performs better on the *KNN* network in the *Equal 10* and *Equal 30* cluster problems in the high noise

mixed Student's-t distributed data than in the low noise mixed Gaussian data. Additionally, the drop in performance on the larger cluster problems is not as significant as the drop in performance of the SBM algorithm.

In Figure 2.15, the ARI performance of the sparsification methods using (A) SBM, (B) Leiden and (C) Spectral is depicted for 10 instances of categorical data. Spectral clustering shows more accuracy across all network types on categorical data. While its variance is increased, it consistently outperforms SBM clustering on data with larger clusters. It even outperforms Leiden clustering on the *Single Large* clustering problem.

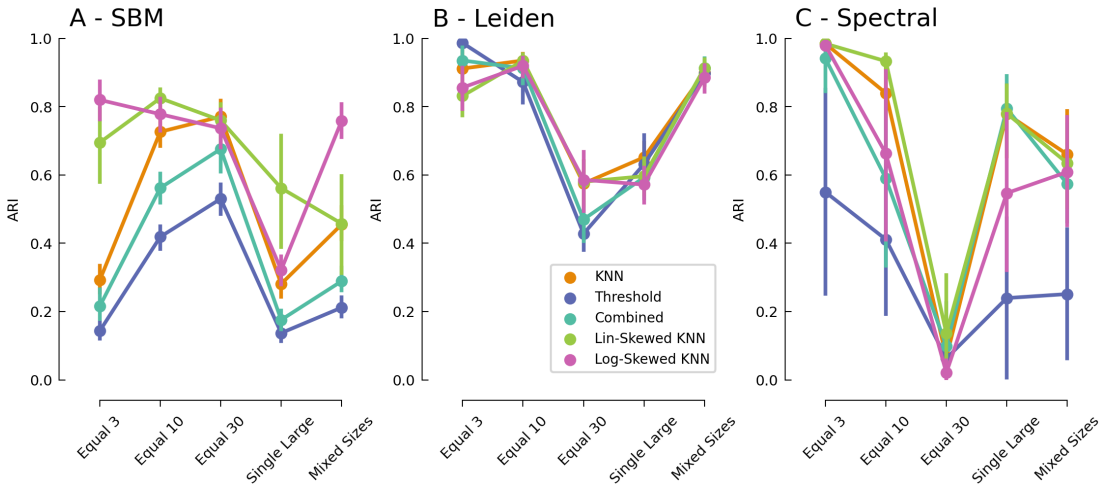

Figure 2.15: ARI Performance of Sparsification Methods Across Clustering Algorithms with Categorical Data. This figure displays the ARI performance of various sparsification methods across three clustering algorithms: **A** SBM, **B** Leiden, and **C** Spectral, evaluated over 10 instances of categorical data. Consistent with results from other distributions (Figures 2.13 and 2.14), the *Threshold* method is the poorest performer across all three algorithms. For SBM clustering, there is a noticeable performance gap between KNN and the Log-Skewed and Linear-Skewed KNN networks, especially for problems with large clusters such as *Equal 3*, *Single Large*, and *Mixed Sizes*. Increased variance across methods and networks highlights greater differences between instances of categorical data.

In line with its results on other distributions (see Figures 2.13 and 2.14), Threshold is consistently the poorest performing method across all three algorithms. Surprisingly, for categorical data with the *Equal 3* clustering problem, Threshold was best performing sparsification method for Leiden clustering (Figure 2.15B). However, this single data point of improvement does not overcome the evidence across the different distbrituions that KNN sparsification is more suited to clustering than Threshold.

Consistent with results observed for other data distributions (see Figures 2.13 and 2.14), the Threshold method consistently performs poorly across all three clustering algorithms. Notably, for categorical data in the *Equal 3* clustering problem, the Threshold method unexpectedly performs better with the Leiden clustering algorithm (Figure 2.15B). Despite this single instance of improvement, the overall evidence from various distributions indicates that KNN sparsification is generally more effective for clustering compared to the Threshold method.

Interestingly, there is a wider gap between KNN and the Log and Linear-Skewed KNN networks for SBM clustering ARI on problems with large clusters (*Equal 3*, *Single Large*, and *Mixed Sizes*). Skewed KNN methods continue to provide an improvement in performance on these types of problems. It should be noted the variance of all methods and networks is increased on categorical. This suggests there greater differences between instances of the categorical data.

Degree Distributions

Figure 2.16 shows the degree distribution for the five sparsification methods. The graph hyperparameters are selected to produce a graph density of 0.025 for each of the cluster problems (A) *Equal 3*, (B) *Equal 10*, (C) *Equal 30*, (D) *Single Large* and (E) *Mixed Sizes*. The distributions for all graphs are quite consistent across the different clustering problems. *KNN* is linear in the log log plot. There are no nodes with degree less than K but the count drops exponentially as degree increases. The *Threshold* network by contrast has very different behaviour, it is log normal with large number of low degree nodes and a number of large degree nodes. The *Combined* method is a combination of both *KNN* and *Threshold* for low K and low t and the effect of this combination is visible in its degree distribution. Similar to the *KNN* network there are no nodes of degree less than K. There is a spike in the frequency of nodes of degree K as seen in the *KNN* network but the distribution for nodes of degree $> K$ is approximately identical to the *Threshold* network.

Both the *Linear-Skewed KNN* and *Log-Skewed KNN* allow the inclusion of nodes with degree $< K$ in a KNN graph. However there is very different behaviour between the two. The *Linear-Skewed KNN* fails to include many low degree nodes. The degree distribution is still quite similar to the *KNN* network but with an addition of node degrees just less than K. The *Log-Skewed KNN* includes a more diverse set of node degrees. It contains many low degree nodes and has a very different degree distribution to the *KNN* network. It closer to the *Threshold* network with nodes of all degrees. The *Log-Skewed KNN* is effective at lowering the density of a *KNN* network. To achieve a density of 0.025 a choice of $K = 50$ can be used for the *KNN* but $K = 400$ is required for the *Log-Skewed KNN*.

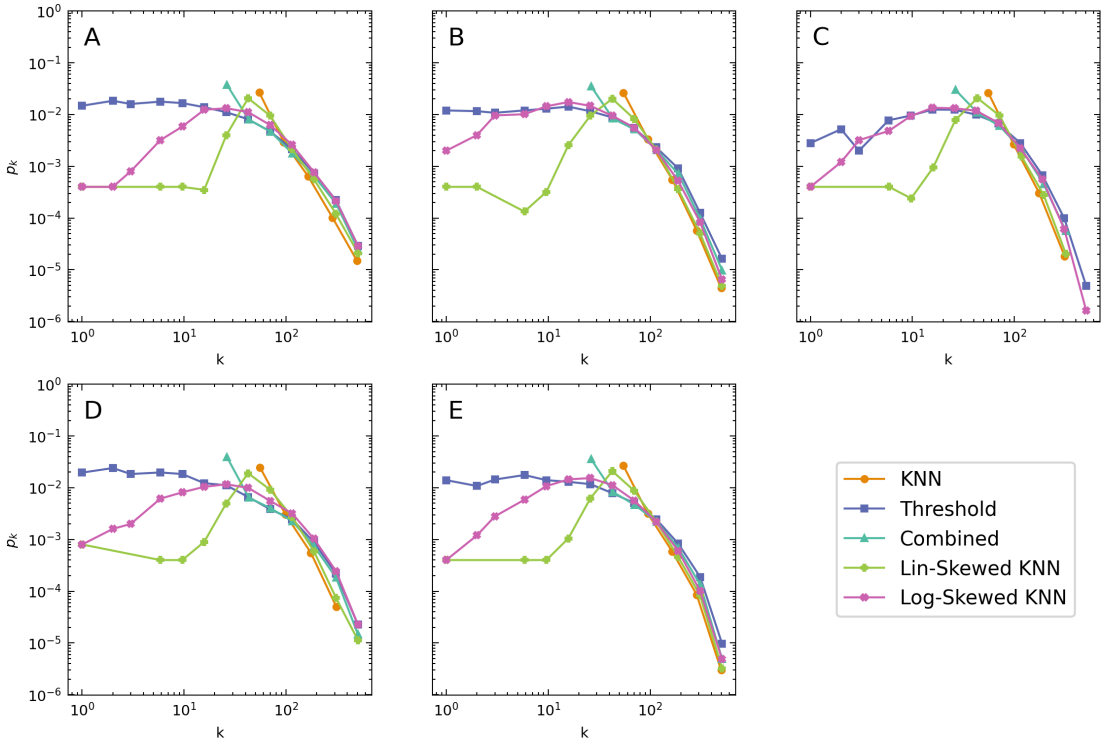

Figure 2.16: Example Degree Distributions of Networks with Identical Density Across Different Clustering Settings. Example degree distributions of networks for all sparsification methods on the five clustering problems; **A** Equal 3, **B** Equal 10, **C** Equal 30, **D** Single Large and **E** Mixed Sizes. Parameters are chosen so that each network has a density of 0.025. Data from a mixture of Gaussians is used in each instance. The *KNN* is linear in the log-log plot showing the count drops exponentially as degree increases. By design it has no low degree nodes. The *Threshold* network has a log normal degree distribution. The *Combined* network resembles the *Threshold* distribution at high degree nodes and the *KNN* for its lowest degree nodes. Both the *Linear-Skewed* and *Log-Skewed KNN* facilitate the inclusion of nodes with degree $< K$, however, the *Linear-Skewed KNN* fails to include a significant number of low degree nodes unlike the *Log-Skewed KNN*.

Network Properties

The differences in graph properties extend beyond the degree distribution. Figure 2.17 shows the pairwise distributions of the *Threshold*, *KNN*, and *Log-Skewed KNN* networks for Gaussian distributed data across all five cluster problems. The graph hyperparameters were selected to optimise SBM ARI performance. The distributions for modularity of (I) ground truth clusters, (II) graph diameter, (III) average path length on the network, (IV) degree assortativity and (V) the ARI of fitted SBM models are shown. The most significant difference between the *Threshold* and *KNN* networks is the average path length (2.17I). Average path length is the average number of steps along the shortest paths between all pairs of nodes in the network. We can see the *Threshold* networks are very compact with only 2-3 steps on average between nodes in the network (2.17III). The degree assortativity also highlights distinct behaviour between the graph types. *KNN* network are dis-assortative (2.17IV). Edges are more likely to occur between low and high degree nodes. There is no clear assortativity to *Threshold* networks. There is no strong connection pattern based on node degree. There are also notable differences in the modularity of the ground truth clusters. *KNN* and *Log-Skewed KNN* networks are significantly more modular and have more interconnectivity within clusters (2.17I).

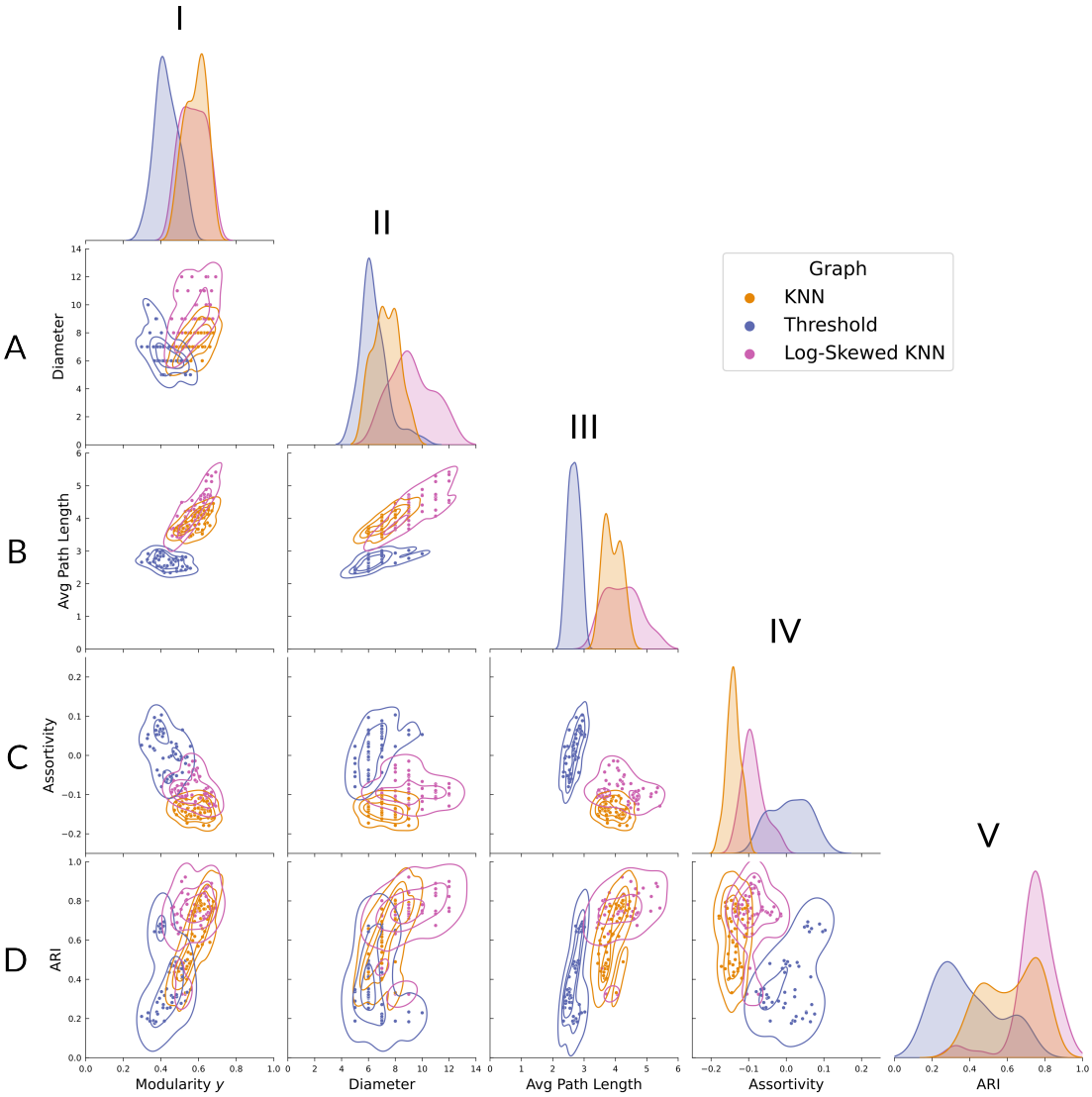

Figure 2.17: Pairwise Distributions of Network Metrics and SBM ARI for Various Sparsification Methods on Gaussian Data. Pairwise distributions of **I)** ground truth cluster Modularity, **II)** Graph Diameter, **III)** Average shortest Path length, **IV)** Degree Assortativity and **V)** SBM algorithm ARI for *Threshold*, *KNN* and *Log-Skewed KNN* networks on Gaussian data. *Threshold* networks are less modular (I) and its clusters are more interconnected with lower diameters and shorter average path lengths (B II). *KNN* networks are dis-assortative (IV) with connections between low and high degree nodes more likely. *Log-Skewed KNN* networks have larger diameters and are more assortative than KNN networks (C II).

2.5.2 Clustering Algorithms

In Figure 2.18, performance of each clustering method is shown for the Gaussian data and euclidean metric. This the average performance for each clustering method on all five network types and all five clustering problems. In Figure 2.18A, the performance is plotted vs density of the corresponding network under a particular parameter e.g. *KNN* for with $K = 20$ might have an average density 0.015 over the five different clustering problem. In Figure 2.18B, we show the performance of each method on 10 other instances of the data (for all clustering problems and all graph types) where the graph hyperparameters are selected using the best ARI performance of each clustering method on a particular clustering problem e.g. *KNN* $K = 10$ is best SBM ARI on *Equal 30*, $K = 20$ is best Leiden ARI on *Equal 30*. We can see the Leiden algorithm is by far the most consistent clustering algorithm across all densities. The SBM algorithm performance is competitive at lower densities but drops as the density increases. This is unsurprising as the number of possible Monte Carlo swaps increases with the number of edges in the network. Most significantly we can see that Spectral clustering is noisy. The frequent spikes and area of significantly lower performance is a result of the eigengap heuristic not identifying neither the correct nor a plausible K. When it does converge the performance is often competitive with the other algorithms. It must be noted a key aspect of the community detection problem is the identification of the number of clusters and the inability to converge is a serious drawback.

As described in Section 1.5.2, clusters of different sizes are equivalent to class imbalances in multi/binary classification problems. In Figure 2.19, we can see the performance of each method when accounting for different cluster sizes — small <7.5% of nodes and large >7.5% of nodes. We show the distribution of F_1-score performance of between best predicted cluster for each true cluster of all three clustering algorithms on all five clustering problems and all five types of network. Again for each network the graph hyperparameters are selected on one instance using the best performing ARI of a particular algorithm and evaluated on 10 data instances. Figure 2.19A, 2.19B and 2.19C show the performance with Gaussian, Student-t and Categorical distributions. We can see the both the Leiden and Spectral algorithms fail to detect smaller clusters accurately. The SBM algorithm is more consistent. It's performance on all cluster sizes is roughly equal across all distributions. The Leiden algorithm is more proficient than the SBM at detecting larger clusters but it very rarely detects smaller clusters.

In Figure 2.20 the predicted number of clusters for (A) SBM, (B) Leiden, and (C) Spectral by network is compared to the ground truth number of clusters on mixed Gaussian data. The ground truth number of clusters varies by cluster problem (as seen by the dashed grey line). From Figure 2.20A, we can see SBM consistently over-predicts the number of clusters in a network. This is true for all network types. SBM is most accurate when dealing with

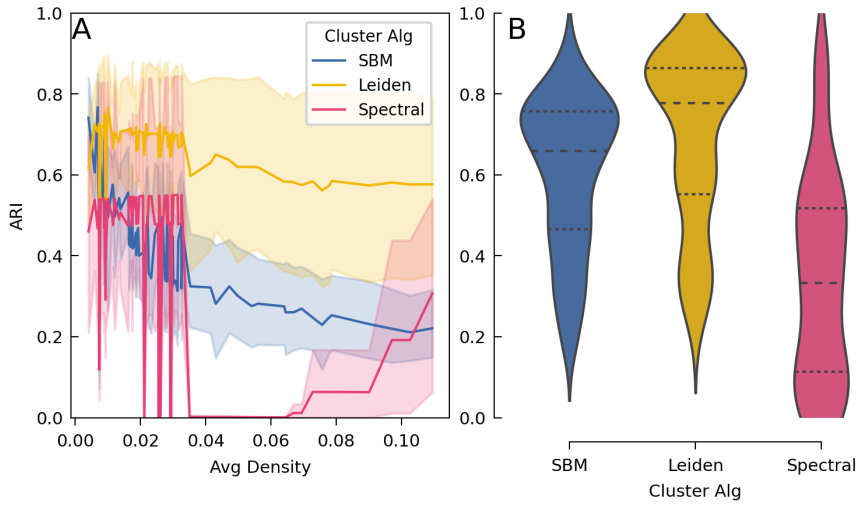

Figure 2.18: ARI Scores of Clustering Algorithms Across Sparsification Methods with Mixed Gaussian Data The ARI score of the three clustering algorithms across all five cluster settings and all five sparsification methods of mixed Gaussian data using euclidean distance as a metric. Panel **A** shows the change in performance for different hyperparameter choices. The Leiden algorithm is the most consistence across all parameter settings. SBM performs better at low network densities and drops in performance the more edges that are added to the networks. Spectral is noisy and frequently fails to converge to a solution. Panel **B** shows the distribution of ARI score across 10 instances where the hyperparameter is selected to maximise each clusters performance. Leiden is again the most consistent and has the highest average performance. The methods vary significantly in performance across the clustering problems but Spectral is noticeably the worst performing method by a significant marge ($p < 1 \times 10^{-20}$).

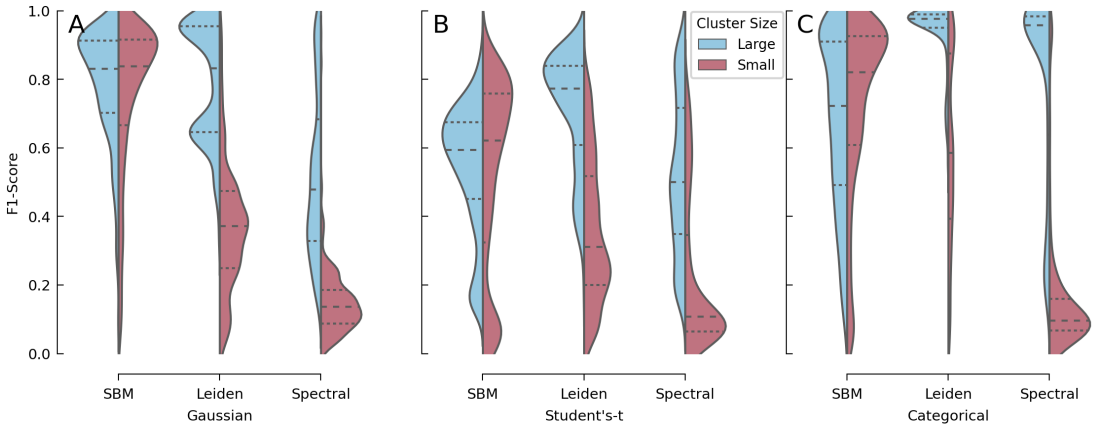

Figure 2.19: Per Cluster F_1-Score of Clustering Algorithms Across Different Data Distributions. Per cluster F_1-score of the three clustering algorithms on **A** Gaussian, **B** Student's-t and **C** Categorical data distributions on all network types and all clustering problems. The performance of the algorithms at classifying small (<7.5% of nodes in the network) and large clusters (>7.5% of nodes) are shown. The SBM is most consistent across all distributions and has equivalent performance predicting large and small clusters. Leiden is very good at detecting large clusters but fails to distinguish small clusters in all 3 distributions. Spectral is also poor at detecting small clusters but also suffers poor performance in predicting large clusters in Gaussian and Student's-t distributed data.

problems with a large number of underlying clusters, *Equal 30*, and overfits to non-existent subclusters in problems with fewer, *Equal 3*. In contrast, Figure 2.20C shows that Spectral generally under-predicts the number of clusters in the network. On problems with mixed clusters sizes, *Single Large* and *Mixed Sizes*, it underfits and groups smaller subclusters together the larger embedded clusters. Leiden is generally the most accurate (Figure 2.20B), successfully detecting the number of clusters for *Equal 3*, *Equal 10* and *Equal 30*. While it similarly underfits the number of clusters on problems with mixed sizes, it does so to a lesser extent than Spectral.

It must be noted that this behaviour varies across networks. On the Threshold network, Leiden consistently predicts a very large number of clusters across all cluster problems. However, the Adjusted Rand Index (ARI) (see Figure 2.13B) does not always show a corresponding drop in performance; for example, *Equal 3* ARI is very high. This large number of clusters is likely comprised of a low number of large clusters combined with many isolated clusters containing only one or two nodes. Additionally The improved SBM ARI of the skewed KNN networks (Figure 2.13A) on problems with large clusters results from predicting fewer clusters (Figure 2.20A). Unlike the KNN network, large clusters are not split into smaller subclusters on these networks.

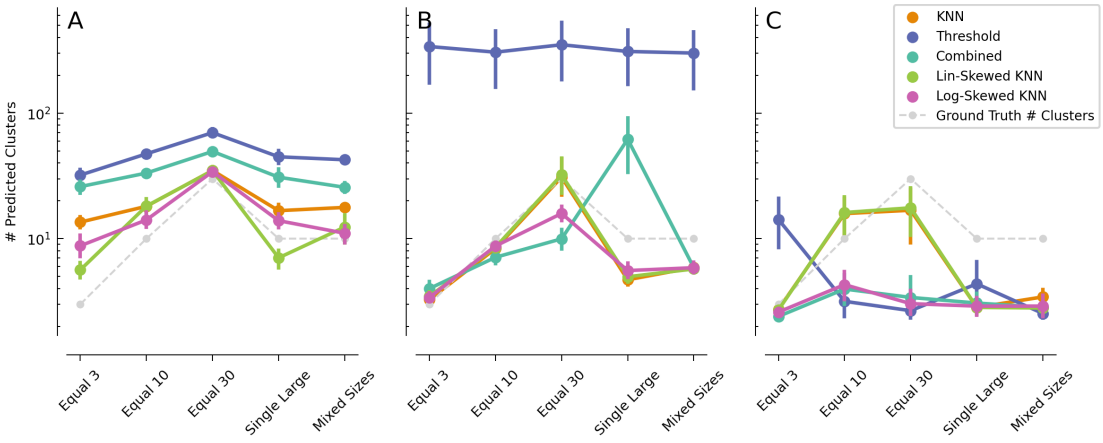

Figure 2.20: Predicted vs. Ground Truth Number of Clusters for Clustering Algorithms on Mixed Gaussian Data. This figure compares the predicted number of clusters for **A** SBM, **B** Leiden, and **C** Spectral clustering algorithms against the ground truth number of clusters on mixed Gaussian data. The ground truth number of clusters varies by cluster problem, indicated by the dashed grey line. The clustering algorithms' behavior depends on the underlying network. On the *Threshold* network, Leiden consistently predicts a significantly larger number of clusters across all cluster problems. Despite this overestimation, the ARI (see Figure 2.13B) does not always show a corresponding decrease in performance, as exemplified by the high ARI for the *Equal 3* problem. This discrepancy suggests that the large number of predicted clusters often includes a few large clusters combined with many isolated clusters of one or two nodes. The improved ARI of the skewed KNN networks in problems with large clusters can be attributed to their closer alignment with the ground truth number of clusters, as these networks do not split large clusters into smaller subclusters unlike the *KNN* network.

Figure 2.21 visualises an example clustering for the three methods (A) SBM, (B) Leiden and (C) Spectral on a *KNN* network for Gaussian *Mixed Sizes* data. The nodes are grouped by predicted cluster and coloured by their ground truth cluster. We can see there is a disparity in the number of clusters predicted with the SBM predicting 22 clusters, Leiden 6 and Spectral only 4. The SBM has high homogeneity, meaning the members of a predicted cluster belong to the same true cluster, but low completeness, as it separates the largest ground truth clusters. In contrast, the Leiden algorithm has high completeness, where all members of the same class are in the same predicted cluster, but low homogeneity, as it combines several smaller clusters together. The Spectral clustering behaves similarly to the Leiden algorithm but is more extreme, combining larger ground truth clusters. This illustrates the reason behind the poor F_1 performance on smaller clusters for both the Leiden and the Spectral algorithms — they fail to detect smaller clusters, while the SBM algorithm performs poorly with larger clusters as it frequently splits them into subclusters.

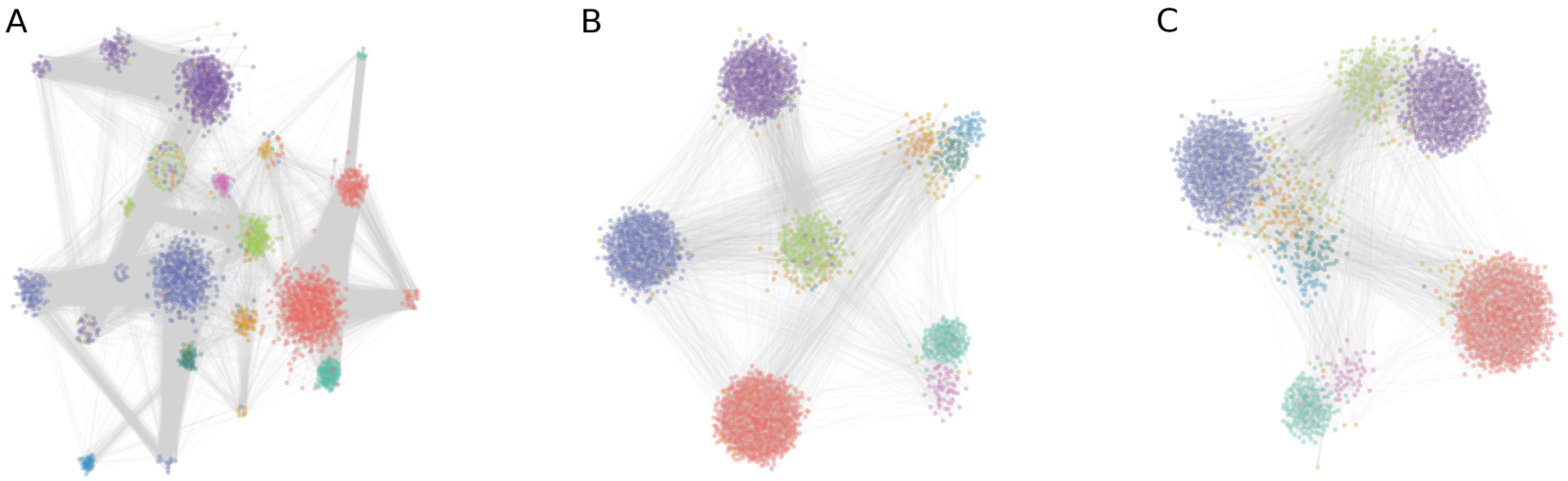

Figure 2.21: Visualization of Clustering Predictions on KNN Network with Mixed Sizes Data. Visualisation of clustering predictions for **A** SBM, **B** Leiden and **C** Spectral on KNN network constructed from *Mixed Sizes* mixed Gaussian data. Nodes are grouped by predicted cluster $\hat{y}$ and coloured by ground truth cluster y. In general, SBM has higher accuracy in identifying smaller clusters but splits larger clusters into relatively homogeneous subgroups. Leiden predicts large clusters well but groups smaller cluster together. It does not distinguish small clusters but has higher ARI due to the fewer number of predicted clusters. Spectral is similar to Leiden but underfits even more — predicting fewer clusters and even failing to separate larger clusters.

2.6 Discussion

Despite its widespread use, *Threshold* sparsification produces networks that are far less conducive to community detection than *K-Nearest Neighbour* sparsification. In this work, I have observed measurable differences between the two approaches in cluster performance across a range of clustering algorithms and in both high and low noise settings. *Threshold* networks constructed from data with high level of noise and numerous outlier nodes are particularly poor. Local behaviour is essential for community structure and the global nature of the threshold approach does not lend itself to use in community detection. While theoretically a metric should accurately rank the highest similarity between entities and ordering by this ranking should result in the inclusion of only the most informative edges in the network, in reality, differences in density and lack of information at the peripheries of clusters result in less modular structure and networks that ultimately do not contain sufficient community information. *Threshold* networks contain more inter community connections resulting in lower average path lengths. The local nature of *KNN* results in more modular networks with less connections between communities.

While some concerns exist with *KNN* construction regarding both the over allocation of edges to outlier nodes and the inclusion of edges with lower similarity evidence, there is no significant improvement in performance when adjustments are made to the number of edges assigned to each node based on density in the feature space. This can be seen from the comparable performance of all clustering algorithms on the *KNN* and *Log-Skewed KNN* networks in the high noise setting in Figure 2.14. While skewed KNN networks enabled SBM clustering to detect larger clusters, my particular approaches to adjusting the number of neighbours are not consistent enough. In some problems Log-Skewed produced a more favourable network, in others Linear-Skewed performed better (see Figures 2.13, 2.14, 2.15). An additional drawback for both methods is the introduction of additional hyperparameters in the density estimation step. The *KNN* is the simplest method of sparsification and the most consistent across the data and clustering algorithms evaluated here. It allows the inclusion of information from both high and low density areas within the data. Furthermore, *KNN* construction is scalable to larger networks through the use of approximate KNN construction. In this work, I have presented strong evidence that *KNN* sparsification should be used when constructing similarity networks for community detection.

I have also demonstrated the utility of SBM and Leiden clustering algorithms. SBM is suited to settings where a large number of smaller clusters are expected. The Leiden algorithm is more accurate in the classification of larger clusters. It is also more accurate in its estimation of the number of clusters, SBM consistently overestimates the number of clusters in a network (Figure 2.20). While I have shown the SBM algorithm detects smaller clusters more accurately (Figure 2.19), applying this in practical real-world scenarios may pose considerable

challenges. Without ground truth labels, distinguishing between SBM splitting large clusters incorrectly in sub-clusters and SBM successfully detecting true small clusters is highly challenging. Leiden is more resilient to noise and, while there are valid reasons to have misgivings on modularity maximisation, ultimately it results in more accurate cluster prediction.

Spectral clustering was found to be the noisiest clustering method. It consistently underfit and failed to detect large numbers of clusters. If the correct number of clusters were identified, the algorithm performed quite well. This highlights the challenges with Spectral clustering — there are no in built methods for identifying the number of clusters. This contrasts with unlike SBM's description length and Leiden's resolution parameter[1]. I made use of the eigengap heuristic which led to selections of low number of clusters. With improved identification of the number of clusters, spectral clustering could prove to be extremely effective.

2.6.1 Limitations

The generalisability of the conclusions that can be drawn from this chapter should be cautioned by the use of synthetic data and the limited set of similarity measures. The synthetic data is derived from a constrained set of data distributions, where clusters are centered around distinct points in space. The variance within each cluster is standardised across all clusters and uniform in all dimensions[2]. Both the mixture of Gaussian and Student's-t distributed data exhibit clusters characterised by dense cores, where interactions between clusters occur only at their periphery. This data may not replicate the intricate features or patterns found in real-world scenarios. Although attempts were made to evaluate noisier data with outliers using a mixture of Student's-t data, this noise distribution might not fully capture real-world complexity but rather increase cluster interactions in a simplified manner. My categorical data generation aims to provide an alternative distribution where clusters do not consist of dense cores scattered through the feature space. The generation process does assign separate distributions to individual clusters within a particular feature, however, the construction of consistent datasets from such features can be challenging. Furthermore, the current distribution generation falls short in producing ordinal features, a key element of many clinical questionnaires.

Another limitation is the current selection of clustering algorithms. The use of a limited selection of clustering algorithms constrains my analysis in two ways. Firstly, I am seeking to evaluate the general suitability of a network for clustering using metrics like mean clustering performance, but these metrics can be highly dependent on the performance of a single method. For instance, the convergence or lack thereof of a noisy method, such as my spectral clustering algorithm, can significantly impact hyperparameter selection. By including more

1. Leiden aims to maximise modularity. Modularity is a clear choice of metric to select between resolution parameters. Spectral does not have such a metric.
2. Each cluster has identity covariance.

algorithms, I can reduce my sensitivity to the convergence of single algorithm. Secondly, while the three selected algorithms are highly representative of network-based community detection algorithms, a more comprehensive set of algorithms would offer a better evaluation of the flexibility and generalisability of different network approaches.

2.6.2 Future Work

A key challenge in network construction is hyperparameter selection. The aim is to select a hyperparameter that leads to optimal clustering performance. In cases where true cluster labels are available, there is a significant danger of overfitting if we perform this selection using clustering performance. In this chapter, I present an approach using synthetic data that allows us to mitigate the risk of overfitting by generating multiple instances of data. In real world scenarios, ground truth cluster information is rare. Fortunately, my data generation framework provides an opportunity to investigate alternative optimisation strategies. A future avenue of research is to conduct a more in-depth exploration of hyperparameter selection for sparsification methods to identify optimal configurations. Network density, clustering consensus and metrics such as modularity are all employed to select and rank networks. With this framework, I can directly relate these metrics to the ground truth, gaining insights into their performance.

An interesting avenue for exploration would be the development of dynamic threshold sparsification methods. In this work, we made use of non-fixed K-nearest neighbours (KNN) networks as simple implementations of dynamic thresholds. These methods estimated local density to create approximate dynamic thresholds for each node by adjusting the number of neighbours to retain when sparsifying. These simple variations in KNN networks — Linear and Log Skewed KNN networks — have shown improvements in estimating the number of clusters, particularly in Stochastic Block Model (SBM) clustering. However, my approach to the selection of dynamic thresholds and the estimation of local density remains crude, as evidenced by the inconsistencies in the optimality of log mapping or linear mappings across different distributions and cluster configurations. Furthermore, both methods added a number of additional parameters that would require optimising in practical settings — number of neighbours to estimate local density, method of mapping density to number of neighbours, maximum number of neighbours to assign to any node. The development of a more robust method for adjusting the number of neighbours in network construction would allow a more flexible structure and more informative network. Moreover, the development of a truly dynamic thresholding approach that adjusted local thresholds without relying on a KNN approach and excluded or isolated outliers within the data automatically (similar to DBSCAN clustering Ester et al. (1996)) would offer increased flexibility in network sparsification.

Hierarchical clustering presents another area for exploration. The differences in performance of my clustering algorithms across data distributions and cluster configurations highlight how the suitability of different methods depend on both cluster number and structure, which are often unknown *a priori*. Hierarchical clustering has the potential to mitigate this variability by capturing clusters at different scales. Dendrograms, which represent the hierarchical clustering structure, can be cut at various scales and provide more interpretability over flat clustering algorithms. An intriguing avenue for further exploration would be to explore network construction for hierarchical clusters. This would require hierarchical data generation and the development of metrics to compare the accuracy of produced dendrograms.

Chapter 3

Multi-Modal Integration

3.1 Introduction

In the digital age, we are witnessing an influx of diverse data in various forms, presenting both exciting opportunities and significant challenges Jordan and Mitchell (2015); Mirza et al. (2019). Effectively incorporating and utilising this wealth of data is a complex task, given the different properties and challenges associated with various data types.

In the field of machine learning, a key challenge is the integration of these various modalities, especially when dealing with general data forms like images or text. Approaches to integration varies depending on the specific application. For example, in supervised learning pipelines, it is common to use independent neural networks to generate embeddings for each data modality, which are then processed together for the specific task Baltrušaitis, Ahuja, and Morency (2018). The need to handle modalities separately, such as text or image input, arises from the effectiveness of specialised models such as transformers for text Vaswani et al. (2017) and convolutional neural networks for images Krizhevsky, Sutskever, and Hinton (2012). This differentiation has led to the emergence of multi-view learning, a field dedicated to addressing the integration challenges posed by modalities with diverse properties.

Biomedical data poses distinctive challenges due to its multi-modal nature, encompassing various forms ranging from high-dimensional multi-omic data capturing genetic information facets like RNA gene expression, DNA methylation sites, and copy number variants, to diverse medical data types, including images and clinical information derived from diagnostic questionnaires Acosta et al. (2022); Santiago-Rodriguez and Hollister (2021). This data exhibits a characteristic combination of a small number of observations and high dimensionality. In addressing this unique landscape, one particularly successful approach is the utilisation of similarity networks for multi-view learning.

By extracting the relationships within datasets using similarity measures, similarity network approaches are effective in overcoming the challenges posed by the high ratio of features to observations, allowing for a nuanced exploration of specific biomedical applications. What sets similarity networks apart is their high interpretability and adaptability, serving as versatile

data structures suitable for both unsupervised and supervised tasks. These tasks range from community detection to node/edge prediction, making similarity networks a valuable tool for uncovering insights in biomedical data Fortunato and Hric (2016); Su, Tong, Zhu, Cui, and Wang (2018).

The application of network integration techniques has witnessed extensive use in disease subtyping across various biomedical domains. Examples include cancer Verhaak et al. (2010), diabetes L. Li et al. (2015), and Parkinson's disease Markello et al. (2021). These approaches showcase the efficacy of similarity networks and network integration methods in unraveling complex patterns within biomedical data, offering insights for disease understanding and classification.

One of the most successful and widely adopted techniques for constructing multi-modal similarity networks is Similarity Network Fusion (SNF) B. Wang et al. (2014). Initially designed for cancer subtype detection, SNF employs a diffusion process to amalgamate similarity networks from different modalities and has been applied to many disease subtyping problems. Despite its success, the original assessment metrics used to evaluate SNF's performance were not conventional clustering accuracy measures like the Adjusted Rand Index (ARI) or Adjusted Mutual Information (Section 1.5.2). Instead, indirect metrics, such as differences in survival rates between clusters and the number of significant genes within clusters, were employed. Similar indirect evaluation metrics were later used for methods like NEighborhood based Multi-Omics clustering (NEMO) Rappoport and Shamir (2019). The primary challenge in these evaluations was the absence of data with known ground truth clusters across modalities. SNF is a relatively complex method of combining networks that relies on the neighbours of node being consistent across modalities. Consequently, determining the optimal conditions for SNF, or when simpler methods like mean similarity are sufficient, remained unclear. Notably, a study employing formal measures such as ARI found that mean similarity consistently outperformed SNF[1] Mitra et al. (2020).

To address this ambiguity, I introduce a framework for generating multi-modal data with straightforward variations in distribution and embedded cluster information. These variations enable us to assess how differences in the consistency of individual similarities across modalities impact the community detection performance of integration methods. This framework offers a systematic approach to evaluating the effectiveness of various integration methods under controlled conditions, shedding light on the circumstances where simpler methods may suffice or where the complexity of SNF proves advantageous.

1. Section 1.3 provides a more comprehensive discussion with certain caveats to this performance

A notable characteristic of biomedical data is the presence of partially complete modalities. In multi-modal datasets, it is rare to have a complete set of measurements for all individuals. *Unit non-response*, where individuals have no measured features, is a frequent occurrence. In uni-modal analyses, these individuals are typically excluded from the study. However, in multi-modal data, this practice can result in significant data wastage, leading to the exclusion of large numbers of individuals or entire modalities to maximise observations. Similarity networks are well suited to mitigating data wastage by incorporating partial data.

NEMO was developed to address this issue, aiming to incorporate partial data by calculating the relative similarity of individuals based on their shared data. Similarly, SNF can be extended to incorporate partial data by imputing similarity values between individuals absent and present in each modality. While partial multi-view data is well-studied, the reasons for partial data are not extensively explored. Individuals are assumed to be partial at random yet the reasons for partial measurement can be complex Nakagawa and Freckleton (2008). For instance, certain diagnostic questionnaires are based on the severity of a condition, resulting in non-random partial data that is related to the objects of interest Gotham et al. (2007). In this chapter, I delve into the impact of partial data, both at random and not at random. I aim to evaluate which integration methods effectively incorporate partial data, shedding light on their performance under different conditions and contributing to a deeper understanding of their applicability in real-world scenarios.

In summary, my aim is to demonstrate the effect of similarity integration approach on community detection performance of constructed networks. A challenge in the evaluation of network construction is the lack of datasets with known ground truth community structure and data properties in each modality. To overcome this, I use multiple instances of synthetic data to allow an exploration of different levels of noise and consistency across modalities. I evaluate several network community detection methods across a range of different cluster settings. Section 3.2 describes the similarity integration methods used in the construction of similarity networks. Section 3.3 describes my multi-modal data generation framework, how I embed different cluster information, the different data distributions used and my approach to creating partial data. In Section 3.4, I introduce the experiments before showing and discussing the results produced in Sections 3.5 and 3.6.

3.2 Multi-Modal Similarity

In this chapter, I want to evaluate the quality of the network produced from multi-modal data by similarity integration methods, specifically when applied to community detection. As discussed in Section 1.3, Similarity integration methods can be categorised as early, intermediate and late. My aim in this chapter is evaluating and identifying which integration methods are most beneficial in the specific context of community detection. I consider five integration approaches

- **Early**
  - **Concatenated** X_i — All modalities are combined into a single feature matrix. Pairwise similarity and network sparsification are subsequently performed.
- **Intermediate**
  - **Mean** S_i — Mean similarity between a pair of nodes i and j across all modalities.
  - **Extreme Mean** — Mean "extreme" similarity/dissimilarity between a pair of nodes i and j across all modalities. For each modality, pairwise similarity is thresholded to only include very similar and very dissimilar connections.
- **Late**
  - **Similarity Network Fusion (SNF)** — *de facto* standard approach for multi-omic integration and unsupervised clustering analysis. Similarity calculated through diffusion across KNN graphs.
  - **NEighborhood Based Multi-Omic Clustering (NEMO)** — Mean relative similarity between nodes i and j based on a K-nearest Neighbourhood in each modality.

3.2.1 Concatenating Features

The simplest approach to integrating multi-modal data is to concatenate the features from each modality into a single "master" feature matrix. This is the prototypical example of early integration. From a set of m modality feature matrices X_i, a single data feature matrix X is constructed as follows:

$$X = [X_1, X_2, \ldots, X_m] \tag{3.1}$$

The benefit of this approach lies in its simplicity and the unadjusted inclusion of each modality. If the features of any particular modality are informative, this should be captured through the similarity calculation. In practice, modalities can have highly different scales of dimensionality — from clinical data with tens of features to DNA methylation data with hundreds of thousands of features Tomczak et al. (2015). In such cases, the higher dimensional modality will dominate the differences between individuals.

3.2.2 Mean Similarity

Perhaps the simplest method of multi-modal similarity integration beyond feature concatenation is to calculate similarity for each modality independently and integrate the similarity scores before constructing a final network G. The pairwise similarity matrix[2] P is given by

$$P_{\text{Mean}} = \frac{1}{m}\sum_{v=1}^{m} P^{(v)} \tag{3.2}$$

where m is the number of modalities and $P^{(v)}$ is the pairwise similarity for modality v. This intermediate integration technique involves calculating the pairwise distances independently for each modality and then merging them to form a single pairwise similarity matrix. From this final pairwise similarity matrix, a network can be created. The benefit of this approach is the ability to process each modality independently. For example, similarity within each modality can be calculated using separate similarity measures. Each measurement of similarity is equally valued, ensuring that modalities with lower dimensionality will not be obscured by modalities with a high number of features.

3.2.3 Extreme Mean

Typically, the focus in community detection is on extreme similarities or dissimilarities, as relationships between nodes that are mildly similar or dissimilar are often considered uninformative in network construction. These less informative relations are usually filtered out during the sparsification process. Connections between nodes that exhibit high similarity form communities within the network. However, in realistic scenarios such as disease analysis, negative relations (high dissimilarity) can be crucial. The distinct dissimilarity between two individuals in a subset of features or measurements can provide strong evidence that these individuals are not alike and likely do not share the same disease or disease subtype. Connecting these dissimilar individuals in network construction would be inaccurate. One approach to creating a network based on strong (dis)similarities is to threshold the similarities of each modality before integration.

To threshold a modality's pairwise similarity matrix P, we apply the following rule:

$$W(i,j) = \begin{cases} P(i,j), & \text{if } |P(i,j)| > \sigma \\ 0, & \text{if } |P(i,j)| < \sigma \end{cases} \tag{3.3}$$

2. This notation is used to keep consistency with the original notation used in deriving the SNF method. With respect to previous chapters, the corresponding notation is $P^{(v)} = S_v$

where σ is chosen threshold value. This thresholding process can be straightforwardly applied to normalised similarity metrics, such as Pearson correlation. After thresholding, only highly positively correlated or negatively correlated relationships will be retained. For unnormalised metrics like Euclidean distance, one can normalise the pairwise distances to obtain a zero mean, identity standard deviation distribution of pairwise distances, making it easier to choose an interpretable threshold value. In this work, a threshold of $\sigma = 1$ standard deviation is used, ensuring that only pairwise values that are significantly similar or dissimilar are retained.

To obtain the final pairwise similarity matrix (P_{Extr}), we compute the mean of the per-modality thresholded pairwise similarities:

$$P_{\text{Extr}} = \frac{1}{m} \sum_{v=1}^{m} W^{(v)} \tag{3.4}$$

This method can be considered as both an intermediate and late integration method. The threshold step is akin to threshold sparsification discussed in Section 2.2.1, making it a form of integration of weighted threshold networks. Unlike threshold sparsification, this method retains highly dissimilar connections, and the resulting weighted pairwise similarities after thresholding do not represent a typical threshold network as described in Chapter 2.

3.2.4 Similarity Network Fusion

Similarity Network Fusion (SNF) B. Wang et al. (2014) is a late integration approach for constructing a multi-modal similarity network. SNF employs an iterative diffusion process to converge on a single pairwise similarity matrix. The primary goal of SNF is to update the similarity between two nodes (in any given modality) based on the similarity of their shared nearest neighbours across all modalities. It can be thought of as creating a weighted K-nearest neighbour (KNN) network for each modality and merging these networks by adjusting the weights between nodes based on their shared neighbours. Therefore, SNF is a late integration method that combines networks in a non-linear manner.

In B. Wang et al. (2014), Wang *et al.* introduce the two key components of SNF: i) a scaled exponential similarity kernel to compute affinity (similarity) between all nodes on each modality and ii) a diffusion process to merge the similarity for separate modalities. The diffusion step was a key contribution for modality integration and the method was validated through the identification of cancer subtypes on multi-omic cancer data from The Cancer Genome Atlas (TCGA).

The pairwise scaled exponential similarity kernel computed between all nodes is given by

$$W(i,j) = \exp\left(-\frac{d^2(x_i,x_j)}{\mu\varepsilon_{i,j}}\right) \tag{3.5}$$

where $d(x_i, x_j)$ is a distance metric (in the original paper, the euclidean distance was used), μ is a hyperparameter that controls which distances can be considered highly similar (for a fixed distance between nodes i and j, lowering mu lowers their similarity), and $\varepsilon_{i,j}$ is a scaling factor that incorporates the distance between the nearest neighbours of i and j. The scaling factor $\varepsilon_{i,j}$ is given by

$$\varepsilon_{i,j} = \frac{\text{mean}(d(x_i, N_i) + \text{mean}(d(x_j, N_j) + d(x_i, x_j)}{3} \tag{3.6}$$

where N_i is the set of neighbours of node i. This scaled similarity kernel is qualitatively different to the euclidean distance. It is normalised between 0 and 1 and more importantly, computed values between nodes are typically either quite close to 1 or quite close to 0. Moreover, the scaled affinity kernel controls for different areas of density in the feature space. For a node x_i, if $d(x_i, N_i)$ is large on average then $d(x_i, x_j)$ being large will not be as penalised. In other words, if a cluster is more spread apart, the pairwise similarity of its nodes will be as high as a more tightly knit cluster.

There are a number of key normalisations that are performed before the diffusion process to ensure numerical stability. A row normalised weighed pairwise affinity for each modality P is created by

$$P(i,j) = \begin{cases} \frac{W(i,j)}{2 * \sum_{k \neq i} W(i,k)}, & j \neq i \\ 1/2, & j = i \end{cases} \tag{3.7}$$

which ensures that $\sum_j P(i,j) = 1$. A normalised weighted KNN network with adjacency matrix[3] S is created for each modality

$$S(i,j) = \begin{cases} \frac{W(i,j)}{\sum_{k \in N_i} W(i,k)}, & j \in N_i \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

where N_i is the set of neighbours of node i. While a KNN network is created at this step, the matrix S is non-symmetric and reflects the adjacency matrix of a directed network, not an undirected network (see Section 1.1.

To integrate the modality together diffusion is performed across the KNN networks using the similarity in other modalities. The diffusion step for modality v is given by

$$P^{(v)} = S^{(v)} \times \left(\frac{\sum_{u \neq v} P^{(u)}}{m-1} \right) \times (S^{(v)})^T, v = 1, 2, \ldots, m \tag{3.9}$$

3. Under the notation in Chapter 1, $S = A$.

The similarity $P^{(v)}$ for each modality v is updated by considering the similarity of the local neighbourhood of nodes i and j in other modalities (the only non-zero elements in $S^{(v)}$ are the nearest neighbours of a node).

This can be more clearly seen if we express the update rule from iteration t to $t+1$ as

$$P^{(v)}_{t+1}(i,j) = \sum_{k \in N_{v_i}} \sum_{l \in N_{v_j}} S^{(v)}(i,k) \times S^{(v)}(j,l) \times \left(\frac{\sum_{u \neq v} P^{(u)}_t(k,l)}{m-1} \right) \tag{3.10}$$

If the neighbours of node i and j are highly similar in other modalities ($P^{(u)}_t(k,l)$), then $P^{(v)}(i,j)$ will increase and the edge (i,j) is more likely to be included in the final network. Conversely, if the neighbours of node i and j are very unsimilar, this reduces the evidence of a relationship between nodes i and j. $P^{(v)}(i,j)$ will decrease, making it less likely the edge (i,j) will be included in the network.

After each iteration the modalities are re-normalised (for numerical stability) until convergence is achieved. The final network pairwise similarity is given by

$$P_{\text{SNF}} = \frac{1}{m} \sum_{v=1}^{m} P^{(v)} \tag{3.11}$$

Typically, very few iterations are required. The authors found only 1 or 2 needed to converge on the TCGA data.

SNF was developed on and applied to multi-omic cancer sets with no known ground truth clusters. As a result, the accuracy of the method had to be evaluated indirectly. Differences in the survival rate of the subtypes and the number of significant genes present in each subtype were used as evidence of the success of the method. It was not evaluated with typical clustering metrics such as ARI or AMI. There still remains unanswered questions over what types data the method is best suited to? Does SNF always outperform simpler approaches such as Mean S_i?

3.2.5 NEMO

NEMO (NEighborhood Based Multi-Omics clustering) is an alternative approach to similarity integration Rappoport and Shamir (2019). Similar to SNF, it is a late integration approach that combines networks from each modality. It is a simpler approach than SNF. NEMO does not make use of diffusion and instead integrates a nodes neighbourhood information by creating a KNN network on each modality. A final network is created by computing a weighted sum of the individual networks. Similar to SNF, NEMO is a late integration method that combines networks rather than pairwise distances in order to create a final network.

Initially for each modality, a KNN network is created from their pairwise similarity matrix and the relative similarity between nodes is calculated using

$$S^{(v)}(i,j) = \frac{W^{(v)}(i,j)}{\sum_{r \in N_{v_i}} W^{(v)}(i,r)} \cdot I(j \in N_{v_i}) + \frac{W^{(v)}(i,j)}{\sum_{r \in N_{v_j}} W^{(v)}(r,j)} \cdot I(i \in N_{v_j}) \tag{3.12}$$

where N_{v_i} is the set of neighbours of node i in modality v, $W^{(v)}$ is the pairwise affinity between nodes. Similar to SNF, NEMO makes use of the scaled exponential affinity kernel (Eq. 3.5).

Finally the average similarity is calculated using

$$P_{\text{NEMO}} = \frac{1}{m} \sum_{v=1}^{m} S^{(v)} \tag{3.13}$$

For partial data the average relative similarity is adjusted to only take the mean of the modalities where both nodes are present

$$P_{\text{NEMO}} = \frac{1}{|\sigma_{ij}|} \sum_{v \in \sigma_{ij}} S^{(v)}(i,j) \tag{3.14}$$

where σ_{ij} is the set of modalities where i and j are both present. In other words, the similarity between two nodes is calculated ignoring missing values and does not "punish" the similarity between two nodes if they have modalities where they are not present. Unlike other approaches the increased uncertainty in similarity between nodes with missing modalities does result in a reduced level of similarity.

3.3 Synthetic Multi-Modal Data

As discussed in Chapter 2, a challenge in evaluating the quality and suitability of constructed networks is a lack of data with known ground truth clusters. This further extends to multi-modal data. As an example, Similarity Network Fusion (SNF) was first proposed on cancer data sets from The Cancer Genome Atlas (TCGA) with unknown cancer subtypes. As a result, the assessment of predicted clusters had to rely on differences in survival rates and the number of significant gene mutations[4]. A key challenge in assessing multi-modal integration lies in the ambiguity surrounding the scenarios where different methods excel. Understanding the specific situations and data types to which each method is best suited is crucial for accurate community detection. Do certain methods perform well in noisy data with inconsistencies

4. Refer to Section 1.3 for an in-depth discussion

between the modalities? Are others more suited to "simpler" problems where modalities are in agreement and the data is consistent? To explore such questions, we need data where the embedded community structure in the feature space of each modality is known. Synthetic data enables us to generate such datasets and explore these types of questions .

A key assumption in multi-modal data analysis is that each modality captures a different aspect or view of the underlying community structure. Another assumption is that different modalities have different distributions and properties that require individual processing. Without this there are few reasons not to combine our data into a single modality to simplify processing and analysis. In biological multi-omic analysis, we typically encounter an additional factor. The dimensionality of certain modalities e.g. gene expression, varies significantly, often having differences in magnitudes that risk overshadowing the information contained in other modalities. The sheer scale of these dimensions necessitates separate consideration to ensure that the insights and patterns within each modality are not obscured by the dominance of a particular high-dimensional modality. With these factors in mind, a "good" data generator will allow the adjustment of both the underlying data distribution and the embedded cluster information of each modality.

In this work, I propose a framework for the generation of high-dimensional data where the distribution and cluster information in each modality can be adjusted separately. Each individual modality's clustering problem is non-trivial and the performance of different multi-modal similarity integration techniques can explored in detail. The data generation method proposed here scales to a high number of modalities. My generation procedure is as follows i) generate ground truth set of clusters y, ii) for each modality i, I generate the modality cluster ground truth y_i derived from y and iii) for each modality i, I generate the data X_i from the modality clusters y_i. Unlike Chapter 2 where I allow unequal cluster sizes, I split the population of nodes N into clusters of equal size. While less realistic than the cluster settings evaluated in Chapter 2, the equal sizes allow improved evaluation by isolating the effect of changes in modality distribution and cluster information. It should be noted that this generation procedure facilitates clusters of any size.

As shown in Figure 3.1, I can adjust both the distribution of the data X_i and the method of generating the modality clusters y_i from y. I want each modality to capture a different aspect of the community structure. To replicate this in my dataset, I require a method of adjusting the embedded clusters in each modality while still ensuring a ground true community structure across the entire dataset. My proposed solution is to sample a set of ground truth cluster labels y for all modalities and generate per modality labels y_i from y that control how individuals are distributed across embedded clusters within a particular modality. I want these embedded

clusters to differ from the ground truth while remaining consistent, for example, I embed 3 clusters in X_0 but ensure each cluster in y_0 is assembled from the 5 clusters in y. I use the framework introduced in Section 2.3 to adjust the data distribution of the features (X_i) of the clusters in each modality (y_i).

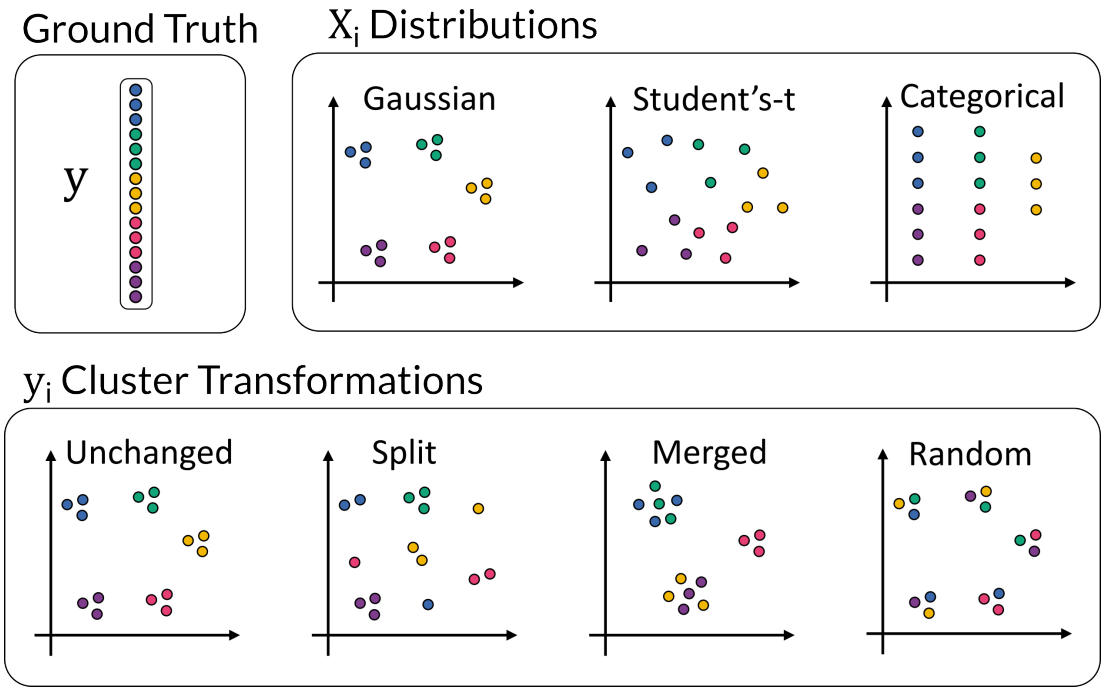

Figure 3.1: Generation of Modality-Specific Clusters and Feature Distributions. This figure illustrates the possible components that can be adjusted in the process of generating modality-specific clusters and features from the ground truth labels y. For each modality i, the modality ground truth clusters y_i are derived by applying one of four transformations to y: (i) keeping y_i identical to y, (ii) splitting clusters in y into subclusters, (iii) merging clusters in y, or (iv) generating random, unrelated clusters. Features X_i are then generated based on y_i using one of three distributions: (i) mixture of Gaussians, (ii) mixture of Student's-t, or (iii) categorical data.

3.3.1 Distributions

I propose using three types of cluster distributions as shown in Figure 3.1

- Mixture of Gaussians,
- Mixture of Student's-t,
- Categorical Data.

A detailed discussion of the generation of these three distributions can be found in Section 2.3. The key properties these three distributions allow us to assess are i) levels of noise in the dataset and ii) the consistency of inter and intra cluster distances across modalities.

The Mixture of Gaussians is a commonly assessed distribution with implanted community structure. Members of each individual cluster are sampled from separate high dimensional Gaussian distributions. I assign each cluster identity covariance so that the sole difference between clusters are the locations of the center of each distribution. Cluster centers are generated in such as way as to ensure some level of overlap between clusters at higher dimensions. This overlap prevents trivial detection of clusters and ensures a challenging cluster problem in each modality.

The Mixture of Student's-ts is a noisier, more challenging variant of the Mixture of Gaussian distributed data. Again members of each individual cluster are sampled from separate high dimensional distributions that differ only in the location of their cluster center. I sample from high dimensional Student's-t distributions with 2 degrees of freedom, identity covariance and unique centers for each cluster. Student's-t distributed data is far noisier than Gaussian distributed data. Due to the heavy tail of the Student's-t distribution, outliers are far more likely and the level of overlap between clusters is increased. This clustering problem is far more challenging and for my multi-modal data the variance of within clusters distances from one modality to next is significantly increased.

The categorical data distribution is comprised of mix of informative and uninformative features. In an informative feature, each cluster has an individual probability distribution across n possible categories and the value of each member of the cluster is sampled according to that distribution. In an uninformative feature each individual in the dataset samples according to a shared distribution. This distribution is highly consistent from modality to modality. The discrete number of possible values each cluster can take ensures that outliers are highly unlikely and within clusters distances do not vary significantly from one modality to the next. A more detailed description of this data distribution can be found in Section 2.3.

A key difference for this distribution is that the euclidean metric is unlikely to be an optimal choice of metric for ranking and measuring dissimilarity between nodes. For the other distributions, the greatest difference between clusters in the mixture of Gaussians and Student's-ts is the euclidean distance between the cluster centers. Euclidean distance is highly suited to measuring similarity for these distributions. In the categorical data, the distribution over categories is random and so the features are not ordinal (at least within a cluster). It is possible for a cluster to be equally weighted to the lowest and highest levels of the categories. The within cluster euclidean distance is not guaranteed to be lower than the intra cluster distances.

3.3.2 Cluster Information

A key assumption of multi-modal and multi-omic data analysis is that the cluster or community information contained in each modality varies from one modality to the next. We are unlikely to encounter real world datasets where each modality captures data with identical cluster distributions (within each particular modality). In such settings individual analysis of only one particular modality might be more optimal. A far more common scenario is one where the distribution of clusters within each modality will vary from one modality to the next. In this data scenario it is more likely that the true cluster distribution can be identified by integrating (in some fashion) all modalities together. I want to generate data that allows us to evaluate both types of scenarios; i) where the cluster distributions changes from modality to modality and ii) where the cluster information is consistent.

To adjust the cluster information embedded in each modality I propose three methods of adjustment.

- Splitting y into sub-clusters
- Merging clusters in y into super-clusters
- Generating random set of clusters unrelated to y

In Figure 3.1, I illustrate these methods of adjusting y_i on a simple two dimensional example. My aim in making these adjustments is to evaluate the ability of similarity integration methods to handle inconsistencies across modalities. Splitting and merging y into sub and super clusters is most reflective of real world settings were possible subtypes or cell types likely have differences in some modalities but share traits in others. While the cluster distribution is not identical to the true cluster distribution it will still be quite consistent from one modality to the next on less noisy distributions.

By including a random modality where the cluster distribution y_i is unrelated to y, I am able to assess the ability of integration methods to not just handle noise within a dataset but also handle the inclusion of uninformative data. There is no guarantee in real world settings that all modalities will be informative for example a particular disease may affect an individuals transcriptomic data but not its genomic data and so the inclusion of genomic data only adds noise to the dataset. In particular, the random modality add significant inconsistencies for the middle/late integration methods as the inter cluster distances and KNN networks generated from random modalities are guaranteed to be unlike the ground truth data.

3.3.3 Generating Partial Data

A number of studies have analysed the problem of partially complete data and its affect on the accuracy of multi-modal methods S.-Y. Li et al. (2014); Rappoport and Shamir (2019); Xu et al. (2022). However, these methods have only analysed the possibility of data being partial at random and have not analysed the effect of partial data on network structure. I define partially complete data to refer to a multi-modal dataset with m modalities where a subset of the individuals have no recorded data in at least 1 of the m modalities and a complete set of measurements data in the other modalities. To reiterate this differs from a more typical missing data scenario where an much smaller proportion of individuals are missing values in a subset of the d features within a dataset.

I want to assess two types of partial data scenarios; i) where the data from each modality is absent at random i.e. each cluster is equally likely to be have entities with no data recorded and ii) where the data is missing from each modality due to its cluster membership i.e. one or more clusters have no data recorded in a modality. I restrict my partial data experiments to partial data where each individual has at most one non recorded modality.

I generate partial datasets as follows. Firstly a multi-modal dataset is generated (as described in Section 3.3. Then to create a partial dataset entities are removed from a modality either at random or based on their true cluster y. To create data partial at random, I randomly generate a set of labels y_{NaN} where each label is $y_{NaN_i} \in \{1, \ldots, m\}$ where m is the number of modalities in the dataset. Entities are removed from a modality based on their label in y_{NaN}. To create partial data based on cluster membership, I create a set of labels y_{NaN} by merging the clusters in y into m super-clusters (if $m < n_c$ where n_c is the number of clusters in y) or splitting y into m sub-clusters if ($m > n_c$). If $m = n_c$ then I set $y_{NaN} = y$. Again entities are removed from a modality based on their label in y_{NaN}.

3.4 Experiment Setup

I conduct three experiments. I evaluate i) the performance of my multi-modal integration methods on a variety of different modality problems, ii) the adaptability of the integration methods to an increasing number of modalities and iii) the ability of each integration method to incorporate partial modalities where a subset of individuals do not have features in some of the modalities.

3.4.1 Integration Methods

As described in Section 3.2, I evaluate several multi-modal similarity integration approaches.

- **Similarity Network Fusion (SNF)** — *de facto* standard approach for multi-omic integration and unsupervised clustering analysis. Similarity calculated through diffusion across KNN graphs.
- **NEighborhood Based Multi-Omic Clustering (NEMO)** — Mean relative similarity between nodes i and j based on a K-nearest Neighbourhood in each modality.
- **Mean** S_i — Mean similarity between a pair of nodes i and j across all modalities.
- **Extreme Mean** — Mean "extreme" similarity/dissimilarity between a pair of nodes i and j across all modalities. for each modality, pairwise similarity is thresholded to only include very similar and very dissimilar connections.
- **Concatenated** X_i — All modalities are combined into a single feature matrix. Pairwise similarity and network sparsification are subsequently performed.

I make use of a python implementation of *SNF*, the `snfpy`[5] package. I use custom python implementations for the *NEMO*, *Mean* S_i, *Extreme Mean* and *Concatenated* X_i similarity integrators. The final network produced by all similarity integration methods is created by constructing a K-nearest Neighbour (KNN) Graph[6] with $K = 25$. For the SNF affinity function (Eq. 3.5), I use the original proposed hyperparameter settings for the SNF Kernel $\mu = 0.5$ and a value $K = 25$ to match the final KNN graph. Unless otherwise specified in the creation of the pairwise similarity matrix, I make use of raw distance for the *Concatenated* X_i and *Mean* S_i methods, and the SNF affinity function for *SNF*, *NEMO* and *Extreme Mean*.

3.4.2 Clustering Algorithms

I perform community detection on the multi-modal graph networks using three distinct network clustering algorithms[7]

- **SBM** — Micro-canonical Stochastic Block Model Peixoto (2018). Python `graph-tool`[8] implementation Peixoto (2014).
- **Leiden** — Modularity maximisation using Leiden algorithm Traag et al. (2019). Python `igraph`[9] implementation Csardi and Nepusz (2006). The resolution hyperparameter is selected using event sampling Jeub et al. (2018).

---

5. — v0.2.2
6. Data with 2500 individuals is evaluated. $K = 25$ was found to produce networks of a desirable density based on the results of Chapter 2.
7. See Section 1.5.1 for detailed discussion.
8. v2.45
9. v0.10.3

- **Spectral** — Spectral decomposition and K-means clustering of "Random Walk" normalised Laplacian $L_{rw} = I - D^{-1}A$. Python `spectralclusterer`[10] implementation Q. Wang et al. (2018).

I evaluate the quality of the networks produced by the similarity integration methods using the following network statistics

- **Modularity** y — Network modularity compares the observed number of edges within a set of clusters to the number of edges expected under a null model (node degrees are fixed and edges are placed at random). The modularity of a graph G is given by

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

  where m is the number of edges in G, A is the adjacency matrix of G, k_i is the degree of node i and C_i is the cluster that node i belongs to. I calculate the modularity of the ground truth clusters y.
- **Triad participation ratio (TPR)** y — is the fraction of nodes in cluster C that belong in a triad,

$$f(C) = \frac{|\{u : u \in C, \{(v,w) : v,w \in C, (u,v) \in E, (u,w) \in E, (v,w) \in E\} \neq \emptyset\}|}{n_c}$$

  where n_c is the number of nodes in cluster c. I calculate the average TPR of the ground truth clusters y.
- **Assortativity** — the Pearson correlation coefficient of degree between pairs of nodes with an edge connecting them. It measures the propensity for edges to exist between nodes of similar degrees. An explicit definition for degree assortativity can be found in Eq. 21 in M. E. Newman (2003). Values range between $[-1, 1]$. A positive value indicates that nodes of similar degree connect. A negative value indicates that high degree nodes are more likely to connect to low degree nodes.
- **Mean Path Length** — Mean length of shortest paths between all pairs of nodes in G.
- **Mean degree** k — Mean degree k of all nodes in G. The degree of node i is the number of edges of between node i and other nodes in G.
- **Median degree** k — Median degree k of all nodes in G.

Modularity and TPR provides a measure of how well the true community structure has been embedded in G. Modularity assesses how tightly knit the communities in G are. Communities with higher modularity have higher internal density and are characterised by more connections within the community than connections with nodes outside it.

10. v0.2.16

TPR measures the fraction of nodes within a community that form triads. This metric draws inspiration from observations on social networks that within communities, friends of a friend tend to be common. In other words, if two individuals share a common friend, they are more likely to be friends themselves. A good community can therefore be defined to be one that contains many such friends of friends. Again this is a measure of internal density — the more triads in a community, the denser the internal connectivity. Yang *et al.* J. Yang and Leskovec (2012) showed that TPR is a reliable metric for detecting communities in real networks.

Assortativity, mean path length, mean and median degree capture distinct facets of the global structure within a network. Mean and median degree provide a summary of the degree distribution — whether the degree distribution is skewed, whether there is an abundance of high or low degree nodes. Assortativity provides insight into the type of nodes that tend to interconnect. This metric helps identify whether nodes with similar degrees are more likely to be linked.

Mean path length, on the other hand, offers insight into the connectedness of communities within the network. All my networks are constructed on the same data with the same KNN hyperparameter and should have similar numbers of edges. Differences in path length are due to differences in how edges connect the global structure. A low mean path length signifies high interconnectivity between communities drawing nodes in distinct communities together. Conversely, a high mean path length suggests fewer inter-community edges and longer distances between nodes, implying greater isolation between communities. These metrics collectively provide a comprehensive overview of a network's structural characteristics.

For a more detailed exploration of these network properties, please refer to Section 1.1, where these metrics and their implications are discussed in greater detail.

3.4.3 Integration Method Performance

To evaluate the similarity integration methods, I desire a mixture of various modality problems. I want to see both the effect of type of distribution of the clusters as well as the effect of differences in the cluster information present in each modality. *How well do the methods handle noisy data? How well do the methods handle inconsistencies in inter and intra cluster distances from modality to modality?* I compare the performance of similarity integration methods on datasets comprised of three generated modalities. This is reflective of real world applications where a large number of modalities are not typically available/collected. For example, the TCGA multi-omic datasets used in B. Wang et al. (2014) were comprised of mRNA expression, DNA methylation and miRNA expression data.

Table 3.1 shows the set of fifteen problems used to evaluate the performance of the integration methods. The settings are listed by order of average ARI performance of SBM and Leiden cluster methods on each individual modality[11]. I briefly describe some of the key modality problems: *Easy* can be thought of as the default setting — a mixture of Gaussians generated with unchanged cluster information per modality. *Cat* evaluates the effect of categorical distributed data. *Noisy* evaluates a high noise setting. *Split* evaluates settings with ground truth clusters broken apart across modalities. *Merged* evaluates the converse where ground truth clusters are combined together. *1Rand* evaluates the addition of a discordant and uninformative modality. *Mixed Normal* and *Mixed Noisy* evaluate effect of sets of modalities with differences in cluster information in low and high noise settings respectively. *Mixed Noisy 1Rand* is the most challenging setting where each modality has high noise and one of the modalities is uninformative.

For each modality problem, I split 2500 entities into 10 equal clusters. This number was chosen to maintain consistency with Chapter 2, where 2,500 entities were used to evaluate sparsification methods, thereby increasing confidence in the choice of sparsification. As discussed in Section **??**, using equally sized clusters helps isolate the effects of integration and ensures greater consistency across generated modalities. A crucial step in the process is the merging and splitting of clusters during data generation. When clusters are unequally sized, the resulting data can be less consistent. For instance, a modality where two large clusters are merged will be qualitatively different from one where two smaller clusters are merged. The number of equally sized clusters, ten, was chosen as it strikes a balance between the number of clusters where the three clustering algorithms perform best. Each generated modality contains 50 features i.e. each X_i is a $N \times d$, 2500 x 50, matrix with the distribution and cluster information as described in Table 3.1. The merged clusters are created by randomly merging y into 5 clusters. The merging is done at random and can be unequal e.g 6 clusters in y merged into 1 cluster in y_i with the remaining 4 clusters unchanged. The split clusters are created by splitting the clusters in y into 20 clusters. Similar to the merged clusters this process is done at random and can be unequal e.g. one cluster in y split into 11 subclusters with the remaining 9 unchanged. A random modality is created by generating assigning each entities to one of 10 equally sized clusters at random and generating the dataset with the random labelling y_i. Each modality problem is evaluated on 20 instances to better estimate the accuracy of the similarity integration methods.

I repeat the evaluation using two metrics i) euclidean distance and ii) correlation. I evaluate using both the raw distance and using SNF affinity (Eq 3.5).

11. Spectral clustering is not included in this ranking due to its high variability.

Name	X_1	X_2	X_3
Categorical	C-0	C-0	C-0
Easy	G-0	G-0	G-0
Single Merged	G-0	G-0	G-1
Single Noisy	G-0	G-0	S-0
Split	G-2	G-2	G-2
Mixed Normal	G-1	G-1	G-2
Merged	G-1	G-1	G-1
Mixed All	C-1	G-1	S-2
Noisy	S-0	S-0	S-0
1Rand	G-0	G-0	G-3
Mixed Noisy	S-1	S-1	S-2
Mixed 1Rand	G-1	G-2	G-3
Noisy 1Rand	S-0	S-0	S-3
Mixed Noisy 1Rand	S-1	S-2	S-3
2Rand	G-0	G-3	G-3

Distributions
G — Gaussian
S — Student's-t
C — Categorical

y_i clusters
0 — Unchanged
1 — Merged
2 — Split
3 — Random

Table 3.1: Modularity Problems for Evaluating Similarity Integration Methods. 2500 samples are split into 10 equally sized clusters, with three modalities are generated for each modality problem. Each modality X_i is characterised by a distribution — Gaussian (G), Student's-t (S), or Categorical (C) — and by cluster information: 0) y_i identical to the ground truth y, 1) y_i with 5 clusters merged from y, 2) y_i produced by splitting y into 20 sub-clusters, and 3) y_i containing 10 random, equally sized clusters unrelated to y. These variations allow for a comprehensive assessment of how well similarity integration methods can handle different cluster structures and data distributions.

3.4.4 Number of Modalities

To evaluate ability of different similarity integration methods to handle increasing number of modalities, I create modality problems by randomly sampling the distribution and cluster information of X_i and y_i. I consider 6 different sets of distributions and cluster types. *Easy* and *Noisy* are consistent modalities where the clusters are identical to y and the distributions do not change from mixtures of Gaussians and Student's-ts respectively. Again for the *All* modality problem, I keep the clusters unchanged but additionally randomly sample a distribution from the three possible types Gaussian, Student's-t and Categorical. In *MergeSplit* I only use mixture of Gaussians distributed data but randomly merge or split the clusters in y into 5 or 20 clusters in y_i (the same settings used in the original modality problems in Section 3.4.3). In *Mixture* I only use mixture of Gaussian distributed data but allow any of the possible types of cluster information; unchanged, merged, split or random. *Mixture* introduces random unrelated modalities that add noise to the set of distances between nodes i and j, $\{S_{ij}^{(k)} : k \in 1,\ldots,m\}$. The final modality problem I consider is *All*. In this setting any combination of distribution and cluster information are possible.

I evaluate the similarity integration methods on $m \in \{3,5,10,15,20,30,40\}$ modalities. For each number of modalities, I evaluate 5 instances of the data. I use the euclidean metric in my evaluations.

Name	Distribution	y_i
Easy	$d_i \in [G]$	$c_i \in [0]$
Noisy	$d_i \in [S]$	$c_i \in [0]$
All	$d_i \in [G,S,C]$	$c_i \in [0]$
MergeSplit	$d_i \in [G]$	$c_i \in [1,2]$
Mixture	$d_i \in [G]$	$c_i \in [0,1,2,3]$
Any	$d_i \in [G,S,C]$	$c_i \in [0,1,2,3]$

Table 3.2: Modularity Settings for Testing Integration Methods Across Increasing Modalities. Set of modularity settings used to explore ability of integration methods to scale with the number of modalities. For each modality, a distribution d_i is randomly selected for the features X_i, and a cluster transformation c_i is applied to the ground truth labels y to produce y_i. The labels for the distributions and transformations align with those detailed in Table 3.1. These settings are specifically designed to challenge the ability of integration methods to maintain performance as the number of modalities increases.

3.4.5 Partial Modalities

I evaluate the effect of partial modality on the similarity integration methods. To include entities with missing modalities, I adapt the methods in the following ways

- **Similarity Network Fusion (SNF)** — For each pairwise modality distance $S^{(K)}$, the pairwise value between a node i with NaN in X_k and any node j is set to max distance/dissimilarity for that modality. SNF is then computed as normal with max dissimilarity included.
- **NEighborhood Based Multi-Omic Clustering (NEMO)** — NEMO was developed to analyse partial data. The mean relative similarity for any pair of nodes i and j is computed over the modalities where both nodes have recorded data.
- **Concatenated** X_i — Feature mean value imputation in X_k for all nodes with NaN values. Then distance/similarity calculated as normal.
- **Mean** S_i **imputing *Max*** — For each pairwise modality distance $S^{(K)}$, the pairwise value between a node i with NaN in X_k and any node j is set to max distance/dissimilarity for that modality. Mean similarity then computed between a pair of nodes i and j across all modalities.
- **Mean** S_i **ignoring** NaN — The mean similarity for any pair of nodes i and j is computed over the modalities where both nodes have recorded data.
- **Extreme Mean** — Thresholding is performed on the pairwise similarity between nodes with recorded values in the modality. The mean similarity for any pair of nodes i and j is computed over the modalities where both nodes have recorded data. If all values between i and j are NaN after thresholding (including NaN for where i has no recorded data in a modality) then the dissimilarity is set to max.

It is important to note how the choice of NaN imputation will affect the similarity integration methods. The KNN step of *SNF* is unlikely to include individuals with partial modalities due to the max dissimilarity imputation. Partial individuals will significantly alter the diffusion step and will require extremely high similarity in the nodes neighbours in other modalities to be included in the final network after KNN sparsification. *NEMO* is designed to incorporate partial data and places importance on relative similarity in other modalities. It is similar to *Mean* S_i *ignoring* NaN. For both of these methods a pair of entities with moderate similarity in all modalities will score lower than a pair of partial entities with high similarity in only one shared modality.

I make use of mean imputation in the features of *Concatenated* X_i. Here two individuals with no recorded values in a modality will score more similar as there will be no distance between the two in the features of that modality. *Mean* S_i *imputing Max* "punishes" a pair of entities for having no recorded data. A pair of entities with moderate similarity in all modalities will be scored higher than a pair of entities with high similarity in only one shared modality. For

Extreme Mean the only values of interest between a pair of entities are the values where they both share recorded data and have data that has been retained after thresholding. It is similar to *NEMO* and *Mean* S_i *ignoring* NaN but entities are more likely to have no shared values due to the thresholding step.

For the partial data evaluation, I select five of the modality problems in Table 3.1 for evaluation; *Easy*, *Mixed Normal*, *1Rand*, *Noisy* and *Mixed Noisy 1Rand*. I create five instances of each modality problem and then mask entities from modalities i) at random and ii) based on cluster membership. I mask a maximum of one modality per entity. I compare the AMI of labels predicted by Leiden and SBM clustering to the both the truth cluster membership y and the list of removed modalities per entity — y_{NaN}.

3.5 Results

3.5.1 Integration Networks

Clustering Performance

In Figure 3.2, the adjusted mutual information (AMI) performance of five similarity integration methods on 20 instances of 15 different modality are shown for A) Stochastic Block Model (SBM), B) Leiden and C) Spectral clustering methods. As a baseline reference, the average performance of the respective clustering algorithm on networks created from each single modality is also shown (*Avg Individual* G_i). The modality problems are ordered by the mean performance of all clustering algorithms on individual modality networks.

SBM clustering on SNF and NEMO networks consistently outperforms Mean S_i and Concatenated X_i networks (Figure 3.2A). There is very little difference in performance between SNF and Mean S_i for Leiden clustering (Figure 3.2B) on more challenging modality problems (Mixed Noisy onwards). SNF performs better on Split clusters and Mean S_i performs better when clusters are merged. There is a significant improvement in the SNF network over the Mean S_i for the Spectral algorithm (Figure 3.2C) on more challenging modality problems.

The most notable differences in clustering performance can be seen on *Split* and *Merged* modality problems. Consider Spectral clustering (Figure 3.2C) on *Split*, SNF and NEMO are nearly perfectly accurate where as Concatenated X_i and Mean S_i only match the average performance on individual modality networks. On *Merged*, the opposite can be seen where SNF and NEMO match the average G_i and Concatenated X_i and Mean S_i perform well. The performances on the other clustering algorithms reinforce this behaviour where SNF and NEMO struggle to incorporate merged clusters and Mean S_i and Concatenated S_i struggle with split clusters.

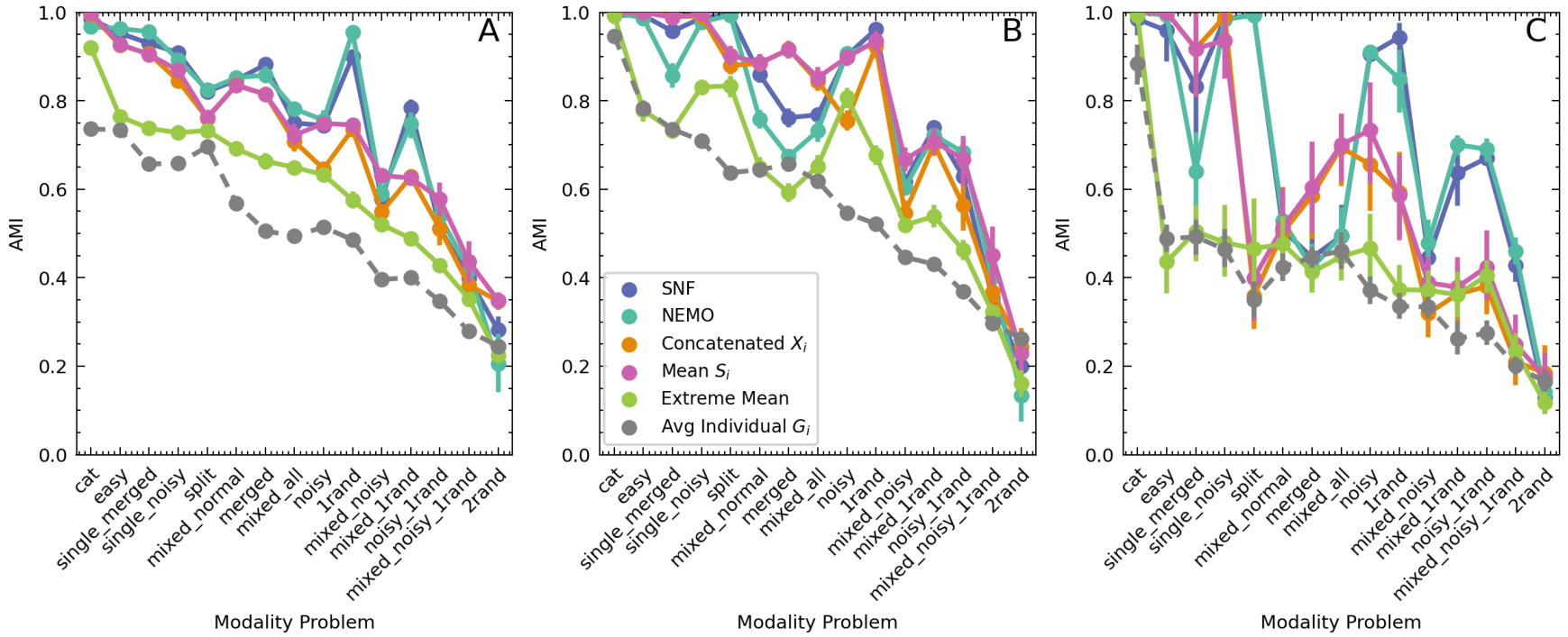

Figure 3.2: AMI Performance Comparison of Similarity Integration Methods Across Multiple Modalities. AMI performance of A) SBM B) Leiden and C) Spectral clustering algorithm on 20 instances of 15 different modality problems using Euclidean distance is presented. Five similarity integration methods are compared: SNF, NEMO, Concatenated X_i, Mean S_i and Extreme Mean. The average performance of each clustering algorithm on a KNN network G_i using each individual modality is also shown. We can see all integration methods (including simple concatenation) provide a significant improvement in performance. SNF is consistently outperformed by simpler integration methods such as Mean S_i and NEMO on Leiden clustering. Both NEMO and SNF do offer improvements in the accuracy of SBM and Spectral clustering methods. A network constructed from simple concatenation matches the performance of more complex approaches on easier modality problems. However, in higher noise settings such as *Noisy* and *Mixed Noisy* assessing each modality independently (i.e. using Mean S_i, NEMO or SNF) provides an improvement across all clustering algorithms.

Modality Problem	Easy		Single Merged		Merged		Split		1Rand		Mixed 1Rand		Mixed Noisy	
	Max	Mean	Max	Mean	Max	Mean	Max	Mean	Max	Mean	Max	Mean	Max	Mean
Graph														
SNF	**0.998**	0.968	0.997	0.906	0.941	0.696	**0.998**	**0.936**	0.987	**0.935**	**0.837**	0.720	0.674	0.546
NEMO	**0.999**	**0.981**	0.985	0.818	0.921	0.651	**0.998**	**0.937**	0.972	0.911	0.794	**0.724**	0.644	0.557
Mean S_i	**1.000**	0.976	**1.000**	**0.937**	**0.983**	**0.778**	0.982	0.688	**0.995**	0.755	0.825	0.571	**0.797**	**0.562**
Concatenated X_i	**1.000**	0.975	**1.000**	**0.938**	**0.982**	0.772	0.958	0.666	0.993	0.750	0.813	0.564	0.647	0.471
Extreme Mean	0.896	0.660	0.781	0.658	0.717	0.556	0.906	0.677	0.749	0.542	0.603	0.463	0.572	0.470

Table 3.3: Mean and Maximum AMI Performance Comparison of Similarity Integration Methods Across Multiple Modalities. Mean and maximum clustering AMI performance on the networks of the five integration methods on 20 instances of several modality problems is shown. We select seven representative modality problems to summarise performance. On problems with multiple merged modalities — *Single Merged*, *Merged*, *Mixed Noisy*, Mean S_i outperforms SNF and NEMO both in Max and Mean AMI. On *Split*, *1Rand* and *Mixed 1Rand*, SNF, Mean S_i's max performance is quite strong. It is close in performance to SNF on all 3, outperforming it on *1Rand*. Yet its mean clustering performance is significantly worse. The drop in performance is more significant than SNF's corresponding drop on merged clusters.

Extreme Mean has the worst performance on all clustering algorithms. Concatenated X_i is very similar in performance to Mean S_i. Perhaps the simplest integration method, Concatenated X_i, does not see any drop in performance over the more complex methods on less challenging clustering problems. It is notable that the Concatenated X_i network produced in datasets containing Mixed Student's-t distributed data are significantly worse for clustering. All 3 clustering algorithms show a significant drop in performance compared to Mean S_i.

While a simpler method in comparison to SNF, NEMO has very similar performance. Notably, NEMO networks also see a significant drop in performance of the Leiden and Spectral algorithms on Merged clusters. Like SNF, both SBM and Spectral clustering see significant improvement on NEMO networks over Mean S_i and Concatenated X_i. NEMO does not handle merged networks as well as SNF and the drop in performance is more significant.

In Table 3.3, we summarise the performance of the integration methods by on a subset of the modality problems. The maximum and average AMI performance of the clustering algorithms on the networks of the five similarity integration methods on 20 instances of each problem is shown. Reinforcing Figure 3.2, the maximum performance of SNF and Mean S_i is closely matched across all problems. There is far greater variation in the average performance — on *Split*, *1Rand* and *Mixed 1Rand* the average performance of Mean S_i is significantly lower. *Single Merged*, *Merged* and *Mixed Noisy* all contain multiple merged cluster modalities and we can see reduction in SNF and NEMO performance is consistent both in maximum and average performance.

Effect of SNF Affinity Kernel

In Figure 3.3, the Log difference in clustering performance between networks constructed using SNF Affinity (Eq. 3.5) and raw distance[12] is shown for Mean S_i, NEMO and SNF on 20 instances of the 15 modality problems using both correlation and euclidean distances. SNF does not gain a consistent benefit from the use of the SNF Affinity kernel. SNF Affinity provides a boost in performance on *Split*, *1Rand* and *Mixed 1Rand* problems. This is data where the clusters are split apart and fractured (each cluster is divided across uncorrelated clusters in the random modality). In contrast, the raw distance is more informative in modality problems where clusters are merged together in multiple modalities; *Single Merged*, *Merged*, *Mixed Normal* and *Mixed Noisy*. Mean S_i shows similar changes in performance but not as pronounced — networks created using SNF Affinity outperforming distance on split cluster data but worse on merged data. NEMO shows a consistent improvement in AMI when using SNF Affinity over raw distance. The benefit can be seen on all modality problems notably *Noisy 1Rand* and *Mixed Noisy 1Rand*. Unlike the other integration methods, the improvement in performance does not depend on the type of modality problem.

12. Log Diff $=$ log(Affinity AMI) $-$ log(Distance AMI)

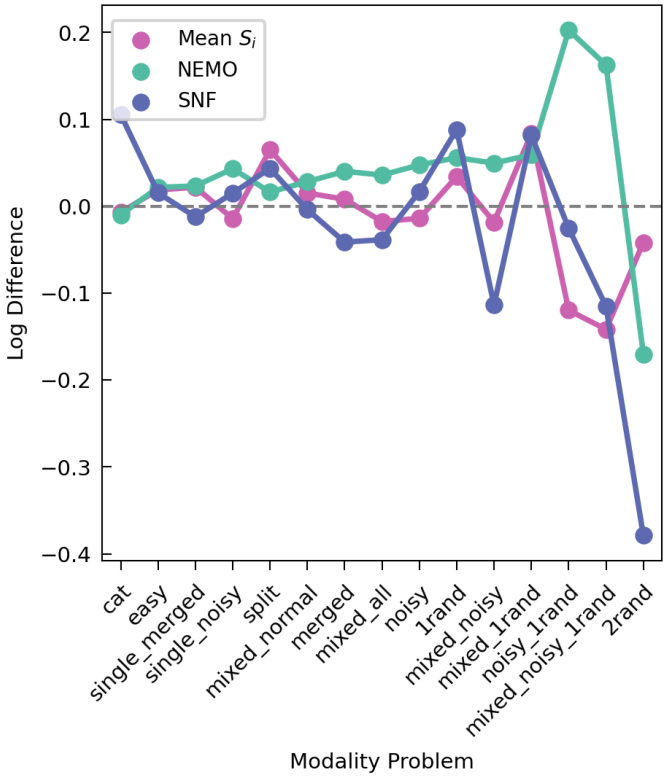

Figure 3.3: Comparison of Use of SNF Affinity vs. Raw Distance on Clustering Performance — SNF, NEMO, Mean S_i. Log difference in SBM and Leiden clustering AMI performance for networks constructed using SNF Affinity (Eq. 3.5) and raw distance for both euclidean and correlation distance metrics across 40 instances of each modality problem are shown. NEMO sees a consistent benefit in using SNF affinity over raw distance. For Mean S_i and SNF, the optimal choice changes depending on the clustering problem. We can see in high noise problems and problems involving $Merged$ clusters raw distance is significantly preferable to the SNF Affinity kernel.

Figure 3.4 again shows the Log difference in performance between SNF Affinity and raw distance but includes Concatenated X_i and Extreme Mean integration methods. The use of SNF Affinity over raw distance shows a striking drop in performance for Concatenated X_i. In contrast, Extreme Mean benefits significantly from SNF Affinity. On problems including Mixed Student's-t data the SNF Affinity provides a pronounced boost over raw distance. This is likely due to the outliers present in Mixed Student's-t data. Extreme Mean filters similarity values to include only the most similar and most dissimilar pairwise values. Outlier distances can be very large and will have a very strong effect on the final similarity between a pair of nodes. When using SNF Affinity the effect of outliers is reduced as similarity values are restricted to $[0, 1]$ and so outliers cannot have a disproportionate effect.

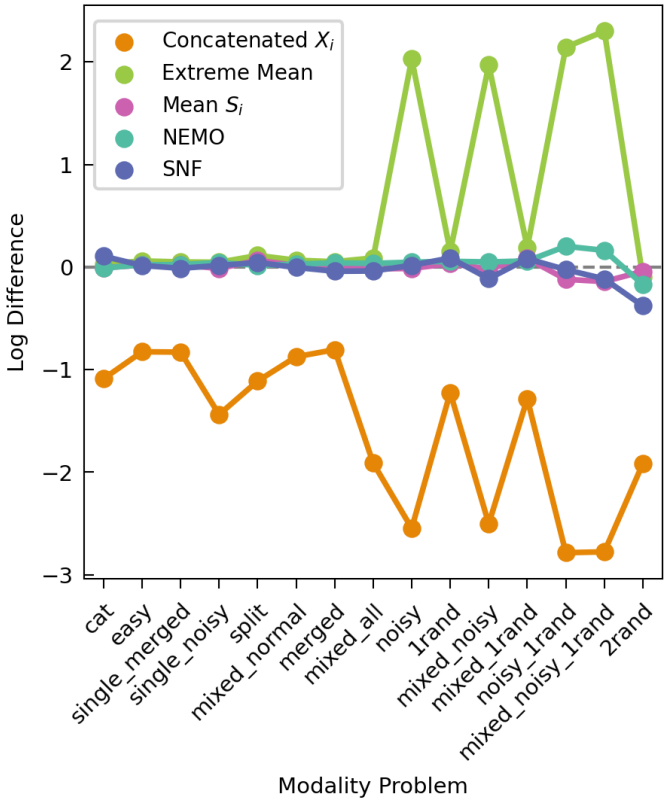

Figure 3.4: Comparison of Use of SNF Affinity vs. Raw Distance on Clustering Performance — All Methods. Log difference in SBM and Leiden clustering AMI performance for networks constructed using SNF Affinity (Eq 3.5) and raw distance for both euclidean and correlation distance metrics across 40 instances of each modality problem. Concatenated X_i performs significantly worse when a KNN network is constructed from SNF Affinity rather than raw distance across all modality problems. Extreme Mean receives a significant jump in performance when using SNF Affinity in noisy settings. This boost is a result of the SNF Affinity removing disproportionate effects of outlier distances.

Comparison to Single Modality Networks

Figure 3.5 shows the Log AMI difference in performance of SNF, Mean S_i and NEMO to the average individual network modality for A) SBM and B) Leiden clustering algorithms on 40 instances of 15 modality problems using both euclidean and correlation metrics. SNF does not show a significant improvement in performance over Mean S_i. When including networks constructed using the correlation metric, we can the improvement in the performance of SBM clustering (Figure 3.5A) on SNF and NEMO networks over Mean S_i is not pronounced with the exception of data containing random modalities. From Figure 3.5B, again we can see that SNF and NEMO do not handle problems with multiple merged modalities as well as Mean S_i but they are superior on *Split* clusters. NEMO's drop in performance on *Merged* clusters are more extreme than SNF but it matches SNF in all other scenarios. All three methods outperform the average performance of networks constructed on each modality.

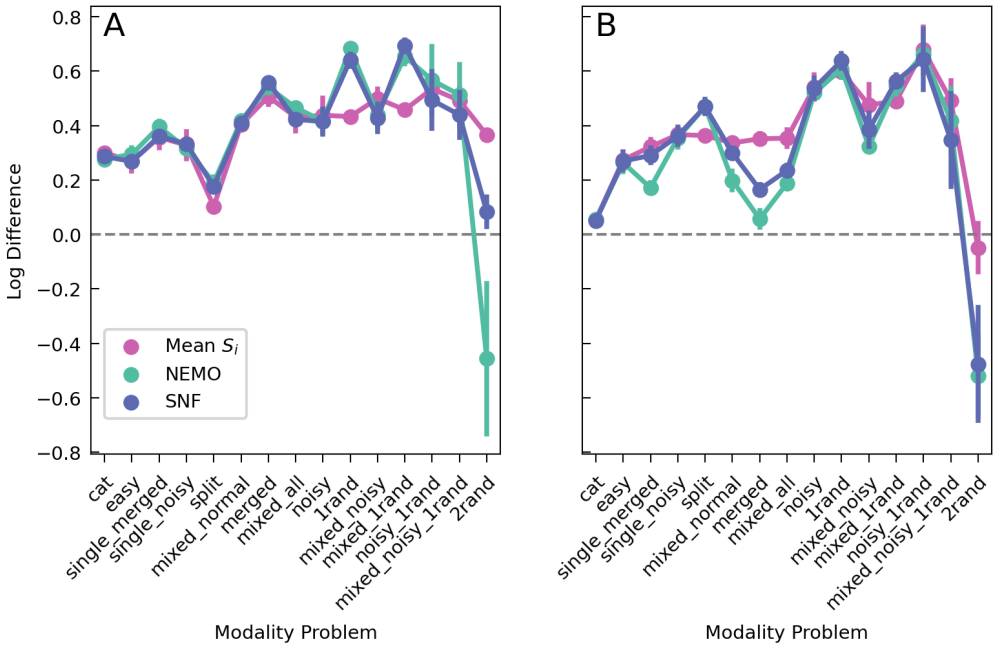

Figure 3.5: Comparison of Multi-modal Integration vs Single Modality Networks. Log AMI difference between average individual networks and SNF, Mean S_i and NEMO for A) SBM and B) Leiden clustering on 40 instances of 15 modality problems using both euclidean and correlation metrics. SNF, NEMO and Mean S_i have very similar performance across all modality problems. The lower SBM clustering performance of Mean S_i networks visible in Figure 3.2A is reduced with the inclusion of the correlation metric. NEMO and SNF struggle with multiple merged modalities and all methods outperform the average clustering performance on networks constructed on each modality.

Network Properties

In Figure 3.6, several properties of the networks produced by the similarity integration methods are shown for the 15 modality problems. The change in A) Modularity of true clusters y, B) Triad Participation Ratio (TPR) of true clusters y, C) Assortativity, D) Mean Path Length, E) Mean Degree and F) Median degree across the modality problems are shown. For all integration methods, a KNN network with $K = 25$ and 2500 nodes is constructed.

We can see Mean S_i and Concatenated X_i produces the most modular networks (3.6A). They are significantly more modular than SNF networks on problems containing multiple merged modalities; Single Merged, Merged, Mixed Normal and Mixed All. Extreme Mean shows an increase in Modularity relative to other network on problems containing Mixed Student's-t data; Single Noisy, Noisy, Mixed Noisy and Noisy 1Rand. SNF networks are as modular as the Mean S_i and Concatenated X_i networks on data without merged clusters. NEMO is significantly less modular than these networks on all modality problems.

In Figure 3.6B the TPR rate is consistently high for all methods with the exception of Extreme Mean. This indicates there is strong internal connectivity within the clusters. Nearly all nodes are triads and it is only on the more challenging noisy modality problems where the rate of triads within clusters begins to drop. As the modality problems increase in difficulty, all methods show a decrease in TPR. Mean S_i is the most resistant and is consistently high.

From Figure 3.6C, we can see SNF networks have positive degree assortativity coefficients on nearly all modality problems. The correlation is not extremely strong but on average connections between nodes of the same degree are more likely than connections between high and low degree nodes. In contrast, Mean S_i and Concatenated X_i networks have negative assortativity but the strength of the correlation is not very strong. NEMO and Extreme Mean show neutral assortativity for most modality problems and within these networks connections between all types of node degree are equally likely. One notable pattern is the drop in assortivity shown by Mean S_i and Concatenated X_i networks on modality problems containing noisy modalities. In contrast, NEMO networks show an increase in degree assortativity on these problems.

Figure 3.6D shows the Mean Path Length for all networks drops as the modality problems becomes more challenging. NEMO and Extreme are the most consistent but this are result of the high interconnectivity i.e. lower mean path length on easier problems. An decrease in mean path length corresponds to clusters becoming less distinct as more inter cluster edges are present in the network. The more connections between clusters the lower the average path length as the network becomes easier to traverse. Mean S_i and Concatenated X_i show a significant drop in mean length on data containing Mixed Student's-t distributed modalities. SNF, NEMO and Extreme Mean are more resistant to the noisy data and do not show a decrease.

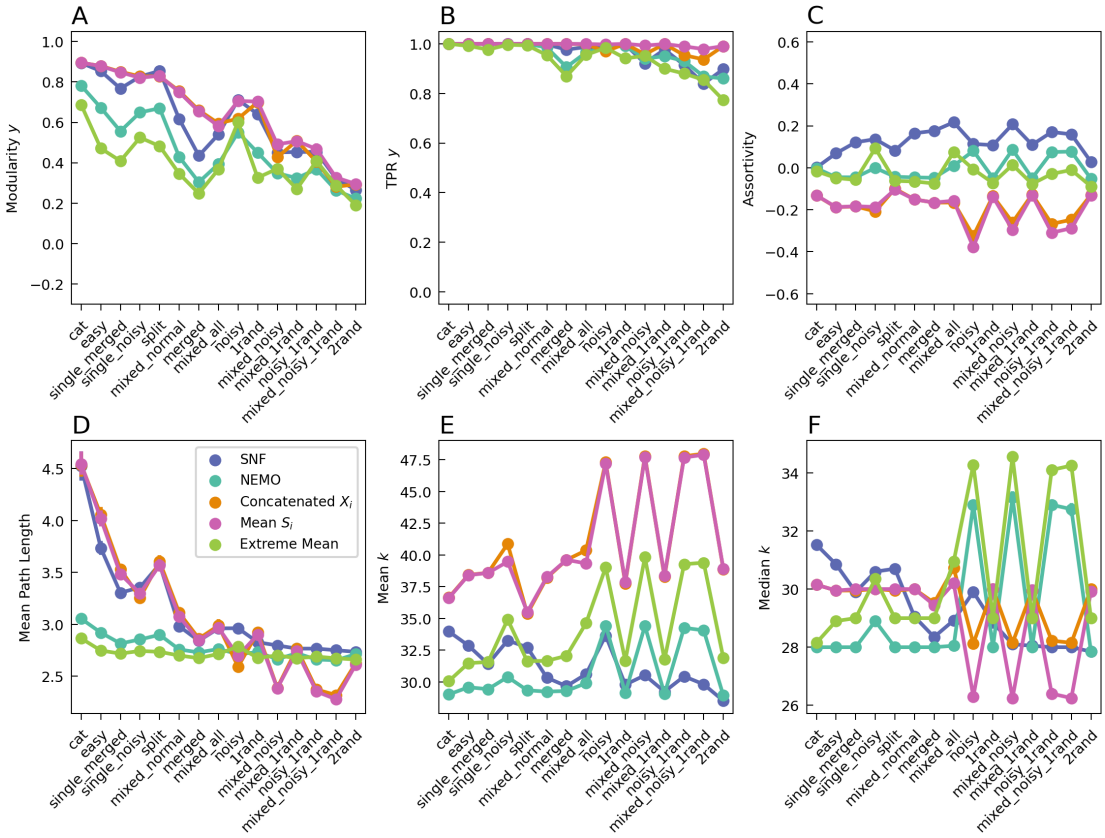

Figure 3.6: Comparison of the Network Properties of Integration Methods. The A) Modularity y, B) TPR y, C) Assortativity, D) Mean path length, E) Mean Degree and F) Median Degree are shown for 20 instances of networks on all 15 modality problems. Mean S_i and Concatenated X_i have very similar properties, with Mean S_i slightly more modular (A) and more likely to contain edges between high and low degree nodes (C). Unlike other methods, SNF structure is less affected by Mixed Student's-t distributed data (D-F). Its density does not increase and the mean path length is consistent. From C), we can see SNF has positive assortativity — connections are more likely between nodes of similar degree. NEMO networks are neutral and connections between nodes of all degrees are equally likely.

The same number of neighbours ($K = 25$) are assigned to each node in each network. As a result, any increase in mean degree i.e. an increase in network density and total number of edges in the network, implies that less nodes are mutual nearest neighbours. Mutual nearest neighbours are nodes which include each other in their set of nearest neighbours (NN). A drop in density occurs when nodes are mutual NN because only one single edge is added to the network instead of the two edges that would exist if they were not mutual NNs. As seen in Figure 3.6E, All networks except SNF show an increase in mean degree on modality problems containing Mixed Student's-t data. Mean S_i and Concatenated X_i consistently have the highest density of all networks.

Very different behaviours occur in the median degree of the distribution however. From Figure 3.6F we can see Mean S_i and Concatenated X_i display a decrease in median degree where NEMO and Extreme Mean show an increase. This can be explained by the change in assortativity on the networks. In NEMO and Extreme Mean the additional edges that result in an increase in density occur between nodes of similar degree. In Mean S_i and Concatenated X_i, these connections are between high and low degree nodes. When we consider the corresponding decrease in mean path length seen on these networks, these edges likely occur between clusters rather than within clusters.

3.5.2 Influence of Number of Modalities

Clustering Performance

In Figure 3.7, the evolution of Adjusted Mutual Information (AMI) clustering performance is depicted with an increasing number of modalities for the Stochastic Block Model (SBM) clustering algorithm across five instances, denoted as A) *Easy*, B) *Noisy*, C) *All*, D) *MergeSplit*, E) *Mixture*, and F) *Any* modality problems. Notably, both Similarity Network Fusion (SNF) and NEighborhood based Multi-Omics clustering (NEMO) demonstrate a trend of converging towards perfect detection as the number of modalities rises. This holds true even for challenging scenarios such as noisy data with a high number of outliers (Figure 3.7B) and data containing uncorrelated clusters (Figure 3.7E and 3.7F).

Surprisingly, Extreme Mean exhibits a significant improvement in performance with an increasing number of modalities, surpassing Mean S_i and Concatenated X_i. However, its convergence is slower than that of SNF and NEMO. Extreme Mean struggles when dealing with noisy data containing outliers, as is particularly evident in Figure 3.7B and Figure 3.7F. Mean S_i and Concatenated X_i perform similarly to one another but consistently underfit the data, reaching a maximum AMI of 0.9. While these methods do not decrease in performance, they tend to fall short of capturing the full complexity of the underlying cluster structure.

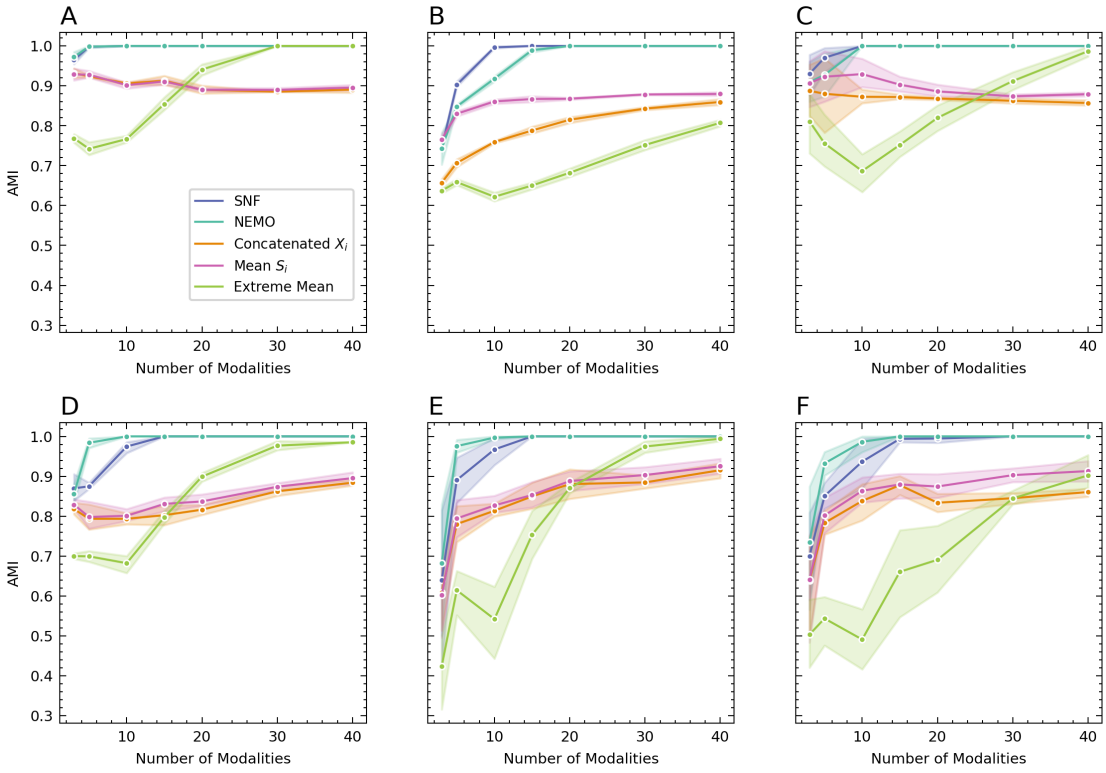

Figure 3.7: Change in AMI Performance With Increasing Number of Modalities. Change in AMI performance with increasing number of modalities for SBM clustering algorithm on 5 instances of A) *Easy*, B) *Noisy*, C) *All*, D) *MergeSplit*, E) *Mixture* and F) *Any* modality problems. SNF and NEMO converge on perfect detection as the number of modalities increase. This is true for both noisy data with a high number of outliers (B) as well as data containing uncorrelated clusters (E and F). Extreme Mean improves dramatically in performance with more modalities even outperforming Mean S_i and Concatenated X_i. Its convergence is much slower than SNF and NEMO. As seen in B) and F), Extreme Mean struggles with noisy data containing outliers. Mean S_i and Concatenated X_i perform similarly but consistently underfit the data and reach a maximum AMI of 0.9.

Ground Truth Modularity

Figure 3.8 shows the change in modularity of ground truth clusters (y) is examined with an increasing number of modalities across five instances denoted as A) *Easy*, B) *Noisy*, C) *All*, D) *MergeSplit*, E) *Mixture*, and F) *Any* modality problems. The modularity of ground truth clusters in SNF, NEMO, and Mean S_i networks plateaus at 0.9. In fact there is no network where the maximum modularity of the ground truth clusters does not exceed 0.9 despite perfect clustering performance (AMI $= 1$) of SNF.

In contrast to its lower clustering performance, Mean S_i exhibits modularity levels on par with SNF. However, it is intriguing that Extreme Mean fails to achieve high modularity in Panels C) and E), when its corresponding clustering performance (Figure 3.7C and 3.7E) being close to the maximum and higher than the more modular Mean S_i.

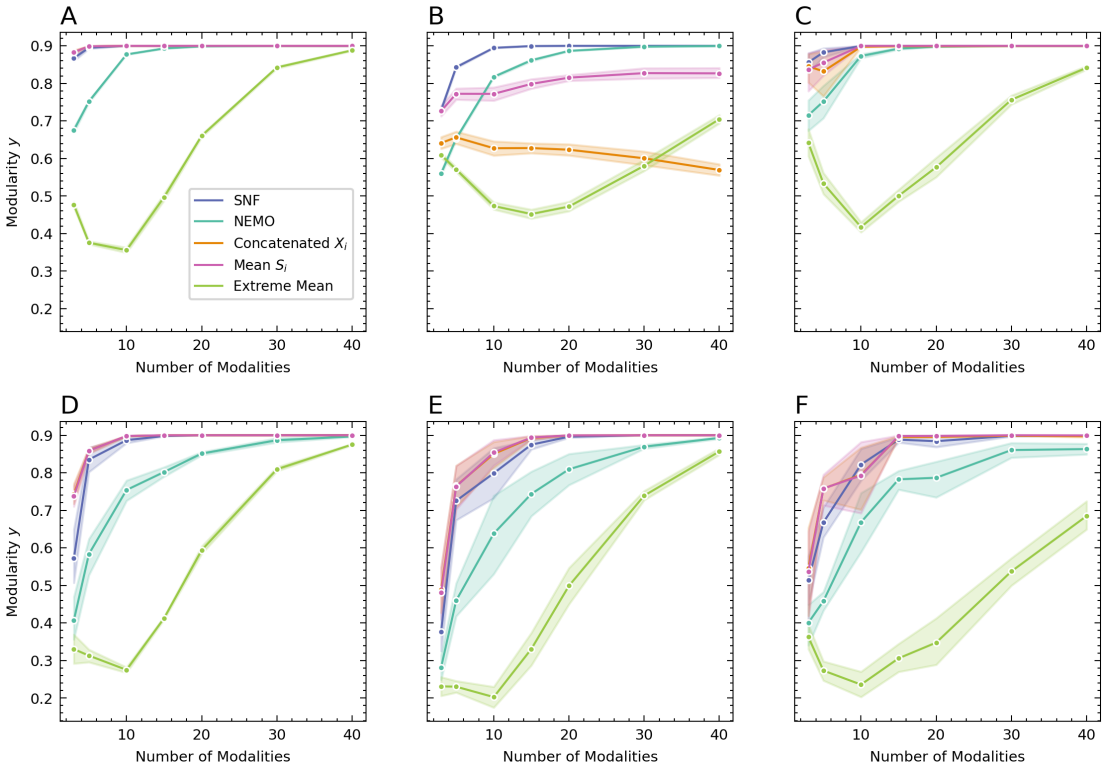

Figure 3.8: Change in Ground Truth Modularity With Increasing Number of Modalities.. Change modularity of ground truth clusters y with increasing number of modalities on 5 instances of A) *Easy*, B) *Noisy*, C) *All*, D) *MergeSplit*, E) *Mixture* and F) *Any* modality problems. The maximum modularity of the ground truth clusters does not exceed 0.9 for any network. Unlike its clustering performance, Mean S_i modularity matches SNF. Surprisingly, Extreme Mean fails to achieve high modularity in Panels C) and E) but the corresponding clustering performance (Figure 3.7C and 3.7E) is close to maximum and higher than the more modular Mean S_i.

Connected Components

Figure 3.9 shows the change in the number of components in the network is examined with an increasing number of modalities across five instances denoted as A) *Easy*, B) *Noisy*, C) *All*, D) *MergeSplit*, E) *Mixture*, and F) *Any* modality problems. As the number of modalities increases, the networks consistently split into separate components.

The SNF network consistently exhibits the phenomenon of splitting into multiple components across all modality problems. Notably, it is the only method to produce distinct components on the *Noisy* problem (B). Achieving a perfect AMI of 1.0 (Figure 3.7), the ten components produced by SNF in A-E correspond to the ground truth clusters. Interestingly, Mean S_i also generates 10 separate components but falls short of achieving an AMI of 1.0. In contrast, Extreme Mean networks do not split into separate components, while NEMO only exhibits splitting in instances A) *Easy* and C) *All* modality problems. This diverse behaviour across methods and modalities underscores the complexity of network dynamics in response to an increasing number of modalities.

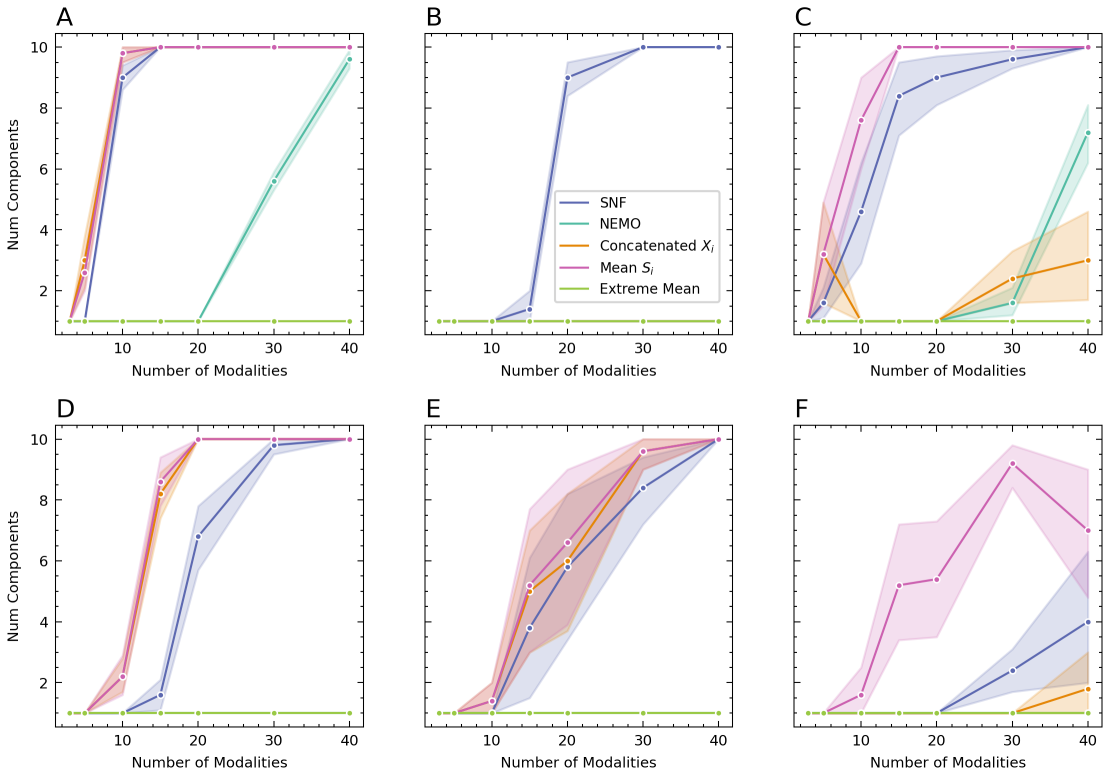

Figure 3.9: Change in Number of Network Components With Increasing Number of Modalities. Change in number of components in the network for increasing number of modalities on 5 instances of A) *Easy*, B) *Noisy*, C) *ALL*, D) *MergeSplit*, E) *Mixture* and F) *Any* modality problems. The SNF network consistently splits into multiple components across all modality problems and is the only method to produce distinct components on the *Noisy* problem (B). With a perfect AMI of 1.0 (Figure 3.7), the ten components produced by SNF in A-E correspond to the ground truth clusters. Interestingly, Mean S_i also produces 10 separate components but fails to achieve an AMI of 1.0. Extreme Mean networks do not into separate components while NEMO only splits on the A) *Easy* and C) *All* modality problems.

3.5.3 Effect of Partial Data

Clustering Performance

Figure 3.10 illustrates the variation in SBM AMI performance for increasing fraction of nodes with data partial at random across five instances of A) *Easy*, B) *Mixed Normal*, C) *1Rand*, D) *Noisy*, and E) *Mixed Noisy 1Rand* modality problems. NEMO emerges as the most resilient method to partial data. It displays the lowest reduction in performance across all five modality problems. The performance of SNF degrades significantly with any inclusion of partial data. Surprisingly, as the level of partial data increases the performance does not degrade further. Mean ignoring NaN initially is resistant to data partial at random in the Easy and Mixed Normal modality problems (Figure 3.10A & B) but once a threshold of partial data is crossed its performance drops. On the more challenging modality problem it displays a consistent reduction for all levels of partial data (Figure 3.10C-E).

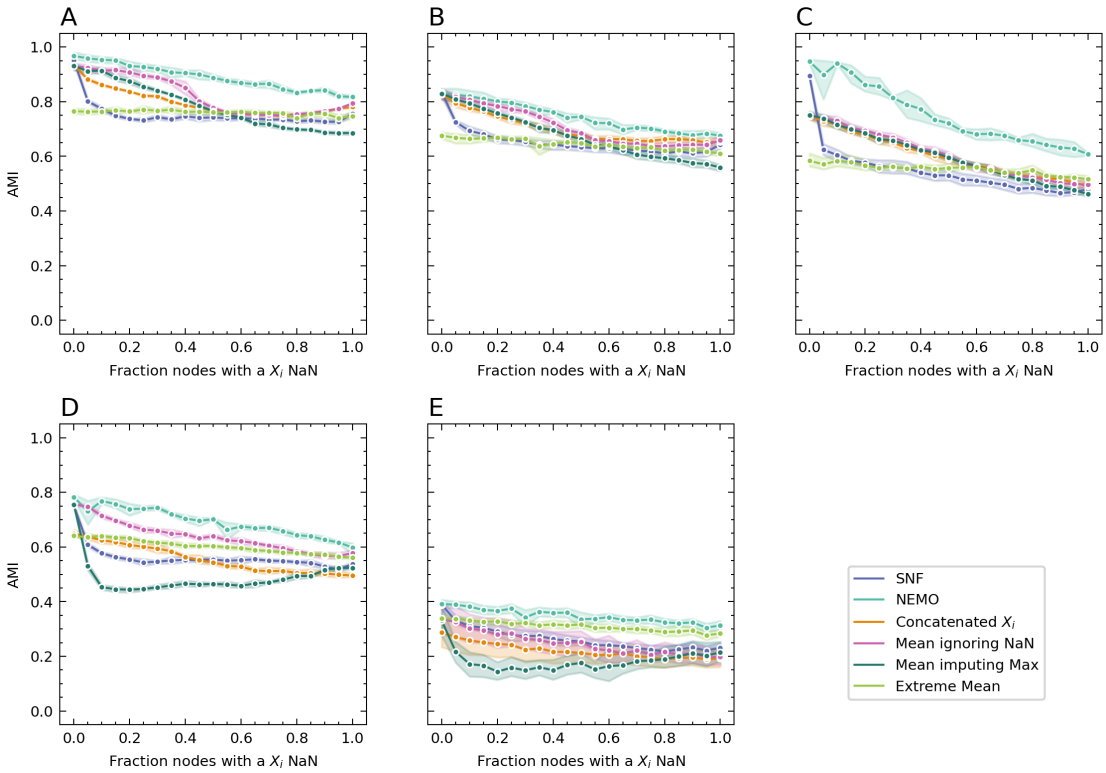

Figure 3.10: Comparison of AMI Performance of Integration Methods on Data Partial At Random. Change in SBM AMI performance for data partial at random on 5 instances of A) *Easy*, B) *Mixed Normal*, C) *1Rand*, D) *Noisy* and E) *Mixed Noisy 1Rand* modality problems. Extreme Mean is the least affected by partial data across all modality problems showing little to no change in performance. Mean ignoring NaN is more resistant to partial data than other methods up to a certain level of partial data before dropping in performance (A and B). SNF is highly sensitive to partial data and initially shows a significant drop in performance but is stable thereafter. Mean imputing Max performance degrades quickly with partial data in Noisy modality problems (D and E).

Figure 3.11 shows the change in SBM clustering AMI for cluster based partial data across five instances of modality problems: A) *Easy*, B) *Mixed Normal*, C) *1Rand*, D) *Noisy*, and E) *Mixed Noisy 1Rand*. The performance of methods demonstrates improvement on certain modalities with cluster based partial data. As the fraction of nodes with partial data increases, the consistency of clusters within each modality improves. The impact of partial data is most pronounced at 50% when there are enough members in the cluster to introduce noise to pairwise similarity within a modality, but not sufficient to form a robust cluster. When 100% of nodes have a NaN X_i, it results in only two measurements of pairwise similarity from the modalities. This explains the heightened noise observed in all methods in the *1Rand* modality (Figure 3.11C) at higher levels of partial data. For the majority of nodes, half of the similarity measurements are entirely random under these conditions.

In this setting, all methods exhibit increased resilience to partial data compared to data partial at random. The improvement of NEMO over other methods is significantly reduced. Mean ignoring NaN particularly demonstrates highly improved performance. Although there is notably more variance in the performance of methods with cluster based partial data over data partial at random, this variance increase aligns with consistently higher performance.

Mean ignoring NaN consistently shows increased performance for 100% partial data compared to no partial data. Other methods also display similar performance increases though no to the same extent. At 100% partial data, entire clusters have been removed from different X_i, resulting in each modality containing fewer clusters. In our underlying data generation procedure we strategically place clusters close enough to one another to ensure overlap. Consequently, when clusters are removed, the increased distance between clusters facilitates easier distinction. Additionally, in merged data, clusters are no longer combined together, making the remaining clusters easier to identify.

In Figure 3.12, the AMI performance of SBM and Leiden algorithms on five instances of *Easy* modality problem with data partial at random and based on cluster is displayed. We show A) SBM partial at random, B) Leiden partial at random, C) SBM partial based on cluster and D) Leiden partial based on cluster. Leiden clustering on Extreme Mean, Mean ignoring NaN, and Concatenated X_i networks appears to be relatively unaffected by cluster-based partial data (Figure 3.12D). In the case of data missing at random, Mean ignoring NaN and Concatenated Xi exhibit higher resilience compared to SBM clustering (Figure 3.12B vs 3.12A) but show a decline in performance at higher levels of missing data. On SNF networks, Leiden clustering experiences a significant drop in performance, accompanied by an increase in the variance of AMI (Figure 3.12B & D). We can see Leiden clustering is more resilient in general but shows sharp declines when the level of partial data reaches a level that corrupts the local structure (Figure 3.12B).

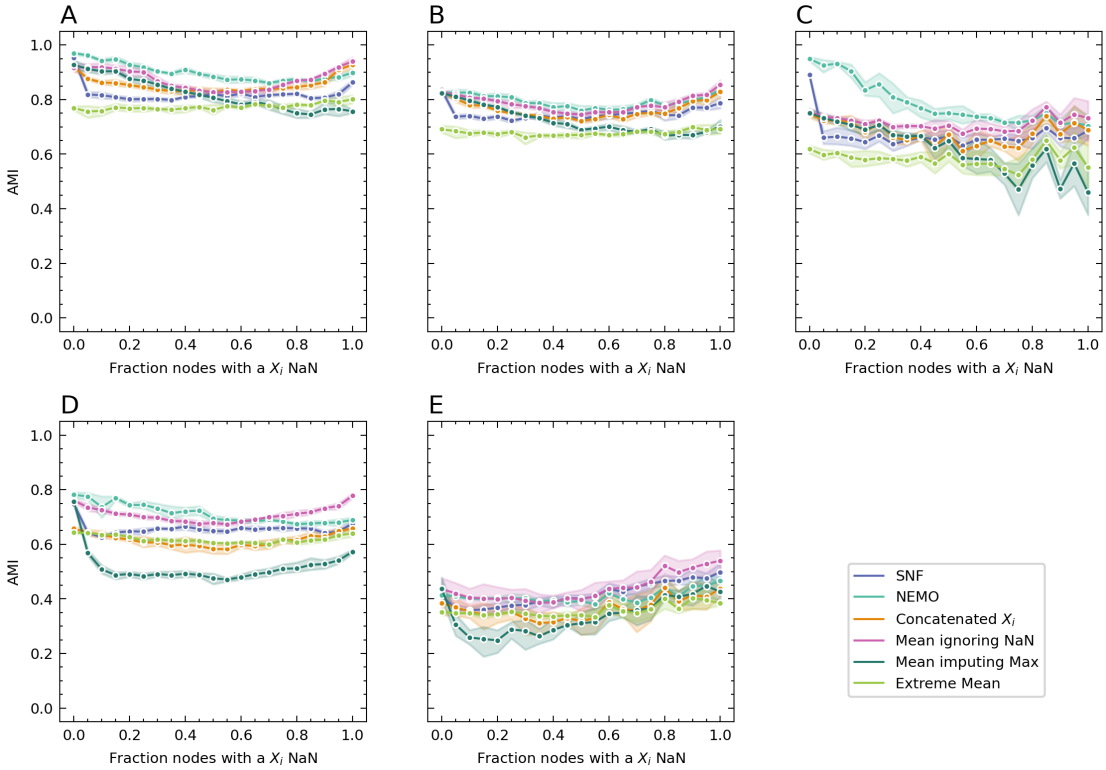

Figure 3.11: Comparison of AMI Performance of Integration Methods on Data Partial Based on Cluster. Change in SBM AMI performance for data partial based on cluster on 5 instances of A) *Easy*, B) *Mixed Normal*, C) *1Rand*, D) *Noisy* and E) *Mixed Noisy 1Rand* modality problems. As the fraction of nodes with partial data increases, the clusters in each modality become more consistent. The effect of partial data is most severe at 50% when the enough members of the cluster remain to add noise to the pairwise similarity within a modality but not enough to form a strong cluster. When 100% of nodes have a $NaN\ X_i$, we only have two measurements of pairwise similarity from the modalities. This explains the increased noise of all methods in $1Rand$ (C) at higher levels of partial data — for a majority of nodes half of the similarity measurements are completely random.

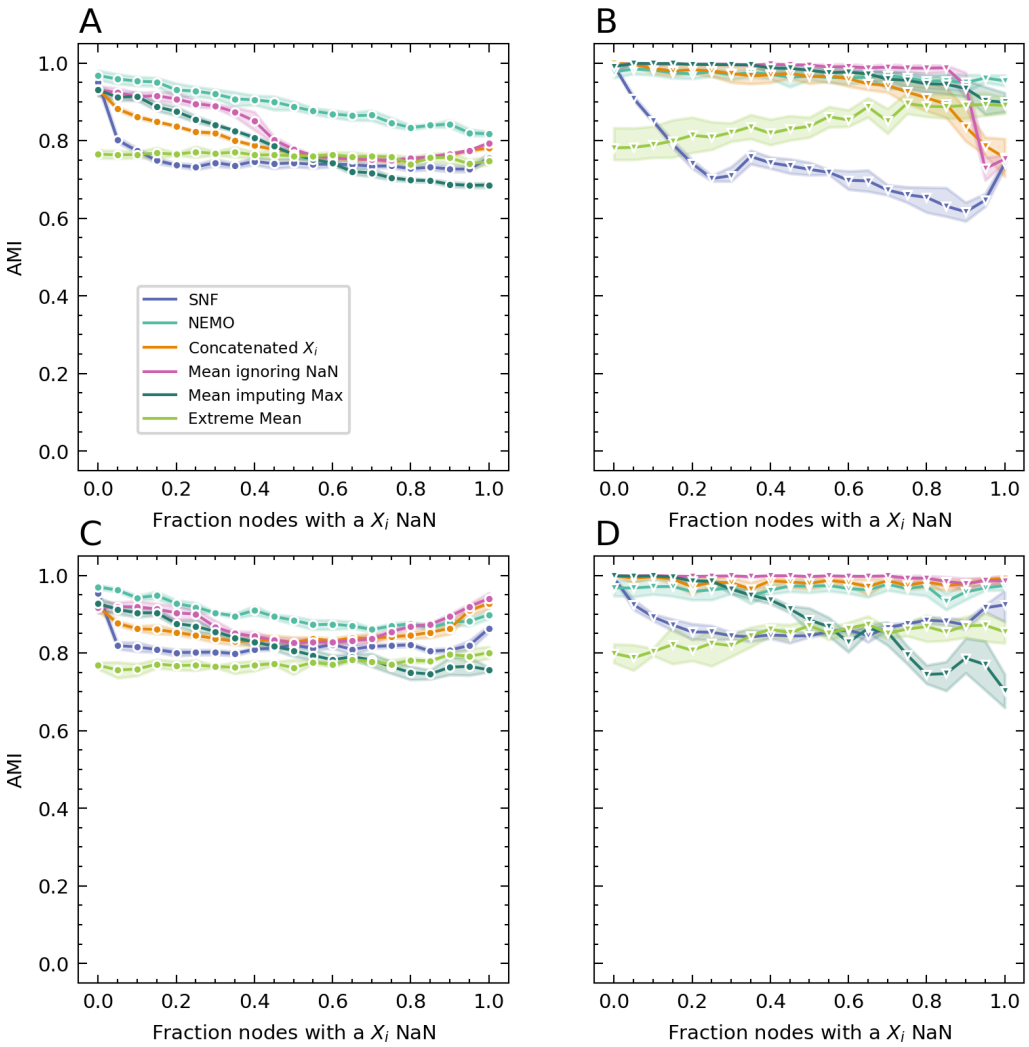

Figure 3.12: AMI Performance of Leiden and SBM Algorithms on *Easy* Modality Problem with Increasing Partial Data. AMI performance of SBM and Leiden algorithms on five instances of *Easy* modality problem with data partial at random and based on cluster. We show A) SBM partial at random, B) Leiden partial at random, C) SBM partial based on cluster and D) Leiden partial based on cluster. Leiden clustering on Extreme Mean, Mean ignoring NaN and Concatenated X_i networks is relatively unaffected by cluster-based partial data. For data partial at random, Mean ignoring NaN and Concatenated X_i are more resilient than SBM clustering but exhibit a drop in performance at higher levels. On SNF networks, Leiden clustering shows an dramatic drop in performance and an increase in the variance of AMI.

Relationship to missing labels

Figure 3.13 shows the SBM AMI between y and y_{NaN} on five instances of *Easy* modality problem with data missing at random and based on cluster. We show A) y partial at random, B) y_{NaN} partial at random, C) y partial based on cluster and D) y_{NaN} partial based on cluster. y_{NaN} AMI measures the agreement between the list of modalities each individual is absent from and the discovered clusters. The higher this AMI is the more influence partial data has on the clustering process. We can see SNF's initial drop in y AMI performance corresponds to a significant increase in y_{NaN} AMI for data both partial at random and cluster based. While the similarity between y_{NaN} and the predicted clusters increases, the y AMI remains consistent. The transition in y clustering performance of Mean ignoring NaN on data partial at random (Figure 3.13A) is amplified in y_{NaN} and the corruption of the cluster structure due to partial data is clearly visible (Figure 3.13B). Surprisingly, Mean imputing Max's y_{NaN} AMI is lower than Mean ignoring NaN at higher levels of partial data despite the worse y performance. This is true for both data partial at random and cluster based (Figure 3.13B & D).

NEMO and Extreme Mean show very interesting behaviour in their y_{NaN} AMI. Both are highly resistant to data partial at random and bear little resemblance to the list of absent modalities of each individual (Figure 3.13B). Yet for cluster based partial data (Figure 3.13D), they display a steady increase in y_{NaN} AMI. NEMO and Extreme Mean offer potential measures for detecting whether partial data is related to underlying clusters within the data — a low resemblance between clusters detected on NEMO and the labels of absent modalities could be indicative data is partial at random. Further investigation is required but the significant difference in behaviour of these methods across the types of partial data is promising.

Ground Truth Modularity

In Figure 3.14, changes in A) Modularity y, B) y SBM AMI, and C) y_{NaN} SBM AMI on five instances of *Easy* data with values partially missing at random are depicted. With 10% partial data, SNF's AMI experiences a significant drop, while its modularity is barely affected. The modularity of Extreme Mean increases, yet its cluster performance remains stable across all levels of partial data. Although NEMO exhibits slight changes in modularity, the decline in performance is much more pronounced. These disparities between AMI and modularity underscore the limitations of modularity as an alternative metric for accuracy in situations where ground truth labels are absent.

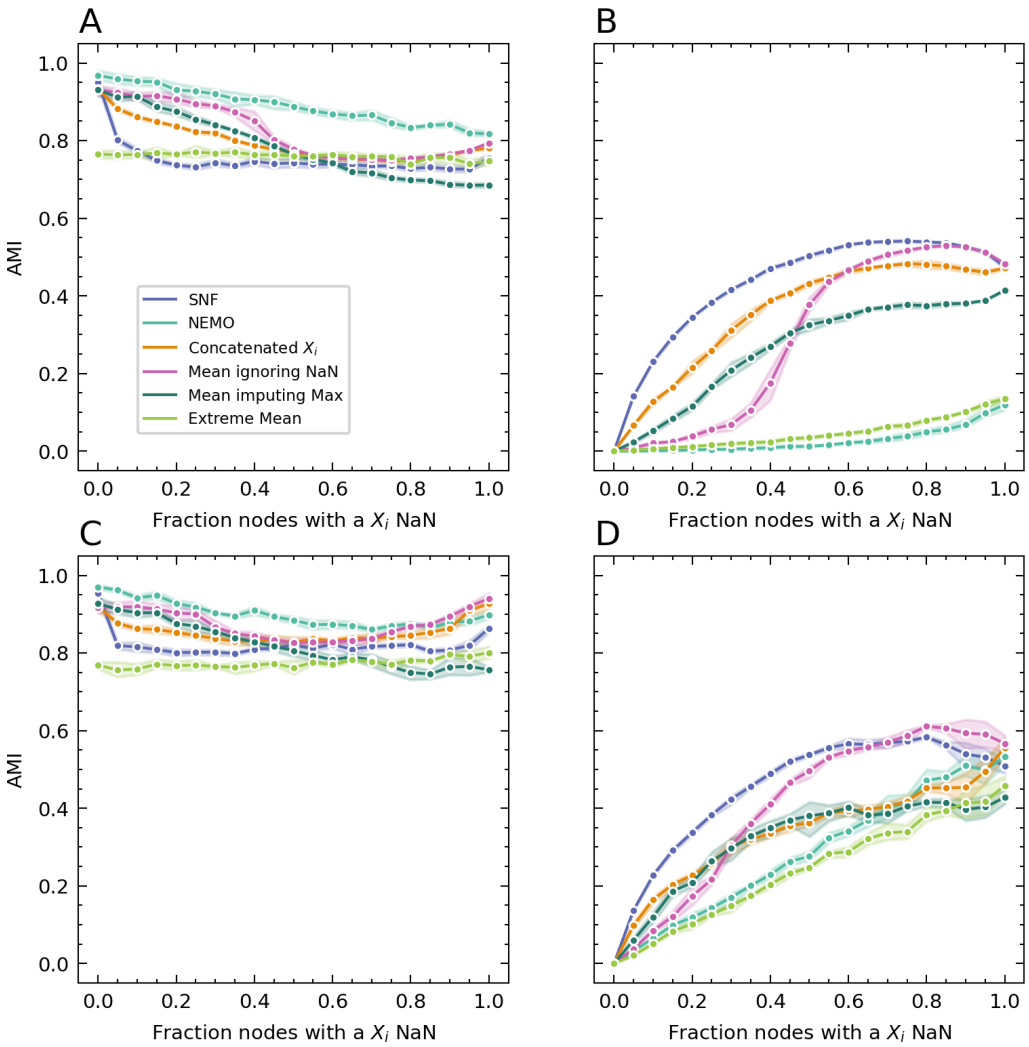

Figure 3.13: SBM AMI Between y and y_{NaN} on the *Easy* Modality Problem With Increasing Partial Data. SBM AMI between y and y_{NaN} on five instances of *Easy* modality problem with data partial at random and based on cluster. We show A) y partial at random, B) y_{NaN} partial at random, C) y partial based on cluster and D) y_{NaN} partial based on cluster. SNF is the most significantly affected by partial both at random and based on cluster. Mean ignoring NaN experiences a change in resistance when around 50% of individuals are absent at random from an X_i. It rapidly drops in performance and becomes more similar to y_{NaN}. Concatenated X_i and Mean imputing Max quickly deteriorate in performance and similar to SNF quickly align with y_{NaN}

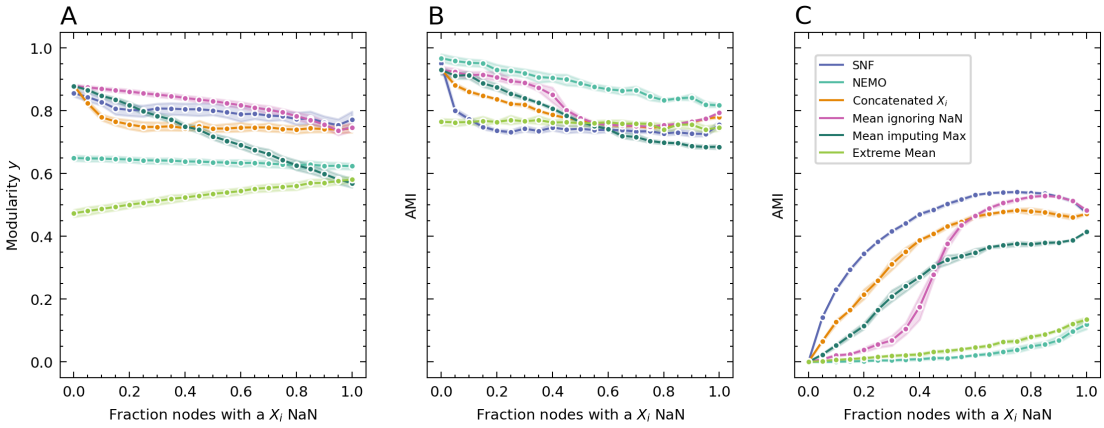

Figure 3.14: Effects of Data Partial at Random on Clustering Metrics: Modularity and SBM AMI. Changes in A) Modularity y, B) y SBM AMI and C) y_{NaN} SBM AMI on 5 instances of *Easy* data with values partial at random. At 10% partial data, SNF's AMI drops significantly yet its modularity is barely affected. Extreme Mean modularity increases with inclusion of partial data yet its cluster performance remains stable across all levels of partial data. While NEMO shows a slight change in modularity, the drop in performance is much more significant. These differences between AMI and modularity highlight the shortcomings of modularity as an alternative metric for accuracy in situations without ground truth labels.

3.6 Discussion

SNF does not emerge from this analysis as the clear choice of integration method. On key modality problems involving merged clusters, it is outperformed by simpler approaches such as Mean S_i (Figure 3.2 & Table 3.3). Merged clusters are particularly pertinent for disease subtype analysis where our expectations is that there will be a significant amount of overlap between subtypes on several modalities. For the integration of partial modalities, again a simpler approach, NEMO, is a far more optimal choice of algorithm (Figure 3.10 & 3.11). The diffusion approach of SNF does show improvements in a number of scenarios. Most notably, the incorporation of random modalities (1Rand) and Split clusters. Furthermore, its mean clustering performance is more consistent with SBM and Spectral clustering performing better across most modality problems on SNF networks than Mean S_i (Figure 3.2 & Table 3.3). That said with a different choice of metric the gap between the Mean S_i and SNF on SBM and Spectral clustering shrinks (see Figure B.1 and Figure 3.5A).

Surprisingly, the SNF affinity kernel does not provide a significant increase in the performance of SNF over raw distance (Figure 3.3). NEMO consistently benefits from the effects of the scaled exponential kernel which incorporates the distance to the nearest neighbours of both nodes. The scaled affinity takes into account a nodes local neighbourhood and normalised.

This both improves the KNN selection of NEMO and smooths out the pairwise similarity values from modality to modality improving the relative similarity calculation. While the affinity kernel is a core component of NEMO, SNF is more flexible and can accept many different similarity functions including raw distance.

Consistency of pairwise similarity scores across modalities emerges as single most important factor differentiating integration method performance. The benefit and drawbacks of KNN selection in SNF and NEMO is highlighted by differences in consistency. Mean S_i is more optimal in *Merged* clusters. In *Merged* clusters, two ground truth clusters are combined and the points of both clusters are placed at random around a single cluster centre. On average the similarity of members of a cluster is quite high. But while the average similarity is high, the K nearest neighbours of a particular node are just as likely to contain nodes from another cluster as its own. Across multiple modalities the increased similarity to nodes in the second cluster will drop while the similarity to nodes within a cluster will remain high. Mean S_i successfully identifies this. SNF and NEMO however do not calculate similarity using nodes that are not in a node's K nearest neighbours. With multiple merged clusters, the neighbours of a node selected in each modality will consistently contain nodes from other clusters increasing the difficulty of successfully identifying nodes within a cluster.

On the other hand, SNF and NEMO show strong performance on *Split* modality problems. In *Split*, clusters are separated apart. A node's K nearest neighbours will contain members of its cluster. Across different modalities, the particular neighbours might change but all will originate from the same cluster. For Mean S_i, the similarity between nodes within a sub-cluster will remain high but for the rest of the sub-cluster there will be low similarity. As we aggregate across modalities, the similarity between nodes within a cluster will oscillate between high and low reducing their overall similarity and increasing the difficulty of connecting nodes within a cluster.

The true advantages of SNF and NEMO become more apparent as the number of modalities increases, as depicted in Figure 3.7. The KNN selection process plays a crucial role in filtering out noise. Although the K-nearest neighbours (KNNs) of each node may not consistently include members of its cluster, the KNN step within each modality tends to eliminate more non-cluster members than actual cluster members. In contrast, Mean S_i faces challenges in effectively leveraging the information from additional clusters. While its performance does not decline, it struggles to capitalise on the information gained from extra clusters, resulting in a plateau in performance as the number of clusters increases.

Even more than SNF and NEMO's performances, the clustering performance of Extreme Mean underscores the advantages that similarity filtering can offer. In Chapter 2, I demonstrated that thresholding leads to sub-optimal community structure. Extreme Mean, being a variant of thresholding, exhibits subpar performance on multi-modal data, reaffirming these

observations (see Figure 3.2). Nevertheless, the notable improvement in the performance of Extreme Mean over Mean S_i as the number of modalities increases (refer to Figure 3.7) underscores the positive impact of a filtering process. Even with a less effective approach like thresholding, there is an observable enhancement in community structure.

It must be noted that the addition of corruptive modalities containing unrelated community structure (random cluster information) can have a significant impact on the ability of integration methods to detect communities robustly. If we consider the original modality problems using three modalities — *Easy*, *1Rand* and *2Rand* (Figure 3.2). There is a slight drop in performance of most methods on *1Rand* but it is not significant. The random modality is successfully incorporated. However, on *2Rand* the performance drops significantly. While the *2Rand* example is extreme, it is illustrative of the dangers of including additional modalities. Not all modalities will necessarily contain the community information we are seeking to identify. That said, the strong performance of all methods in the most challenging problems *Mixture* and *Any* (Figure 3.7) illustrates that these integration methods are quite robust.

SNF struggles significantly with partial data. It is highly sensitive to partial data and shows a significant drop in performance with its inclusion (Figures 3.10 & 3.11). In contrast, NEMO is a method developed with partial data in mind and the benefit shows. It is highly resistant to partial modalities and shows the lowest drop in performance as the rate of partial data increases. Within the methods shown here, partial data strategies that focus only on shared modalities and avoid punishing increased uncertainty show more success — Mean Ignoring NaN, NEMO and Extreme Mean. Interestingly Mean Ignoring NaN begins to struggle after a threshold of partial data is reached. The methods that filter similarity values prior to aggregating, Extreme Mean and NEMO, are more consistent across all levels of partial data.

The differences in behaviour between cluster based partial data and data partial at random is highly remarkable. Most notably certain methods improve in performance with cluster based partial data over their complete versions (Figures 3.10 & 3.11). This has significant implications. As discussed in Section 1.4, partial data is typically removed from analysis. Yet here are a number of scenarios where increased partial data is highly beneficial. I should caution here that there is likely a simplification of the clustering problem that occurs with partial that is a result of my synthetic data generation process (less clusters in a modality results in higher separation between the remaining clusters). A more in depth examination comparing the clustering with partial data to clustering on the set of data with complete measurements across modalities is needed. In any case, the reasons for partial data can be complex and, more importantly, the optimal methods for incorporating partial data can change based on the partial data process. For example, Mean S_i outperforms NEMO at high levels of cluster based partial data as seen in Figure 3.11.

3.6.1 Limitations

It must be acknowledged that my use of synthetic data introduces limitations. My synthetic data generation process is constrained by my assumptions and lacks realistic complexity. These factors may potentially compromise the generalisability of my findings. Similar to Chapter 2, the data distributions used to embed clusters are relatively simple and do not contain complex interactions between clusters. While the addition of high noise distributions like mixture of Student's-t introduces outliers and increased difficulty, there is no guarantee this is reflective of real world challenges.

There is limited variety in how clusters are embedded across modalities. The variation in cluster structures is highly simplified. While some implications can be gleaned from my merged and split clusters, these simplifications do not encompass the full spectrum of possibilities in real-world scenarios. Understanding the sensitivity of integration methods to the specific combination of modalities used is crucial for practical applicability.

Comparisons with multi-omic data sources reveal further limitations. My synthetic data maintains consistent cluster distributions with limited variations in consistency across modalities. There are no changes in the number of features, and the data remains relatively low-dimensional compared to real multi-modal data, where modalities may have tens of thousands to hundreds of thousands of features.

There are additional limitations in my partial data analysis. My choice of imputation is highly conservative, prioritising the penalisation of individuals with missing data to reflect the increased uncertainty in their features. This approach likely explains sensitivity of Similarity Network Fusion (SNF) to partial data. A less punishing imputation strategy might enhance SNF's performance. Additionally, my exploration of partial data is quite constrained, as each individual is at most missing from one modality. Real world datasets individuals are missing from several modalities. I also do not examine the effect of partial data on local structure or determine when cluster information collapses.

3.6.2 Future Work

In terms of future research, a key direction would involve expanding the modality generation framework. The consistency of a pair of nodes' pairwise similarity and similarity to their wider neighbours within a cluster are essential factors determining the addition of edges between nodes in the network. With this in mind exploring diverse configurations for cluster information across modalities is crucial. For instance, introducing targeted random noise or increasing pairwise swaps of features could be a method to test new types of consistency.

Another avenue could involve adding fully random modalities without embedded clusters, diverging from the current approach of introducing random clusters. The primary goal is to systematically delve into pairwise similarity consistency in a targeted manner, enhancing my understanding of when specific methods prove more effective.

An additional enhancement to my analysis would be a concerted effort to better mirror real-world multi-omic and multi-modal data. Introducing variations in the number of features, cluster size, and cluster distribution would increase the realism of my synthetic data, making it more representative of the intricacies found in real-world datasets. When combined with my recommendations regarding data distributions, as outlined in Chapter 2, these adjustments could significantly enhance the generalisability of my framework.

Another avenue of investigation is the assessment of partial data using less conservative imputation strategies. As outlined in Section 1.4, numerous sophisticated methods exist for imputing *item non-response*. If we consider our similarity measurements for each modality as a set of features, we can leverage more complex imputation strategies, potentially leading to improved performance. This is particularly relevant for methods like Similarity Network Fusion (SNF), which exhibited high sensitivity to partial data, making them potential beneficiaries of such strategies.

Moreover, a deeper investigation into the effects of partially complete modalities would be highly beneficial. Through the application of more complex partial modality strategies, I can better mirror the types of incompleteness observed in real-world data. My strategy for introducing partial modalities factors that relate to the underlying clusters is relatively simple — a realistic factor to likely to be more complex. Furthermore, in real world datasets, individuals may be absent from several modalities. Is there a threshold where the level of absent modality data renders an individual more of a hindrance than a benefit? To what extent do partial individuals corrupt their neighbours? Do some methods handle increased partial data more effectively than others?

Lastly, a valuable future direction involves comparing network clustering approaches to other multi-modal clustering methods, such as dimensionality reduction and matrix factorisation methods. Beyond just a comparison of clustering accuracy, exploring whether networks created from the embeddings produced by these alternative methods are more informative and reflective of community structures would provide valuable insights.

Chapter 4

Biomedical Applications

4.1 Introduction

Understanding the true capabilities of algorithms requires testing beyond the confines of synthetic data. While synthetic datasets provide a useful starting point for exploring similarity network construction in controlled environments, they often fall short of replicating the intricate complexity and diverse properties of real-world data, particularly in the biomedical field. Biomedical datasets can range from extremely high-dimensional multi-omic data, including measurements from genes, proteins, and methylation sites, to lower-dimensional medical questionnaires characterised by interlinked questions, frequent missing observations, and qualitative ordinal features Arslanturk et al. (2016). Despite my efforts to design synthetic data that approximates these characteristics, it inevitably lacks the nuanced variability and challenges that real-world biomedical data presents.

In previous chapters, I evaluated how the construction of similarity networks affects community detection in both single-modality and multi-modal settings, using synthetic data designed to mimic the properties of biomedical datasets. However, real-world biomedical data presents additional challenges, including imbalanced datasets with a low number of observations relative to the high number of features[1] Feldner-Busztin et al. (2023); S. Wang et al. (2021). This observation-feature imbalance is further complicated by issues such as partial data, which arise due to resource constraints, the diversity of measurement tools, and the rarity of conditions or willing participants Hall et al. (2019); Piantadosi (2005); Santiago-Rodriguez and Hollister (2021). These challenges starkly contrast with the extensive labelled training data available in more common machine learning settings, such as image classification He, Zhang, Ren, and Sun (2016).

In this chapter, I build upon the findings from synthetic data evaluations by testing on real-world biomedical datasets. Specifically, I verify these findings on two data sources with known ground truth that encapsulate key properties of biomedical data: high-dimensional feature sets, partial modalities, and unbalanced class memberships. The datasets include three cancer types from The Cancer Genome Atlas (TCGA) Tomczak et al. (2015) —— breast invasive

1. With the arrival of large scale resources such as the UK Biobank, this is improving.

carcinoma (BRCA), lower grade gliomas (LGG), and the pan-kidney cohort (KIPAN) —— as well as phenotypic measurements from the Simons Simplex Collection (SSC) Fischbach and Lord (2010), which includes a cohort of probands with confirmed autism spectrum disorder (ASD) diagnoses and their unaffected siblings.

Previous chapters identified several key findings: Similarity Network Fusion does not provide a significant advantage over simpler integration methods; Spectral clustering underperforms compared to the modularity maximisation of Leiden clustering and the generative approach of Stochastic Block Modelling (SBM); and NEMO demonstrates much more effective incorporation of partial data compared to other integration methods. This chapter aims to assess whether these findings hold true when applied to the more complex challenges posed by real-world datasets. The TCGA and SSC datasets introduce increased rates of partial data, significantly higher dimensionality, and a greater number of modalities, making them more challenging than the synthetic datasets previously examined.

Building on the approaches outlined in Chapter 3, I evaluate the quality of networks produced by a set of multi-modal integration techniques by measuring the clustering performance of various algorithms. These integration methods are tested on both complete and partial versions of the TCGA and SSC datasets. Additionally, to further investigate the quality of the constructed networks, I assess the predictability of the discovered clusters and illustrate the factors that influence cluster membership.

4.2 Related Work

4.2.1 Cancer Subtypes

The Cancer Genome Atlas (TCGA) Programme is a collection of 33 different types of tumour samples from over 11,000 individuals with genomic sequence, expression, methylation and copy number variation data publicly available to analyse Tomczak et al. (2015). This collection of open data has facilitated the identification of multiple cancer subtypes and improved the understanding, care and treatment of a plethora of different cancers Grossman et al. (2016); Verhaak et al. (2010). Data analysis of TCGA data has varied from supervised detection of important features Malta et al. (2018) to unsupervised clustering with a focus on identifying tumour subtypes Brannon et al. (2010).

The aim of cancer subtyping is the identification of tumours with different molecular underpinnings so as to better understand tumour biology and improve treatment strategies. Subtypes are often characterised by different rates of survival Brannon et al. (2010); Verhaak et al. (2010); B. Wang et al. (2014). By gaining an understanding of the underlying cause of a

particular cancer subtype, better treatment options or molecular targets can be identified for that particular group Olopade, Grushko, Nanda, and Huo (2008). Furthermore, by reducing the heterogeneity within each subtype, altered features which were previously unidentifiable can be uncovered Saria and Goldenberg (2015).

Multi-View Learning

While TCGA has been extensively used to develop techniques for multi-omic data analysis, it has also emerged as a common test setting in the wider field of multi-view learning Rappoport and Shamir (2018); Serra et al. (2015). Multi-view learning is focused on the integration of multi-modal data in data analysis approaches. TCGA is a large, well maintained, publicly accessible cohort. Furthermore, the challenges posed by TCGA data are reflective of the wider challenges faced by multi-modal data. It is comprised of modalities that differ significantly in size, quality, and completeness. Only a subset of samples within each TCGA dataset have a full set of measurements in each modality, leading to difficulties with data wastage or complexities in analysis.

TCGA has been notably used to assess similarity network integration methods. Two state of the art approaches to multi-modal similarity network construction; Similarity Network Fusion (SNF) B. Wang et al. (2014) and NEighborhood-based Multi-Omics clustering (NEMO) Rappoport and Shamir (2019), were both evaluated using subsets of TCGA data. However, a challenge arises when evaluating community detection and clustering methods in TCGA due to the scarcity of known subtypes. In both the SNF and NEMO papers, the lack of a comprehensive set of ground truth tumour subtypes required alternative assessment based on differences in survival rates and the number of differentially expressed clinical features found within identified clusters. These measures, survival rates and number of significant features, were also the a primary metric used in a review of multi-omic approaches Rappoport and Shamir (2018). While evaluating the effectiveness and plausibility of clusters identified by a single method is feasible through these assessments, the benefit of comparing these methods to others using the same approach is less obvious. As demonstrated in Chapter 3, NEMO and SNF perform similarly, with SNF excelling with well-structured and less noisy data, while NEMO is more suitable for partial data with unknowns (the context for which it was initially developed). However, this level of detail is cannot be uncovered within the current benchmark approach that relies on comparing group survival trends and the count of enriched clinical features.

Another alternative has emerged in the multi-view community, involving the comparison of clustering performance on pan-cancer cohorts, where various TCGA sets are combined for assessment. These methods face equivalent issues due to the unrealistic nature of the prediction tasks they pose. Tumours from different origins vary significantly in their molecular and genomic characteristics. Moreover, the presence of batch effects, which are often overlooked, makes pan-cancer clustering problems far simpler and are not truly representative of the complex multi-omic subtyping challenges that these clustering methods aim to address.

Subtypes within the TCGA

Some molecular subtypes have been identified since the inception of TCGA. The Pam50 breast cancer classification Mathews et al. (2019) is a stratification of Breast invasive carcinoma that uses the gene expression of collection of 50 genes to separate tumours into 5 distinct subclassifications; HER2, Basal-like, LumA, LumB and Normal-like. These classifications are an extension of differentiation based on overexpression of the growth factor receptor HER2. All the Pam50 subgroups have been shown to have significantly different survival profiles; HER2 and Basal-like (also known as triple-negative) classifications typically show poorer prognosis compared to the other classifications. Furthermore, treatments particular to individual Pam50 sub-classifications have been developed and are currently in use clinically, highlighting the benefit of tumour subtyping Nielsen et al. (2010).

Another TCGA dataset where molecular subtypes have been identified is within Lower Grade Gliomas (LGG) Deng et al. (2023). The three subclassifications of LGGs are characterised by mutations in the isocitrate dehydrogenase (IDH) gene and co-deletion of chromosome arms 1p/19q (1p/19q co-deletion). The IDH-wildtype is associated with poorer prognosis. The IDH Mutant tumours are split into tumours with 1p/19q co-deletion and tumours without codeletion. Both subtypes are associated with better prognosis.

Furthermore, TCGA has gathered several Renal Carcinoma cohorts, including Clear Cell Renal Cell Carcinoma (KIRC), Papillary Renal Cell Carcinoma (KIRP), and Chromophobe Renal Cell Carcinoma (KICH). These subtypes are identifiable by their unique histological appearances. Yet, beyond their visual differences, their molecular underpinning is also notably diverse. As a test set for assessing clustering methods, these renal carcinoma subtypes offer a simpler evaluation scenario compared to the complexity of the PAM50 and LGG subtypes. In contrast to the pan cancer benchmark used in multi-view learning, these tumours all originate from the same anatomical area, the Kidney.

4.2.2 Autism Spectrum Disorder

Autism Spectrum Disorder (ASD) is a neurodevelopmental disorder that is estimated to affect about 1% of people worldwide. Presentation of ASD in patients is extremely heterogeneous, with symptoms including alterations in social functions, restricted interests, abnormal repetitive behaviours, and problems with verbal communication. There is also great variation in the severity and developmental trajectory of individuals with ASD Lai, Lombardo, and Baron-Cohen (2014). As a result, determining the best strategy to support and treat patients is a complex and poorly understood process Lord, Elsabbagh, Baird, and Veenstra-Vanderweele (2018). While research has largely focused on identifying and understanding the genetic components of ASD Lord et al. (2022), significant questions remain over the existence of robust and distinct patient subtypes within ASD cohorts Agelink van Rentergem, Deserno, and Geurts (2021).

The diagnoses of ASD poses significant challenges. A clinical diagnosis typically involves extended observation by a trained clinician, interviews with the individual and their parents, and relies on attentive parents, teachers, or family physicians for referrals Falkmer, Anderson, Falkmer, and Horlin (2013); Lord et al. (2022). This process is not only time-consuming but also financially demanding. The intricate nature of ASD amplifies the challenges in providing effective treatments and support to individuals affected by it. Additionally, the heterogeneous nature of ASD makes understanding developmental trajectories difficult. For young children, predicting the manifestation of ASD and determining the most suitable treatments for a fulfilling life can be challenging.

Identifying distinct subtypes within Autism offers promising advantages for both patients and clinicians. The identification of distinct Autism subtypes has the potential to assist patients and clinicians alike. Improved homogeneity within specific subtypes could enable clinicians to provide patients with more accurate prognoses, offering patients a clearer understanding of how Autism may influence their lives. Moreover, it opens avenues for tailored support strategies for different subgroups. Subtyping holds the potential to improve our understanding of the genetic component of ASD. By minimising contradictory effects between groups and reducing variance within specific subgroups, it can enhance the statistical power of genome-wide association studies (GWAS). Presently, the vast heterogeneity within Autism presents a significant obstacle in improving prognostic accuracy and delivering tailored support.

ASD Subtypes

A number of previous work has attempted to identify subtypes of ASD however, they have been hampered by small cohort sized (c.500 individuals), or the use of only one single diagnostic survey, such as the Repetitive Behaviour Scale-Revised, as a basis for stratification Agelink van Rentergem et al. (2021). The development of larger and more comprehensive datasets such as the Simons Simplex Collection (SSC) Fischbach and Lord (2010) has enabled subtyping of larger cohorts. However, analyses of the SSC (as well as analyses of other smaller cohorts) have focused on stratification through a small subset of summary scores Matta, Zhao, Ercal, and Obafemi-Ajayi (2018). The SSC has a rich set of phenotypes available providing a very comprehensive description of each individual's particular presentation of Autism. While summary scores provide very accurate diagnostic information, their utility as a basis for stratification has not been definitive. Often stratification has been performed using Latent Class Analysis (LCA) which provides a highly interpretable model but has poor scalability and is less suited to analysis with a large number of variables Greaves-Lord et al. (2013); Wiggins et al. (2017).

The standard diagnostic questionnaires used in ASD assessments are typically condensed and standardised across the population before being employed in subtyping analyses. It's plausible that delving into lower-level phenotypic features could unravel the heterogeneity within ASD. These features are currently captured by the questionnaires but have not yet been incorporated into a comprehensive multi-modal analysis. The construction of similarity networks and subsequent clustering offer the potential to unlock the multidimensional heterogeneity observed in ASD. As discussed in Section 4.2.1, similar network approaches have proven successful in delineating tumour subtypes within multi-modal settings characterised by tens of thousands of features.

SFARI Collections

The Simons Foundation Autism Research Initiative (SFARI) has organised a number of studies with the aim collating several Autism cohorts; Simons Simplex Collection (SSC) Fischbach and Lord (2010), Autism Inpatient Collection (AIC) Siegel et al. (2015), Simons Searchlight Simons VIP Consortium (2012) and Simons Foundation Powering Autism Research for Knowledge (SPARK) Feliciano et al. (2018). Each cohort was created with different aims, each falling under the general scope of research into the genetics of Autism. The AIC is a collection of phenotypic and genetic data of c.1500 individuals that can be characterised including a large cohort of individuals with profound autism; which can be characterised as minimally verbal, display very low adaptive functioning and/or engage in challenging behaviours. The Simons Searchlight is a collection of c.1500 individuals with rare genetic disorders and Autism diagnoses. SPARK is an online collection of c300,000 individuals (c100,000 with an Autism

diagnosis) containing phenotypic information collected through online surveys and genetic information. It is the broadest and most comprehensive sample of individuals with Autism yet its' phenotypic information is far more limited than the other SFARI cohorts.

Simons Simplex Collection

The Simons Simplex Collection (SSC) is a set of data from 2868 families collected with the aim of identifying de novo mutations that contribute to the risk of ASD Fischbach and Lord (2010). As indicated in the name, each family in the study is a "simplex", the only member of family with a diagnosis is the proband. At least one unaffected sibling and unaffected parents were required for inclusion in the study. If any parents or family members were found or suspected to have an ASD diagnosis, they were excluded from the study. Other exclusion criteria included a nonverbal mental age below 18 months, medically significant perinatal incidents and not meeting the criteria for an ASD diagnosis. The families were recruited from several clinical sites in the US and the probands can be characterised as individuals with relatively "severe" Autism.

A comprehensive set of diagnostic measurements were collected assessing a probands phenotypic presentation of ASD. Examples of the types of phenotypes assessed include social communication (Social Communication Questionnaire — SCQ Eaves, Wingert, Ho, and Mickelson (2006) and Social Responsiveness Scale — SRS Constantino and Gruber (2012)), cognitive ability (Differential Ability Scales — DAS-II Elliott, Salerno, Dumont, and Willis (2007)), problem Behaviour at home and in school (Child Behaviour Checklist — CBCL Achenbach and Verhulst (2010)), developmental coordination (Developmental Coordination Questionnaire — DCDQ Wilson et al. (2009)) and adaptive behaviour (Vineland Adaptive Behaviour Scales — Vineland II Sparrow and Cicchetti (1989)). A full list of the SSC measures is provided in Appendix C.1). For each individual, a formal diagnosis was performed using Autism Diagnostic Interview, Revised (ADI-R) Rutter, Le Couteur, and Lord (2003) and the Autism Diagnostic Observation Schedule (ADOS) Gotham et al. (2007).

The SSC provides a clear example of some issues that typically arise with data collected through biomedical studies. It is a large cohort of with rich phenotypic data available. However, it has a notable issue with partial data; the measurements taken for each individual are not identical. A number of measurements require the use of different modules based on age or language ability; ADOS, DAS-II, Ravens (Raven's Progression Matrices Raven (2003)), CBCL or TRF (Teacher Report Form Achenbach and Verhulst (2010)). As a development disorder, ASD can be identified at any age although typically presents itself in preschool or early school

years. However differences in development and language ability require significantly different measures and assessment. CBCL, DAS-II and TRF all have different modules based on whether a child is of an age to attend school or not.

Identifying a shared cohort is a non-trivial task. Some of this data is central to a clinical diagnosis; ADOS is one of the key criteria for a diagnosis of Autism within this cohort. Restricting analysis to a particular ADOS module significantly restricts the extent of possible analysis of the SSC cohort to either a specific age group or specific level of language ability. While the modules collect equivalent aspects of an individual phenotypic profile, it is non trivial to combine modules. Each module is standardised and comprised of a varying number of measures. For example CBCL 6-18 has investigates a child's behaviour at school but this is not available within CBCL 2-5. Imputation does not work for these types of issues as the feature does not make sense in the context of that individual.

It is important to emphasise that there is no expectation that the split of individuals based on ages is reflective of any potential underlying subtypes. In traditional analysis, the incorporation of distinct diagnostic modules require a choice to be made between limiting analysis to individuals of set age groups (toddler/school going) and language ability (non-verbal/phrase speech/verbally fluent) or discarding established diagnostic tools from the analysis. This prevents the comparison of individuals across age groups without the removal of the diagnostic tool in question. Partial data multi-modal analysis allows us to incorporate additional modules without restricting the cohort to be included in our analysis. This approach should improve the granularity and understanding within a particular cohort by including these distinct modules. A key question to answer is whether the inclusion of this partial data corrupts the analysis of the complete cohort?

4.3 Datasets

4.3.1 The Cancer Genome Atlas

There are over 30 different cancer tumour type collections available for analysis in the Cancer Genome Atlas (TCGA). I focus on 3 subsets; Breast Cancer (BRCA), Lower Grade Glioma (LGG) and Pan Kidney Cohort (KIPAN) which contain the Clear Cell Renal Cell Carcinoma (KIRC), Papillary Renal Cell Carcinoma (KIRP), and Chromophobe Renal Cell Carcinoma (KICH) cohorts (as detailed in Section 4.2.1). For all three cohort a variety of multi-omic meas-

urements are available; DNA methylation CpG sites, Reverse-phase protein array (RPPA), mRNA gene expression, mi-RNA gene expression and copy number variants. This provides a detailed multifaceted description of the molecular underpinnings of each tumour sample.

For each dataset, we have an established ground truth subclassification. I use the PAM 50 subclassfication as subtype targets for TCGA-BRCA, the IDH/Codeletion subclassfications as targets for TCGA-LGG and the renal subtypes KIRC, KIRP and KICH within the KIPAN cohort. As detailed in Table 4.1, the cohorts vary in size with the LGG the smallest at 455, and BRCA the largest with 1083. There are also varying numbers of ground truth classes; 5, 3 and 3 respectively. The classification problems are imbalanced; in particular, the LumA and LumB classifications comprise the majority of the BRCA population and KIRC comprises the majority of the KIPAN cohort. Care is required when selecting our metrics to ensure that class imbalances are accounted for in this analysis.

To preprocess the TCGA datasets, I follow the procedures outlined in Ryan, Marioni, and Simpson (2023). I perform outlier removal, missing-data imputation and normalisation. Any features with more than 50% missing values were removed. Mean value imputation was used for remaining missing values. For normalisation, I remove the mean and scale to unit variance for each feature[2]. Unlike Ryan et al. (2023), I do not perform feature selection. They make use of LASSO regression and differential expression analysis with the ground truth subtypes as targets. My aim in this work is to evaluate unsupervised clustering and the use of ground truth labels in feature selection may artificially inflate the clustering performance.

As can be seen from Table 4.2, the modalities vary not just in terms of the information captured but in terms of dimensionality[3]. The DNA methylation contains a significant number of CpG sites leading to very high dimensionality 300,000 yet has only 780 observations in the largest dataset (BRCA). In contrast the RPPA, while still of high dimension, is orders of magnitude smaller with only 460 features.

There is a significant data completeness issue present in these three TCGA cohorts. From Table 4.2, we can see that the number of observations in each modality varies. LGG is the most consistent, 325 out of 455 (70%) of individuals have a full set of measurements. By contrast, the number of complete measurements is much lower in the larger cohorts; 49% and 55% respectively. If we count the number of times an individual is absent from a modality then only 6.30% of possible measurements are missing within LGG. BRCA and KIPAN have

2. The empirical mean and variance are calculated using the values observed in each feature. This will differ for the partial and complete data.
3. These are the number of features after preprocessing.

much higher rates with 14.7% and 12% of possible measurements absent in each respective cohort. Figure 4.1 highlights that the rate of missing data is not consistent across ground truth subtypes. The least frequent subtypes within each dataset, Normal-like, IDH WT and KICH, all have significantly higher rates of partial modalities than the larger subgroups.

(a) BRCA

Sub-Group	P	C
LumA	562	301
LumB	209	107
Basal	190	86
Her2	82	37
Normal	40	—
Total	1083	531

(b) LGG

Sub-Group	P	C
IDH Mut Non-Codel	218	152
IDH Mut Codel	151	121
IDH WT	86	52
Total	455	325

(c) KIPAN

Sub-Group	P	C
KIRC	533	261
KIRP	290	202
KICH	66	26
Total	889	489

Table 4.1: Subtype Distribution in TCGA Cohorts: BRCA, LGG, and KIPAN. Breakdown of the subtypes (ground truth cluster labels) within the (a) BRCA, (b) LGG and (c) KIPAN TCGA cohorts. I further divide the cohorts into two sets of data: i) Complete (C) — entities with complete observations across all modalities and ii) Partial (P) — all available entities including those with missing measurements in one or more modalities.

Dataset	DNAm		mRNA		miRNA		CNV		RPPA	
	D	N_i	D	N_i	D	N_i	D	N_i	D	N_i
BRCA	293,649	780	27,605	1,007	1,598	978	60,265	1,014	464	840
LGG	321,999	457	22,185	437	1,515	408	60,274	450	457	389
KIPAN	310,045	658	28,212	846	1,552	793	60,274	864	469	752

Table 4.2: Data Characteristics of TCGA Modalities: Feature Count and Observations. The number of features D and observations N_i for each dataset across the five data modalities within TCGA. Modalities vary significantly both in dimensionality and completeness.

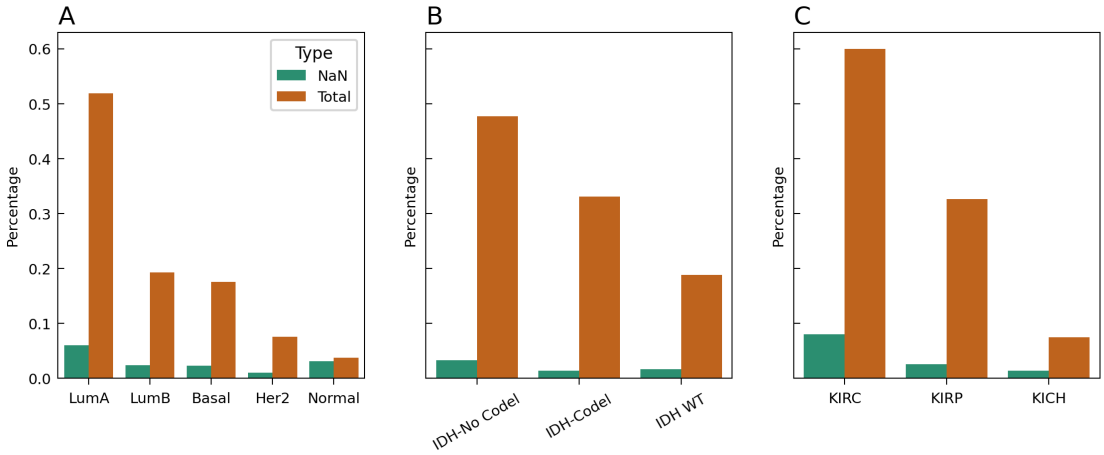

Figure 4.1: Partial Measurement Rates per Subtype in TCGA Datasets. Frequency of partial measurements per subtype within the TCGA datasets. We can see that within BRCA the Normal subtype has a significantly higher rate of individuals with incomplete modality measurements.

4.3.2 Simons Simplex Collection

The Simons Simplex Collection (SSC) is a set of data from 2868 families each containing an individual formally diagnosed with Autism between 4-18 years old. The SSC cohort is comprised of 2869 probands. After the loss of individuals due to data privacy and censoring, 2365 of the 2868 families (82%) have an unaffected sibling. Fortunately due to the longitudinal nature of the study, some of the measured Simplex's contain more than one sibling creating a cohort of 381 additional unaffected siblings. As shown in Table 4.3, this leaves a total population of 5615 individuals with 2869 probands with ASD and 2926 unaffected siblings.

While there is not a set of ground truth subtypes within the SSC, the cohort of unaffected siblings provides a control group to use as a ground truth target. We can be highly confident in the accuracy of the control group. A formal diagnosis for the proband was required for inclusion within the SSC and the group of siblings were required to have been classified as unaffected. In a network analysis, we expect this cohort to cluster separately to the ASD cohort.

It must be noted however, that not all diagnostic questionnaires were completed by the unaffected siblings limiting the phenotypic information available for analysis. After limiting our analysis to measures shared between the probands and siblings, for each individual, we have measurements of adaptive behaviour (Vineland II), autistic traits (SRS Parent and SRS Teacher), social communication (SCQ Parent and Teacher), problem behaviour at home (ABCL 18-59, CBCL 2-5 and CBCL 6-18) and problem behaviour at school (TRF 6-18 and CTRF 2-5). This create a set of 11 distinct modalities. Similarly to the TCGA data, I perform outlier

Sub-Group	Partial	Complete
Proband	2869	788
Unaffected Sibling	2365	574
Other Sibling	381	73
Total	5615	1435

Table 4.3: Subtype Distribution in SSC. Breakdown by sub-group of the partial and complete data splits within the SSC. The complete set of data is formed by limiting the cohort to children aged 6-18 that are present within the CBCL 6-18, TRF 6-18, SRS Parent & SRS Teacher, SCQ Parent & SCQ Teacher and Vineland-II modalities.

removal, missing-data imputation and normalisation. Any features with more than 80% missing values were removed. Mean value imputation was used for remaining missing values. For normalisation, I remove the mean and scale to unit variance for each feature. A breakdown of the modalities by number of observations and number of features[4] is shown in Table 4.4.

There is a far higher data completeness issue within the SSC compared to TCGA. As discussed in Section 4.2.2, certain diagnostic questionnaires are split into modules based on age or language ability. As a result individuals within CBCL 2-5 will not have measurements within CBCL 6-18. To create a complete data cohort, I limit my analysis to individuals with complete measurements within the CBCL 6-18, TRF 6-18, SRS Parent & SRS Teacher, SCQ Parent & SCQ Teacher and Vineland-II modalities. This leaves us with 1435 individuals, 25.5% of the 5615 individuals in the partial data cohort. A breakdown by individual subgroup is shown in Table 4.3. The percentage of the cohort that is complete is far lower than within the TCGA datasets. Moreover, within the partial data cohort, 53.96% of possible measures are absent (supposing that CBCL 6-18 and CBCL 2-5 could both be completed by an individual). This is twice the rate of partial data found within BRCA (14.7%) and far more challenging to incorporate.

Another difference between the TCGA and SSC data is the presence of ordinal variables with a limited set of possible values. The instruments used in the SSC data are diagnostic questionnaire comprised of ordinal features that might assess the characteristics of an individual with questions such as level of speech where the possible answers are very weak, weak, moderate, strong, very strong. Unlike numeric features such as RNA Gene expression data, the feature space is restricted and may require very different approaches to the calculation of

4. These are the number of features after preprocessing.

Modality	N_i	D
CBCL 2-5	1177	30
CBCL 6-18	4259	42
ABCL 18-59	119	50
SCQ Parent	3906	41
SCQ Teacher	2351	42
SRS Parent	5456	72
SRS Teacher	2875	72
SRS Adult	105	71
TRF 2-5	590	42
CTRF 6-18	2080	50
Vineland II	5521	47

Table 4.4: Data Characteristics of SSC Modalities: Feature Count and Observations Number of individuals N_i and number of features D within each modality of the SSC Proband and Sibling cohort. The number of observations within a particular modality varies from 5521 (Vineland II) to SRS Adult (105). Partial modalities are a significant challenge within SSC.

similarity. The dimensionality is much lower with 15-50 features typically in each measurement tool but there are far more modalities that have to be integrated together. Additionally there are far more observations with 5615 individuals in the SSC compared to 1083 in TGCA BRCA.

4.4 Experiment Setup

I want to evaluate the performance of community detection on similarity networks created using multi-modal integration methods on biomedical data. To evaluate the community detection performance, I need ground truth controls. In the TCGA, I have known subtypes for each dataset; PAM50 subclassification, IDH-Codel subtypes and Clear Cell, Papillary and Chromophobe subtypes. In the SSC, I use two subtypes as ground truth targets; Probands with a confirmed ASD diagnosis and unaffected siblings. The datasets are processed using the preprocessing steps outlined in Section 4.3.

The analysis pipeline for each of the multi-modal integration methods is as follows; i) calculate pairwise similarity on each modality M, ii) combine the pairwise similarity matrices using the multi-modal integration methods, iii) construct a K-Nearest Neighbour network from the integrated similarity score, and iv) perform community detection. I use the Pearson correlation metric to calculate pairwise similarity on each modality. The Pearson correlation metric between two individuals U and V is given by

$$d_{\text{corr}}(U,V) = 1 - \frac{(U-\bar{U})\cdot(V-\bar{V})}{||U-\bar{U}||\,||V-\bar{V}||}.$$

I normalise the pairwise similarity distributions on each modality to have zero mean and unit variance when calculating Mean S_i.

To assess the quality of the integrated networks, I employ two main approaches. Firstly, I evaluate the quality of labellings produced by several cluster algorithms using various metrics for clustering performance. Secondly, I gauge the 'predictability' of the clusters through a supervised prediction model. This model is built using a combined set of features derived from the separate modalities.

Within the supervised model, I address two aspects of the network. First, I assess the quality of the cluster labellings produced on the network by training a model to predict these labels. Second, I evaluate the network structure's quality by training a model to predict the ground truth labels using partial modalities and leveraging the network structure to impute missing values within each modality. For nodes absent from a modality, I impute the mean values of its nearest neighbours in the network.

Networks

For each integration method, I construct a KNN network from their integrated pairwise similarity. I use the same K for all integration methods. I opt for a heuristic approach to select the value of K. I select K for each dataset, both partial and complete, by taking $K = 80\% \cdot \sqrt{N}$ where N is the number of individuals in the dataset. This heuristic was chosen as it generally led to a density that corresponded to the most consistent performing networks across algorithms as shown in Chapter 2.

I evaluate the following multi-modal integration methods

- **Similarity Network Fusion (SNF)** — *de facto* standard approach for multi-omic integration and unsupervised clustering analysis. Similarity calculated through diffusion across KNN graphs. I consider two approaches to imputation in partial modalities. For each pairwise modality distance $S^{(K)}$, the pairwise value between a node i with NaN in X_k and any other node j is set to
  - **SNF Mean Mod** — Mean similarity of all complete nodes in modality K
  - **SNF Mean Pair** — Mean pairwise similarity between node i and node j in other modalities i.e. $\sum_{m \neq K} S_{ij}^{(m)}$. If nodes i and j are never present in the same modality i.e. pairwise similarity in all other modalities are NaN, then set their pairwise similarity to max distance/dissimilarity for that modality.

  SNF is then computed as normal with imputed values included.
- **NEighborhood Based Multi-Omic Clustering (NEMO)** — Mean relative similarity between nodes i and j based on a K-nearest Neighbourhood in each modality. NEMO was developed to analyse partial data. The mean relative similarity for any pair of nodes i and j is computed over the modalities where both nodes have recorded data.
- **Mean** S_i **imputing *max*** — For each pairwise modality distance $S^{(K)}$, the pairwise value between a node i with NaN in X_k and any node j is set to max distance/dissimilarity for that modality. Mean similarity is then computed between by averaging the pairwise similarity of nodes i and j across all modalities.
- **Mean** S_i **ignoring** NaN — The mean similarity for any pair of nodes i and j is computed over the modalities where both nodes have recorded data. If nodes i and j are never present in the same modality i.e. pairwise similarity in all other modalities are NaN, then set their pairwise similarity to max distance/dissimilarity.

- **Extreme Mean** — for each modality, pairwise similarity between nodes with recorded values in the modality is thresholded to only include very similar and very dissimilar connections. $S_{ij}^{(K)} = 0 \quad \text{if} \quad |S_{ij}^{(K)}| < \theta$ where θ is set to one standard deviation of the normalised pairwise similarity distribution in a modality. The mean similarity for any pair of nodes i and j is computed over the modalities where both nodes have recorded data. If all values between i and j are NaN after thresholding (including NaN when i has no recorded data in a modality) then the dissimilarity is set to max.

I also evaluate the performance of clustering on single modality networks. I calculate pairwise similarity on each modality and construct a KNN network from its individual pairwise similarity matrix. To allow fair comparison between the multi and single modality approaches, I compare their performance on the complete datasets.

4.4.1 Clustering Algorithms and Metrics

As in Chapter 3, I perform community detection on the multi-modal graph networks using three distinct network clustering algorithms

- **SBM** — Python `graph-tool`[5] Peixoto (2014) implementation of the Micro-canonical Stochastic Block Model Peixoto (2018). The number of clusters K is selected by minimising the description length.
- **Leiden** — Modularity maximisation using Leiden algorithm Traag et al. (2019). I use the Python `igraph`[6] implementationCsardi and Nepusz (2006). The number of clusters K is selected through the resolution parameter. A set of twenty potential resolution hyperparameters are generated using event sampling Jeub et al. (2018). The parameter with maximum modularity is selected.
- **Spectral** — Spectral decomposition and K-means clustering of "Random Walk" normalised Laplacian $L_{rw} = I - D^{-1}A$. The number of clusters K is chosen using the eigengap heuristic. I make use of the Python `spectralclusterer`[7] implementation Q. Wang et al. (2018).

These algorithms detect the number of clusters automatically and take distinct approaches to network community detection.

To evaluate the performance of clusters produced by the clustering algorithms, I employ three metrics:

5. v2.45
6. v0.10.3
7. v0.2.16

- **Adjusted Mutual Information (AMI)** — an information theory measure derived from joint and individual entropies of cluster labellings, given by the formula $AMI = \frac{MI - E(MI)}{f_{\text{mean}}(H(y), H(\hat{y})) - E(MI)}$. The mutual information (MI) measures the agreement between two labellings y and $\hat{y}$. The AMI contains a correction for chance Vinh et al. (2009) and, while similar to the adjusted rand index (ARI), penalises differences in the number of clusters less than ARI. (For an in-depth discussion, refer to Section 1.5.2)
- **Homogeneity (H)** — $h(y, \hat{y}) = 1 - \frac{H(y|\hat{y})}{H(y)}$ is 1 if all $\hat{y}$ clusters contain only data points which are members of a single y class. If two cluster labellings predict a higher number of classes than the true labelling, the cluster labelling that splits true clusters into subclusters has high homogeneity.
- **Number of predicted clusters** — the number of clusters proposed within a cluster labelling. All methods considered here select the number of clusters automatically.

Unlike the datasets discussed in Chapters 2 & 3 where known ground truth clusters have been embedded into the data, the datasets here are not synthetic. The presence of ground truth clusters within each modality are not guaranteed. Furthermore, additional subtypes may be present within the data that have not yet been identified. In this context, homogeneity can serve as an indicator that an algorithm detecting more clusters than expected might have identified previously unknown subtypes, rather than merely failing to accurately detect the established ground truth clusters. AMI will be our primary measure of performance but Homogeneity and Number of predicted clusters allow us a more nuanced understanding of cases where algorithms perform poorly.

4.4.2 Prediction

As choice of predictive model, I make use of a Random Forest model. This model strikes a balance between predictive power and interpretability. I employed the `scikit-learn`[8] Pedregosa et al. (2011) Python library to train random forest models. Given the high dimensionality of a number of modalities within TCGA (DNAm modalities have >290,000 features), I use Principal Component Analysis (PCA) to merge separate modalities into a unified set of features, reducing each modality to 64 dimensions. Equal dimensions were selected across modalities to facilitate the analysis of feature importance. By maintaining equal dimensionality, we can better assess the origin of the most important features, providing insight into each modality's contribution to cluster label prediction. Each modality is transformed into 64 dimensions. This transformation was applied to both complete and partial datasets within each TCGA dataset. For the partial data, the PCA transformation was fitted on the set of individuals with observations. Imputation of missing values was performed on the original features, followed by transformation into the PCA space.

8. v1.3.2

To evaluate predictability, I conducted 5-fold cross-validation splits for each dataset. Each random forest model comprised 500 estimators with a maximum depth of 4. Gini impurity importance enables us to evaluate the contribution of individual variables to model predictions, offering insight into which features play the most significant role in determining cluster labels. Unfortunately, due to the PCA transformation, the training features are linear combinations of the original features, limiting their biological interpretability. However, each modality contributes an equal number of features within the training data. We can gauge the impact of each modality on the predictability of a cluster by assessing the rate of a modality's features appearing among the top 10% most informative features.

The tasks of predicting subtypes and cluster labels are multi-classification problems. As noted previously, class imbalances exist within the TCGA data. To address these imbalances and ensure an accurate overall assessment, I employed the weighted F_1-score. This score is calculated by computing the F_1-score for each class and weighting it by its frequency

$$\text{Weighted } F_1\text{-score} = \frac{\sum_{i=1}^{C} F_1\text{-score}_i \times N_i}{N} \tag{4.1}$$

where C is then number of classes, F_1-score$_i$ is F_1-score for each class i, N_i is the number of instances of class i and N is the total number of instances in the dataset. This metric is a trade off that penalises poor performance in less frequent class while also accounting for the overall performance of the model by increasing the weight of more frequent classes. The F_1-score for each class is calculated using the formula:

$$F_1\text{-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.2}$$

where Precision = $\frac{\text{True Positives}}{\text{True Positives + False Positives}}$ and Recall = $\frac{\text{True Positives}}{\text{True Positives + False Negatives}}$.

4.5 Results

4.5.1 TCGA

Clustering Performance

Table 4.5 and Figure 4.2 illustrate the average and maximum AMI clustering performance of SBM, Leiden, and Spectral clustering within multi-modal integration networks across complete and partial datasets in the BRCA, LGG, and KIPAN datasets of TCGA. *NEMO* consistently emerges as the top-performing integration method across both complete and partial datasets

for BRCA and LGG. It notably outperforms all other methods in the complete LGG dataset; however, there is a marked decline in *NEMO*'s performance when applied to the KIPAN dataset.

Conversely, *SNF* never emerges as the optimal method. The mean clustering performance for both *Mean Modality* and *Mean Pairwise* imputation either matches or falls short in comparison to *Mean Max* or *NEMO*. Specifically, *SNF* encounters challenges with consistency when handling partial data. The BRCA partial performance demonstrates that the choice of imputation strategy can yield either strong clustering (*Mean Modality*) or poor clustering (*Mean Pairwise*). Moreover, *Mean Max* performs exceptionally well in the KIPAN dataset, while, the Extreme Mean method consistently displays the poorest performance.

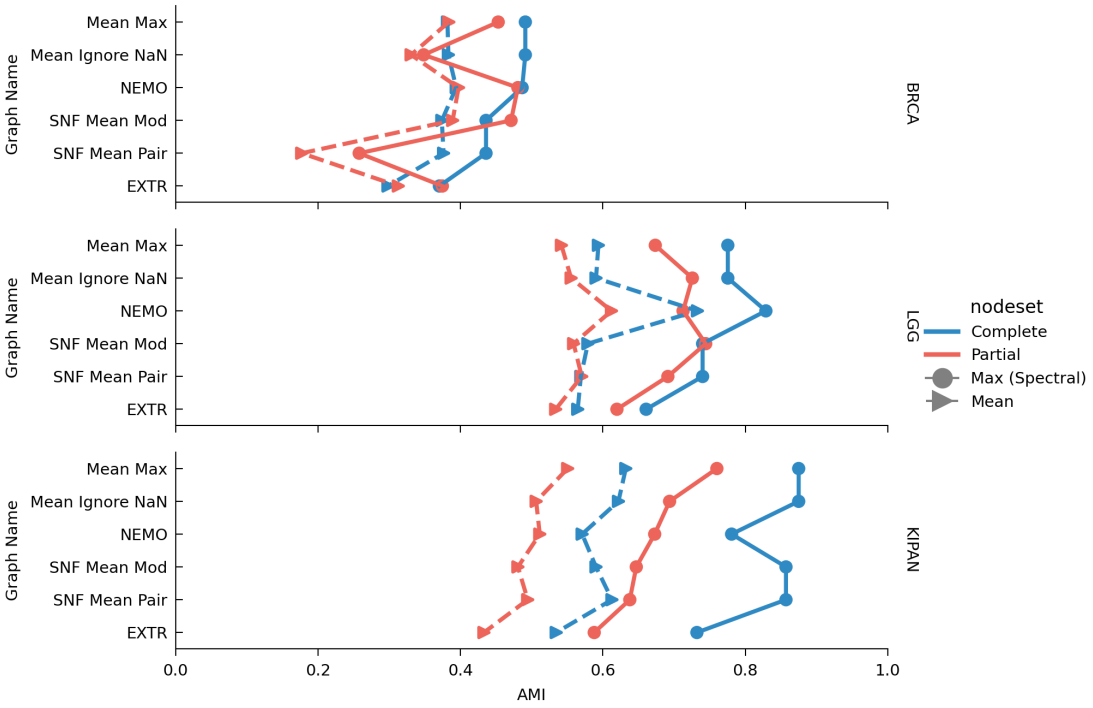

Figure 4.2: Mean and Maximum AMI Performance of Integration Methods on TCGA Datasets. Mean and Maximum Adjusted Mutual Information (AMI) clustering performance of SBM, Leiden, and Spectral algorithms on multi-modal integration networks constructed from complete and partial datasets across three TCGA datasets: BRCA, LGG, and KIPAN. *SNF* struggles with partial data and fails to outperform *NEMO* or *Mean Max* integration methods. *NEMO* consistently outperforms all methods on both complete and partial BRCA and LGG datasets. There is a notable drop in performance on complete KIPAN data where *Mean Max* exhibits superior performance over other methods. The optimal SNF imputation strategy is contingent upon the underlying dataset and selecting an optimal strategy is challenging in unsupervised clustering scenarios.

Cancer Type	BRCA				LGG				KIPAN			
Nodeset	complete		partial		complete		partial		complete		partial	
	Max	Mean	Max	Mean	Max	Mean	Max	Mean	Max	Mean	Max	Mean
Graph												
Mean (Max Dissimilarity)	**0.491**	0.382	0.453	0.384	0.775	0.594	0.674	0.542	**0.875**	**0.632**	**0.760**	**0.551**
Mean (Ignoring NaN)	**0.491**	0.383	0.348	0.331	0.775	0.590	0.726	0.555	**0.875**	0.622	0.694	0.506
NEMO	0.487	**0.394**	**0.481**	**0.397**	**0.829**	**0.733**	0.713	**0.612**	0.781	0.571	0.673	0.511
SNF (Mean per Modality)	0.436	0.374	0.471	0.389	0.740	0.579	**0.744**	0.559	0.857	0.590	0.647	0.481
SNF (Mean Pairwise)	0.436	0.376	0.258	0.177	0.740	0.569	0.691	0.570	0.857	0.613	0.638	0.494
EXTR	0.371	0.298	0.375	0.313	0.661	0.565	0.620	0.534	0.732	0.535	0.588	0.433

Table 4.5: Mean and Maximum AMI Performance of Integration Methods on TCGA Datasets.. Mean and Maximum Adjusted Mutual Information (AMI) clustering performance of SBM, Leiden, and Spectral algorithms obtained on networks constructed by different integration methods— *Mean (Max Dissimilarity)*, *Mean (Ignoring NaN)*, *NEMO*, *SNF (Mean per Modality)*, *SNF (Mean Pairwise)*, and *EXTR* — from complete and partial data across three TCGA datasets: BRCA, LGG, and KIPAN. Notably, NEMO consistently demonstrates strong performance in both complete and partial datasets across multiple cancer types, while SNF's efficacy varies based on imputation strategies and dataset completeness.

Figure 4.3 depicts the clustering performance metrics—(A) AMI, (B) Homogeneity, and (C) the number of predicted clusters—associated with the SBM, Leiden, and Spectral clustering algorithms across the complete and partial BRCA, LGG, and KIPAN datasets. Both SBM and Leiden consistently discover a higher number of subclusters. SBM tends to predict a number of clusters significantly greater than both the ground truth number of clusters and the predictions made by Leiden and Spectral algorithms. Although all algorithms exhibit high homogeneity, indicating consistency within the clusters identified by SBM and Leiden to the ground truth, the AMI scores are reduced due to the higher predicted cluster count. The Spectral algorithm's predicted number of clusters is more accurate, resulting in a lower AMI penalty due to chance correction. This discrepancy in the predicted cluster count among the algorithms influences their respective AMI scores.

While the BRCA problem presents increased challenges with a higher number of ground truth clusters (five subtypes compared to three in the LGG and KIPAN datasets), the lower AMI scores of the algorithms on BRCA compared to LGG and KIPAN are not solely attributable to a more significant chance correction resulting from the increased number of clusters. All three algorithms exhibit low homogeneity scores on BRCA, indicating a lack of consistency within the clusters identified by the algorithms. Unlike the performance of SBM on LGG, where poor AMI scores could be attributed to the splitting of ground truth subtypes into subclusters, the sub-optimal AMI performance on BRCA is primarily due to inaccuracy rather than cluster fragmentation.

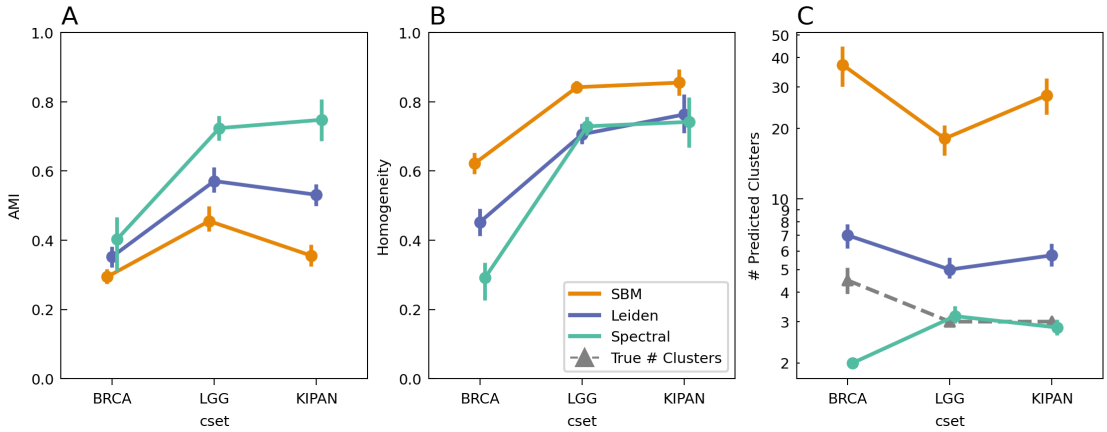

Figure 4.3: Comparison of Clustering Algorithms on TCGA Datasets by AMI, Homogeneity and Number of Predicted Clusters. The (A) AMI, (B) Homogeneity and (C) Number of predicted clusters of the SBM, Leiden, and Spectral clustering algorithms on the complete and partial BRCA, LGG and KIPAN datasets. The reduced AMI of SBM and Leiden is a result of overfitting. They have high homogeneity, an indication that they split the true clusters in subclusters which results in a drop in AMI due to chance correction. Spectral predicts fewer clusters and in two of the datasets actually detects the correct number of clusters. SBM predicts an order of magnitude more clusters than both Leiden and Spectral. The clusters have high homogeneity but SBM has a significant reduction in AMI.

Cluster Prediction

Figure 4.4 demonstrates the effect of graph imputation using networks from each of the integration methods for ground truth subtype prediction in partial datasets. The test set weighted F_1-score of random forest prediction models trained with 5 fold cross validation is shown for each of the TCGA BRCA, LGG and KIPAN datasets. We compare graph-based imputation using integration methods against two benchmarks: complete data prediction and imputation based on mean values. Across all methods, BRCA consistently yields lower F_1 scores, whereas LGG presents the most straightforward prediction task. With the exception of KIPAN, the prediction performance using partial modalities is higher than the complete dataset.

SNF Mean Mod graph imputation outperforms mean value imputation for all datasets although the difference is very slight in the LGG data. NEMO imputation is the best performing imputation method for the BRCA data which aligns with the higher clustering performance of NEMO on BRCA. The success of SNF Mean Mod imputation is surprising given the relatively poor clustering performance in KIPAN and LGG. On LGG, there is minimal difference between graph-based and mean value imputation. This is unsurprising due to the lower rate of partial within LGG.

BRCA, despite the highest rate of partial modality data, shows a clear prediction performance boost using partial data over complete data prediction. While the variance of the models are higher, graph-based imputation consistently outperforms mean imputation across all networks. The improvement in performance over the complete data and mean imputation on the BRCA dataset is especially surprising given the poor clustering performance of the networks.

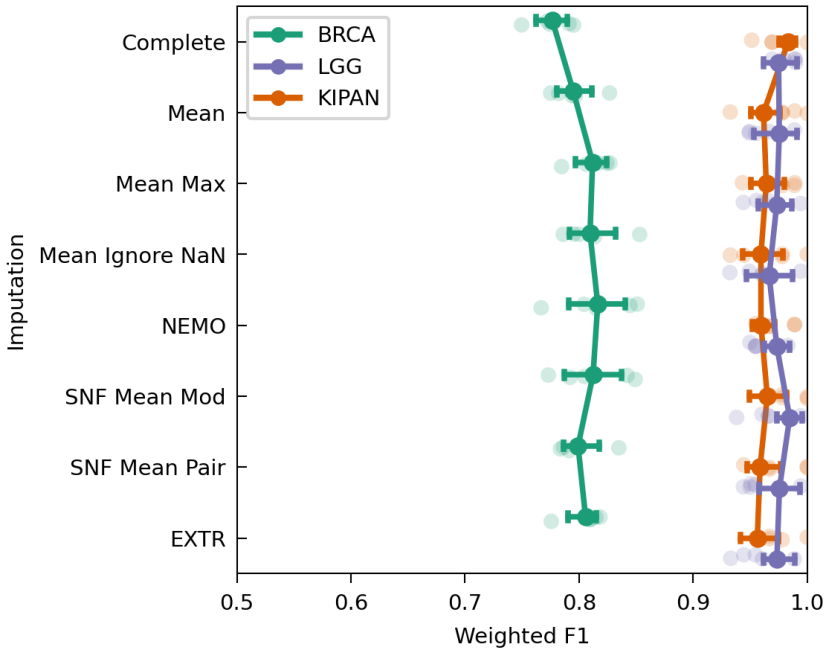

Figure 4.4: Comparison of Imputation using Graph Neighbours on Prediction Performance. Test set Weighted F_1-score of random forest prediction models trained with 5 fold cross validation is shown for each of the TCGA BRCA, LGG and KIPAN datasets. We compare graph based imputation to mean value imputation on partial data and complete data prediction. The prediction of partial data outperforms complete data prediction in BRCA and KIPAN. Graph based imputation outperforms the more naive mean value imputation on KIPAN and BRCA. Both datasets have higher rates of partial data.

Figure 4.5 displays the weighted F_1 scores of the prediction of cluster labels generated by SBM, Leiden, and Spectral clustering algorithms. These scores are derived using 5-fold cross-validated random forest models trained on both partial and complete datasets across the three datasets. The overall prediction performance of all three algorithms remains consistent across the datasets.

Spectral clustering produces the simplest cluster prediction problem across the networks. Despite producing clusters with lower AMI scores than the Spectral clusters, the prediction of Leiden clusters in the LGG and KIPAN datasets is notably accurate and far closer to the performance of Spectral clusters than would be expected given their respective differences in AMI.

The prediction accuracy of the clusters found in the BRCA data is far higher than would be expected given their low AMI scores. In fact the prediction of Leiden and Spectral clusters on *NEMO*, *Mean Max* and *Mean Ignore NaN* are on par or outperform the prediction of the ground truth subtypes.

There is a notable difference in predictability between clusters found on SNF with Pairwise Mean imputation and Modality Mean imputation methods. In particular, there is a significant drop in the F_1-score of the Leiden algorithm across all three datasets. In contrast to its clustering performance, Mean Max clusters exhibit high predictability consistently across all datasets.

Figure 4.6 depicts the relationship between Adjusted Mutual Information (AMI) scores obtained from SBM, Leiden, and Spectral clusters generated by our multi-modal integration methods, alongside the weighted F_1 scores achieved by a model trained to predict these clusters. This plot highlights a noticeable correlation (0.71) between AMI scores and cluster predictability, the strength of the correlation varies based on cluster type. Within SBM clusters in particular, higher AMI is strongly linked to elevated weighted F_1 scores and higher predictability.

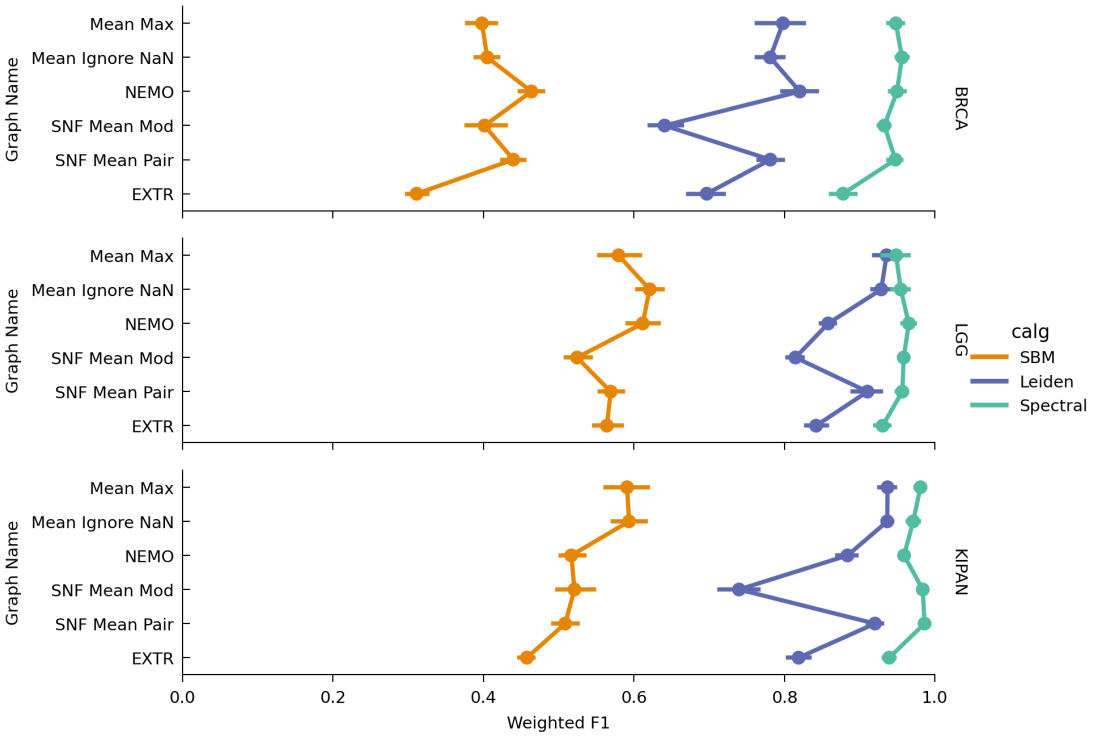

Figure 4.5: Predictability of Clusters Detected by SBM, Leiden and Spectral Algorithms on TCGA Datasets. Weighted F_1 scores of the prediction of cluster labels generated by SBM, Leiden, and Spectral clustering algorithms. These scores are derived using 5-fold cross-validated random forest models trained on both partial and complete datasets across the three datasets. The predictability of each clustering algorithm remains consistent across datasets. Leiden clusters found on SNF Mean Mod networks are harder to predict than SNF Mean Pair despite the poorer AMI score of SNF Mean Pair compared to SNF Mean Mod.

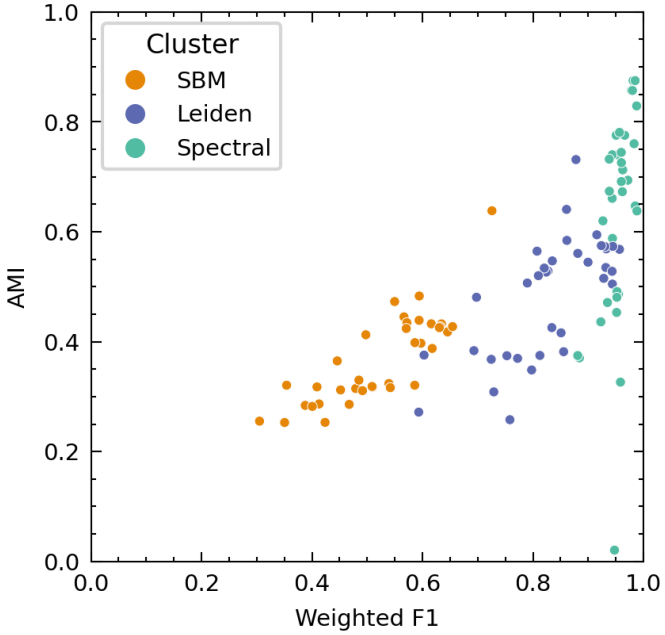

Figure 4.6: Comparison of Cluster AMI Performance and Predictability on TCGA Datasets. Cluster AMI score of SBM, Leiden and Spectral compared to the cross validated weighted F_1-score of models trained to predict cluster label. There is a strong correlation between cluster predictability and AMI score (0.71).

Feature Importance

Table 4.6 presents the mean and maximum AMI clustering performance achieved by SBM, Leiden, and Spectral clustering on networks constructed from individual modalities, DNAm, mRNA, miRNA, CNV and RPPA, within the complete BRCA, LGG, and KIPAN datasets. Across both BRCA and LGG datasets, no individual modality network matches the clustering performance attained by the multi-modal integration methods with the notable exception of the miRNA network on KIPAN. CNV consistently exhibits robust performance across all datasets. Furthermore, distinct modalities emerge as informative within different datasets: mRNA, DNAm, and CNV closely align with the ground truth in LGG; miRNA, CNV, and RPPA show strong consistency with the ground truth in KIPAN; and CNV stands out as the sole modality detecting a signal, albeit still a poor AMI score, within the BRCA dataset.

The clustering performance of single modality networks are surprising when considering the underlying nature of the ground truth subtypes. While the robust performance of mRNA and CNV in LGG aligns with expectations—given that LGG subtypes rely on codeletion and gene expression—it's surprising that the mRNA network in BRCA isn't more informative. mRNA is the worst performing network alongside DNAm. The PAM50 classification depends on the expression of a specific set of 50 genes within mRNA. The under-performance of the single modality mRNA networks highlights the difficulties associated with noise in gene expression

data and underscores the need for feature selection to extract meaningful subtypes. Furthermore, the lack of feature selection in this work might also contribute to the underwhelming performance of DNAm networks across all three datasets.

Cancer Type	BRCA		LGG		KIPAN	
	Max	Mean	Max	Mean	Max	Mean
Modality						
DNAm	0.271	0.184	0.622	0.552	0.348	0.296
mRNA	0.271	0.217	**0.697**	**0.579**	0.321	0.295
miRNA	0.297	0.280	0.230	0.172	**0.873**	**0.635**
CNV	**0.462**	**0.378**	0.606	0.514	0.621	0.483
RPPA	0.331	0.279	0.409	0.252	0.680	0.536

Table 4.6: Comparison of AMI Performance on Single Modality Networks. Mean and Maximum AMI clustering performance achieved by SBM, Leiden, and Spectral clustering on networks constructed from individual modalities, DNAm, mRNA, miRNA, CNV and RPPA, within the complete BRCA, LGG, and KIPAN datasets. Single modality networks fail to match the performance of clustering on multi-modal networks shown in Table 4.5.

Figure 4.7 illustrates the distribution of the top 10% (32) most influential features across modalities in cross-validated random forest models for predicting the ground truth, SBM, Leiden, and Spectral clusters. We restrict our analysis to the clusters identified by the highest-performing networks, specifically *Mean Max*, *NEMO*, and *SNF Mean Mod*[9]. Additionally, we display the average influence of each modality across all cluster types.

Of the three clustering algorithms, the origin of the features identified by Spectral clustering aligns closest with that of the ground truth clusters. Leiden and SBM place higher importance in other modalities to the ground truth and spectral clusters. The is most notable in the BRCA dataset where DNAm, mRNA and RPPA have increased importance in SBM and Leiden models. The differences between the ground truth and detected clusters is highest in KIPAN, the models of all three cluster algorithms identify mRNA and miRNA as having higher importance than the ground truth clusters which make use of RPPA.

The feature importance in ground truth cluster prediction does not align with the highest performing modalities for cluster detection in LGG and KIPAN. CNV and RPPA have the highest importance in LGG yet the clusters identified on DNAm and mRNA networks were more accurate than both. Similarly in KIPAN, the miRNa network detected the clusters with high accuracy, its AMI score was on par with multi-modal methods yet it has the joint fewest

9. The ground truth prediction models that are included used graph imputation based on these networks

features included in the top 10% of ground truth cluster prediction. In BRCA, the poorest performing networks are those of the DNAm and mRNA modalities. These modalities have higher importance in all four cluster types.

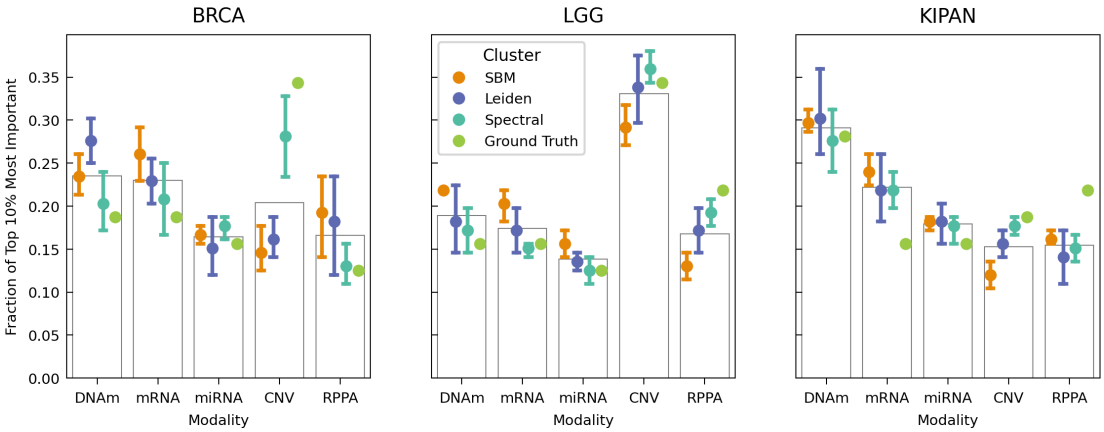

Figure 4.7: Distribution of Top 10% Most Important Features in Cluster Label Prediction Across Modalities. Distribution of the top 10% (32) most influential features across modalities in cross-validated random forest models for predicting the ground truth, SBM, Leiden, and Spectral clusters. Our analysis is restricted to the prediction of clusters identified by the highest-performing networks, specifically *Mean Max*, *NEMO*, and *SNF Mean Mod*. The feature importance in ground truth cluster prediction does not align with the highest performing modalities for cluster detection in LGG and KIPAN seen in Table 4.6.

Figure 4.8 displays the differences in the distribution of the top 10% (32) most influential features across modalities between the complete and partial datasets. Feature importance is extracted from cross-validated random forest models predicting SBM, Leiden, and Spectral clusters on *Mean Max*, *NEMO*, and *SNF Mean Mod* networks. While the origin of influential factors is generally consistent across most modalities between the two datasets, several differences emerge. DNAm has heightened importance in predicting partial data on KIPAN, while mRNA's influence diminishes, and RPPA gains importance when predicting partial BRCA compared to complete data. LGG factors exhibit sustained consistency across nodesets. This consistency across nodesets is not unexpected due to the lower prevalence of partial data.

The variability within the importance of BRCA factors is notably higher within both nodesets. By comparison, KIPAN and LGG factors are far more uniform. This increased variability could be attributed to two potential reasons: noisy levels of feature importance across models in BRCA; or second, substantial differences between clusters identified within BRCA data. First, there is relative equality in rate of importance among modalities within BRCA. The difference in level of importance of the top 5-15% most important features might be minimal and so the order changes significantly from model to model causes significant changes to the distribution

of the origin of the features. Secondly, the lower homogeneity and AMI of BRCA clusters compared to the ground truth clusters, unlike LGG and KIPAN datasets, suggest substantial inter-cluster differences within BRCA.

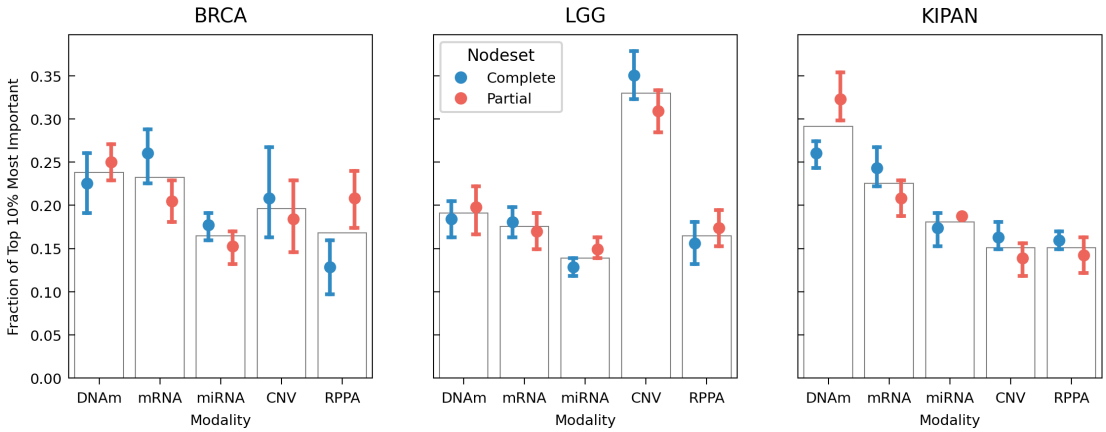

Figure 4.8: Distribution of Top 10% Most Important Features in Cluster Label Prediction Across Modalities for Complete and Partial TCGA Datasets. Comparison of the distribution of the top 10% (32) most influential features across modalities between Partial and Complete data in cross-validated random forest models for predicting SBM, Leiden, and Spectral clusters. The included models are restricted to the prediction of clusters identified by the highest-performing networks, specifically *Mean Max*, *NEMO*, and *SNF Mean Mod*. The variance of distribution of factors within both nodesets is higher in BRCA. Two possible for this increase in variance is a lack of agreement between the clustering algorithms or minimal differences in the importance of modalities resulting in noisy ordering of the features.

Effect of including partial data in analysis

Figure 4.9 compares partial and complete data clustering performance by nodetype through the mean AMI scores generated by SBM, Leiden, and Spectral clustering algorithms on multi-modal integration networks across BRCA, LGG, and KIPAN datasets. We show the overall AMI scores for partial data (*y — Partial*) and complete data (*y — Complete*). We also examine the clustering performance of different nodes in *y — Partial*. Specifically, the AMI score between predicted and ground truth labels for nodes exclusive to partial data (*y — P only*) and nodes present in both the partial and complete datasets (*y — P & C*). Additionally, the AMI agreement among between the predicted clusters on the complete data and the predicted clusters on the partial data is shown for nodes present in both partial and complete datasets (*P & C consensus*). This allows us to examine whether the inclusion of partial nodes cause a drop in the clustering accuracy of the complete nodes (*y — Complete* vs *y — P & C*).

In the BRCA clusters, the performance of partial only nodes is better than that of the complete nodes in both datasets. The consensus is very poor amongst the complete nodes. Highlighting the algorithms struggle to identify a consistent set of clusters.

The clusters identified on Mean Max networks are the most consistent between the complete and partial data across the three datasets. The inclusion of partial data does not alter the network structure in such a way as to split or change the clusters to the same extent found in other algorithms. In some instances this lack of agreement is due to improved accuracy e.g SNF Mean mod on KIPAN but often a reduction in consensus coincides with poorer accuracy e.g. NEMO on LGG and Mean Ignore NaN on BRCA.

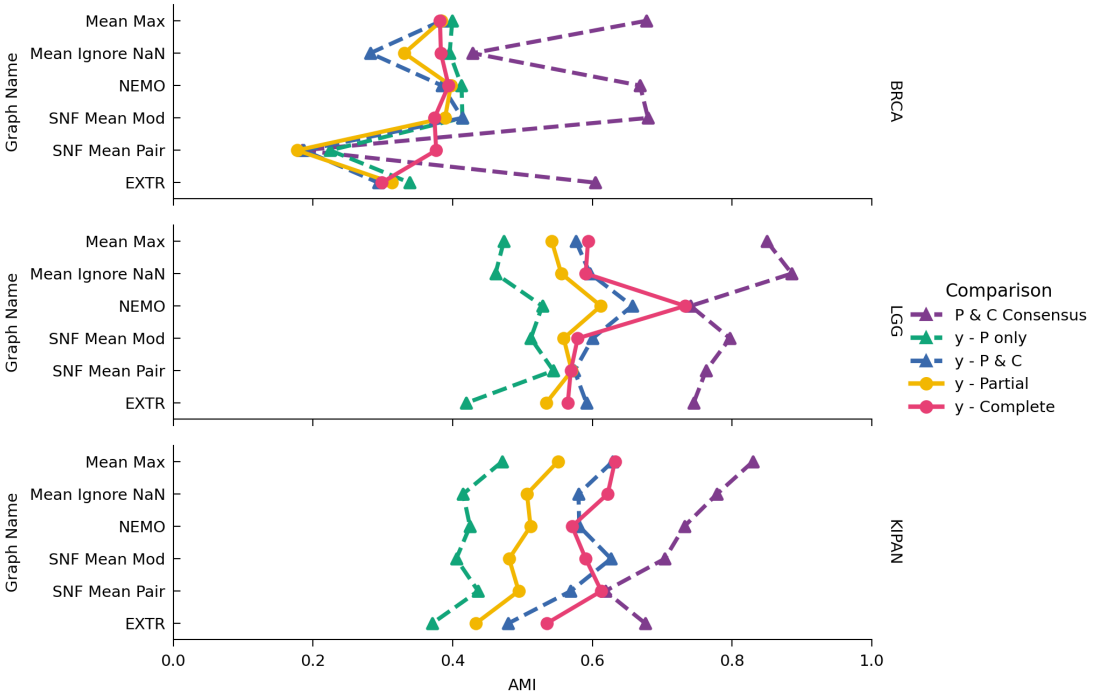

Figure 4.9: AMI Clustering Performance for Partial and Complete TCGA Datasets by Nodetype. Breakdown of partial and complete data clustering performance by nodetype through the mean AMI scores generated by SBM, Leiden, and Spectral clustering algorithms on multi-modal integration networks across BRCA, LGG, and KIPAN datasets. We show the AMI scores for partial data (*y — Partial*), complete data (*y — Complete*), a breakdown of *y — Partial* based on nodetype, nodes exclusive to partial data (*y — P only*) and nodes present in both partial and complete datasets (*y — P & C*), and the AMI agreement of complete nodes between their partial and complete clusters (*P & C consensus*). Adding nodes with partial data to the network does not reduce the clustering performance of nodes with a complete set of measurements.

Figure 4.10 shows the comparison of ground truth cluster prediction in both complete and partial data using mean and graph-based imputation. We show the weighted F_1 scores for predicting ground truth clusters across the BRCA, LGG, and KIPAN datasets, categorised by nodetype. We show the weighted F_1 scores for partial data (*y — Partial*) and complete data (*y — Complete*). We also examine the prediction performance of different nodes in *y — Partial*. Specifically, the weighted F_1-score between predicted and ground truth labels for nodes exclusive to partial data (*y — P only*) and nodes present in both the partial and complete datasets (*y — P & C*). Additionally, the weighted F_1-score agreement among between the predicted clusters on the complete data and the predicted clusters on the partial data is shown for nodes present in both partial and complete datasets (*P & C consensus*). This allows us to examine whether the inclusion of partial nodes cause a drop in the prediction accuracy of the complete nodes (*y — Complete* vs *y — P & C*).

In the LGG and BRCA datasets, the prediction of the labels of complete data improves with the inclusion of partial data (*y — P & C* vs *y — Complete*). Mean Max imputation in KIPAN also improves the prediction of complete node classes. The prediction of partial nodes is far worse than that of complete data. Only within BRCA is the prediction of partial data is higher than the prediction of complete nodes, for both KIPAN and LGG the performance is far worse. While a weighted F_1-score of 0.93 still indicates strong performance, the performance is significantly worse than that that of the complete nodes within these datasets.

Partial only data prediction (*y — P only*) using Mean imputation often performs on par with graph based imputation or outperforms graph based imputation. Yet this comes at cost as Mean imputation reduces the performance of the complete data nodes as seen in BRCA and KIPAN. In KIPAN, this success of partial only data prediction does not result in a drop in performance relative to the graph based imputation methods.

SNF Mean Pair has poor partial data prediction in BRCA and KIPAN as a result of difficulties in predicting partial only nodes. It is the converse of Mean imputation that predicts partial nodes quite well but at the cost of poor complete node prediction. SNF is very successful in partial data only prediction. It is consistently the best performing graph based method. This contrasts with its clustering performance. NEMO for example has significantly higher clustering AMI score on LGG as seen in Figure 4.9. SNF Mean Mod also struggles with poor predictability of its clusters see Figure 4.5. The difference in performance is unexpected but highlights the optimal choice of network for clustering performance may not align with optimal choice of network in other applications.

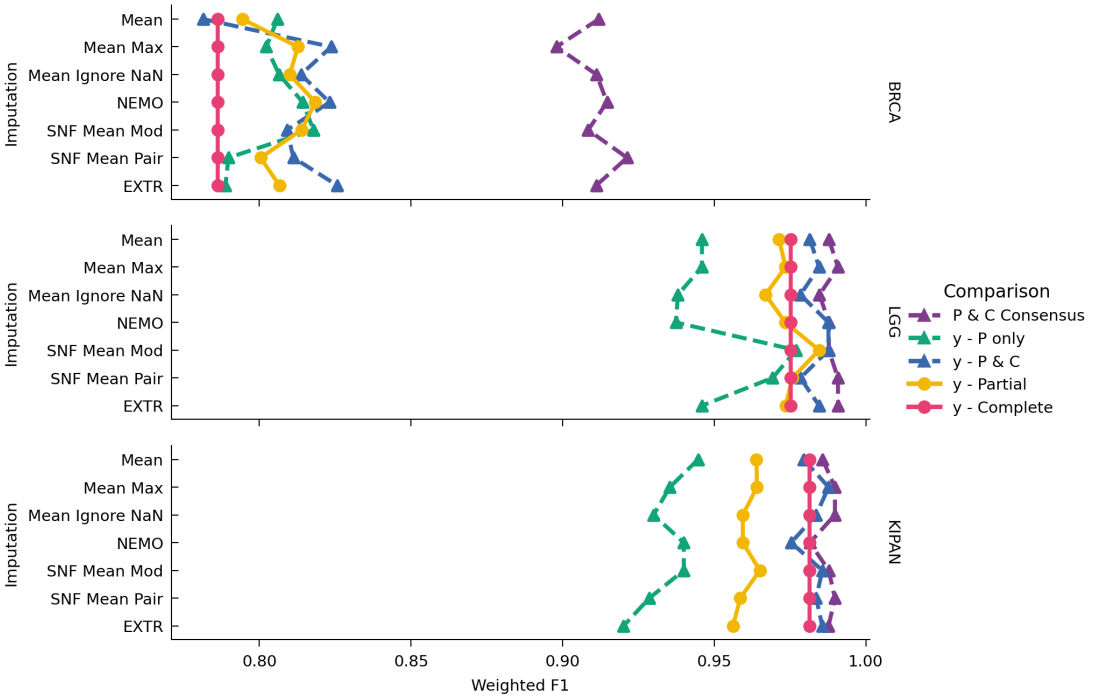

Figure 4.10: Weighted F_1 Prediction Performance for Partial and Complete TCGA Datasets by Nodetype. Breakdown of partial and complete data ground truth prediction performance by nodetype through the mean weighted F_1 scores on multi-modal integration networks across BRCA, LGG, and KIPAN datasets. We show the weighted F_1 scores for partial data (*y — Partial*), complete data (*y — Complete*), a breakdown of *y — Partial* based on nodetype, nodes exclusive to partial data (*y — P only*) and nodes present in both partial and complete datasets (*y — P & C*), and the F_1-score agreement of complete nodes between their partial and complete clusters (*P & C consensus*). The prediction of nodes with complete data improves significantly with inclusion of partial data in training of the prediction model on TCGA BRCA.

4.5.2 SSC

Table 4.7 show the average and maximum AMI clustering performance of SBM, Leiden, and Spectral clustering within multi-modal integration networks across the complete and partial data within the SSC for two different metrics — correlation and euclidean. Similar to the TCGA, consistently emerges as the top-performing integration method across both metrics in both the complete and partial datasets. Its mean clustering performance is particularly strong. It notably outperforms all methods when calculating similarity using the euclidean metric.

SNF continues to struggle with partial data. The performance is inconsistent. When the algorithm successfully incorporates the partial data, it performs very strongly. For example, SNF Mean Pair using the correlation metric. When a large number of partial modalities are present, as in TCGA BRCA and the SSC data, the partial data corrupts the network structure leading to poor clustering performance. This can be seen in SNF Mean Mod using both correlation and euclidean metrics and SNF Mean Pair using the euclidean metric.

Figure 4.11 visualises the clustering performance of the integration methods using the correlation metric. It shows average and maximum AMI clustering performance of SBM, Leiden, and Spectral clustering within multi-modal integration networks across the complete and partial SSC data using correlation metric. The drop in performance of SNF Mean Mod with the inclusion of partial data is particularly notable. It is outperformed by the consistently poor performing Extreme Mean method when incorporating partial modalities.

Metric	Correlation				Euclidean			
nodeset	Complete		Partial		Complete		Partial	
	Max	Mean	Max	Mean	Max	Mean	Max	Mean
Graph Name								
Mean Max	**0.860**	0.558	0.561	0.444	0.804	0.509	0.534	0.250
Mean Ignore NaN	**0.860**	0.558	0.765	0.500	0.804	0.523	0.777	0.490
NEMO	0.826	**0.629**	**0.802**	0.507	**0.853**	**0.643**	**0.801**	**0.521**
SNF Mean Mod	0.857	0.572	0.277	0.164	0.811	0.582	0.270	0.161
SNF Mean Pair	0.857	0.611	0.796	**0.526**	0.811	0.583	0.227	0.098
EXTR	0.718	0.527	0.666	0.423	0.653	0.487	0.605	0.400

Table 4.7: Mean and Maximum AMI Performance of Integration Methods on SSC Data. Average and maximum AMI clustering performance of SBM, Leiden, and Spectral clustering within multi-modal integration networks across the complete and partial data within the SSC using correlation and euclidean metrics. NEMO is the most consistent model and achieving high performance across both metrics and both partial and complete nodetypes. SNF is unstable when incorporating partial modalities and often leads to drastic decline in network structure.

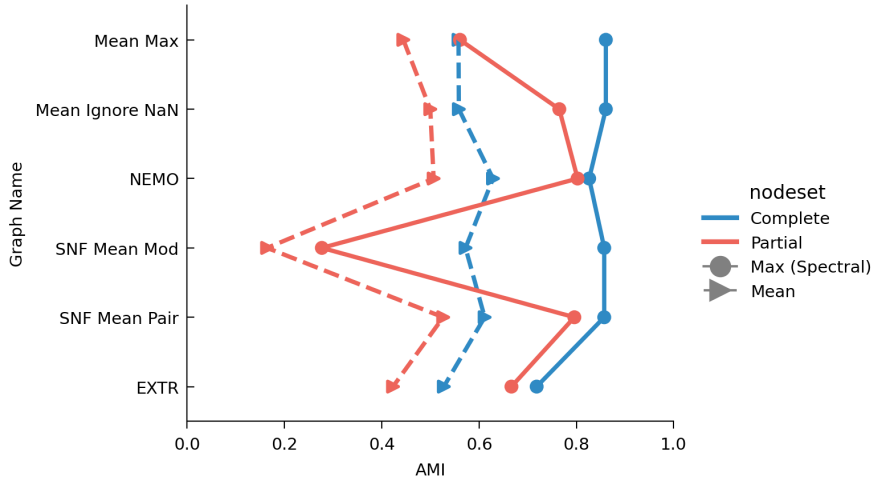

Figure 4.11: Mean and Maximum AMI Performance of Integration Methods on SSC Data. Average and maximum AMI clustering performance of SBM, Leiden, and Spectral clustering within multi-modal integration networks across the complete and partial data within the SSC using correlation metric. NEMO performs consistently well across nodesets show high maximum and mean clustering performance. The drop in performance of SNF with the inclusion of partial data is inconsistent.

Figure 4.12 shows the (A) AMI, (B) Homogeneity and (C) number of predicted clusters of the SBM, Leiden, and Spectral clustering algorithms on the complete and partial SSC data. SBM, Leiden and Spectral display the same properties found within the TCGA data and synthetic datasets used in previous chapters. The number of clusters detected by each method is consistent. Again Spectral performs best for high level splits. For example, here we have two large ground truth clusters of c.2800 nodes. Spectral finds a highly accurate split of these clusters. Leiden still detects large clusters but splits them into subclusters. Across both TCGA and SSC, SBM does not detect large clusters. The average cluster size of SBM clusters across the SSC networks is 45 individuals within the partial data ($N = 5615, N_c = 124$) and 37 individuals within the complete data ($N = 1435, N_c = 38$).

Figure 4.13 shows the comparison of ground truth cluster prediction in both complete and partial data using mean and graph-based imputation. We show the weighted F_1 scores for predicting ground truth clusters within the SSC data, categorised by nodetype. We show the weighted F_1 scores for partial data (*y — Partial*), complete data (*y — Complete*), and a breakdown of *y — Partial* based on nodetype distinctions, specifically nodes exclusive to partial data (*y — P only*) and nodes present in both partial and complete datasets (*y — P & C*). Finally, the weighted F_1-score agreement between partial and complete predictions (*P & C consensus*) is shown. Again the complete dataset shows improved prediction when partial data is included (*y — P & C* vs *y — Complete*). This can be seen using both the NEMO and Extreme Mean imputation. The increased number of observations introduces more comprehensive covering of the feature space.

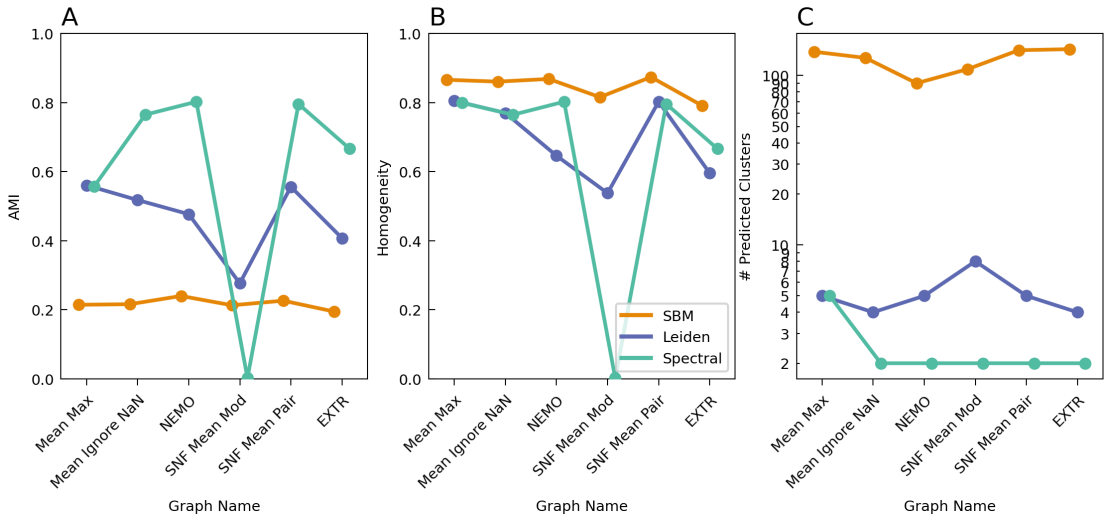

Figure 4.12: Comparison of Clustering Algorithms on SSC Datasets by AMI, Homogeneity and Number of Predicted Clusters. The (A) AMI, (B) Homogeneity and (C) number of predicted clusters of the SBM, Leiden, and Spectral clustering algorithms on the complete and partial SSC data. Again Leiden and SBM discover a more fine grained split of the data with larger number of predicted clusters. The contrast between SNF Mean Mod and SNF Mean Pair is stark. SNF Mean pair discovers clusters with high homogeneity across all three algorithms while SNF Mean Mod fails to separate siblings and probands.

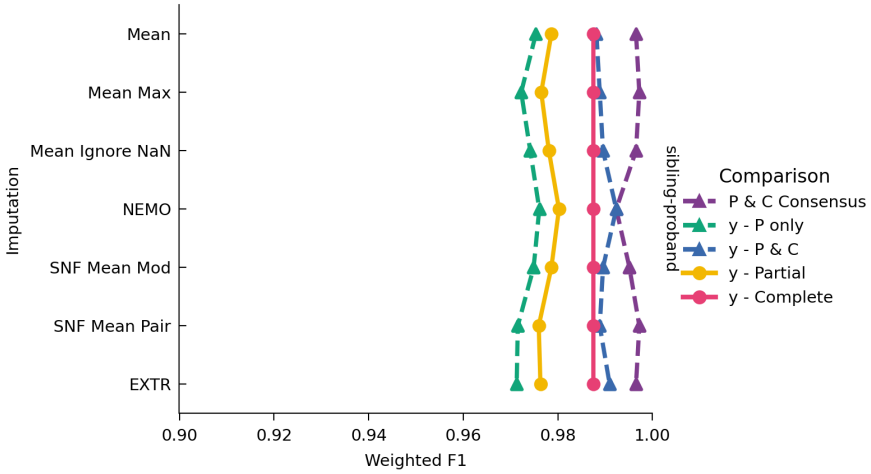

Figure 4.13: Weighted F_1 Prediction Performance for Partial and Complete SSC Data by Nodetype. Comparison of ground truth cluster prediction in both complete and partial data using mean and graph-based imputation on SSC data. We show the weighted F_1 scores for partial data (*y — Partial*), complete data (*y — Complete*), a breakdown of *y — Partial* based on nodetype, nodes exclusive to partial data (*y — P only*) and nodes present in both partial and complete datasets (*y — P & C*), and the F_1-score agreement of complete nodes between their partial and complete clusters (*P & C consensus*). Prediction performance is highly accurate using all imputation methods with all achieving a weighted F_1-score > 0.98. The prediction of complete nodes improves with the inclusion of partial data using NEMO imputation.

Figure 4.14 shows a wordcloud visualisation of top 15 most informative variables used in the prediction of A) Ground Truth, B) SBM, C) Leiden and D) Spectral clusters on the NEMO network created on the complete set. These are the variables with the highest feature importance across the cross validated random forest models used in the prediction of each cluster. The size of the text of each feature is weighted by its importance. There is strong consistency across the A) Ground Truth, C) Leiden and D) Spectral clusters. Summary scores associated with SRS Parent, SCQ Parent and Vineland II modalities are highly important and effectively distinguish between cluster labels. There is more variability in the importance of features within SBM clusters. Few features are of large size and the origin of the features differ from the other clusters with Vineland II feature ranking highly.

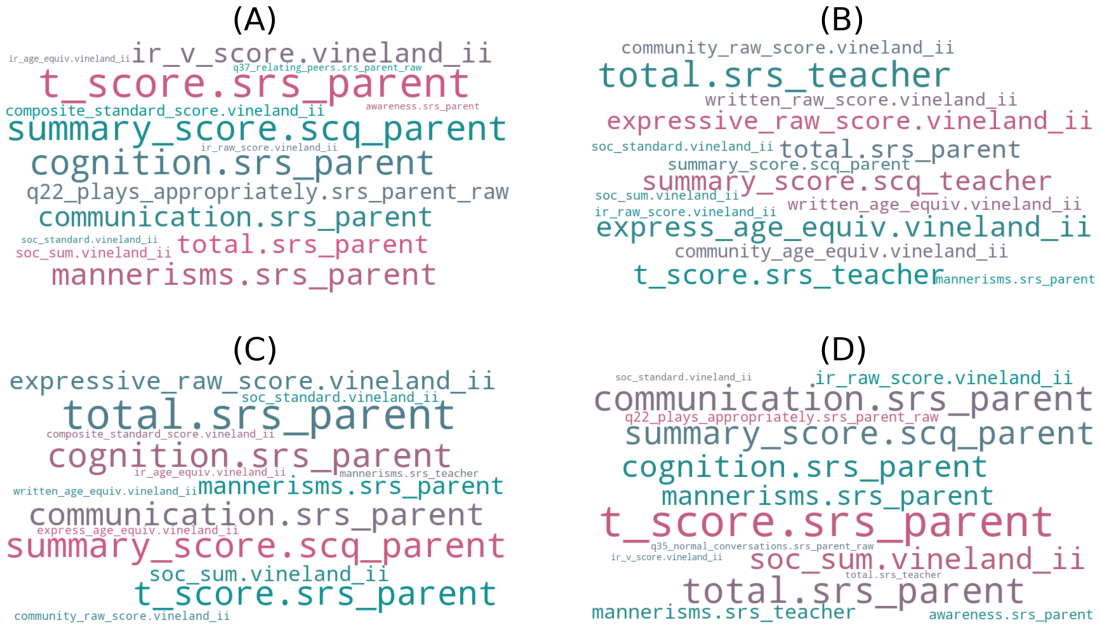

Figure 4.14: Top 15 Most Informative Variables in Cluster Label Prediction. Wordcloud of top 15 most informative variables in the prediction of A) Ground Truth, B) SBM, C) Leiden and D) Spectral clusters on the NEMO network created on the complete set. The size of the visualised feature name corresponds to its relative importance. We can see the overall summary scores of SRS Parent and SCQ Parent are highly informative for Ground Truth, Leiden and Spectral clusters. For SBM clusters, the SRS Teacher and Vineland scores are more informative, reinforcing the differences found in the AMI scores between the detected clusters.

4.6 Discussion

While maximum accuracy of a set of clustering labels holds paramount importance in specific applications, mean performance of several clustering algorithms serves as a more indicative measure of the overall quality of a network generated by an integration method. The choice of optimal algorithm depends heavily on the application. A network where all clustering algorithms perform strongly is far more flexible and robust. This is highly beneficial in scenarios where ground truth labels are unavailable and algorithm selection is difficult.

Among the TCGA datasets, BRCA has the most challenging clustering problem. The mean performance of all algorithms is lowest on BRCA. There are a higher number of clusters to detect and the difference between the subtypes is based on gene expression of a set of 50 genes. The subtypes in LGG and KIPAN arise as a result of more significant adjustments such as chromosomal deletion and histological features. The distribution differences between subtypes is more significant in these datasets and the accuracy of the clustering algorithms and integration methods reflect that.

The difficulty of the KIPAN and LGG clustering problems are similar; both have three ground truth subclusters that are detected very successfully by spectral clustering and the homogeneity of clustering algorithms on both datasets are similar (Figure 4.3. However, the average AMI score is higher in LGG. This difference is a result of the number of clusters predicted by SBM and Leiden in LGG. Both algorithms detect fewer clusters which more closely aligns with the actual number of clusters, resulting in a higher AMI.

Despite its complexity, SNF fails to exhibit substantial improvements in clustering accuracy compared to simpler methods. Both NEMO (a mean of per-modality KNN networks) and Mean Max (a simple mean of per-modality similarity scores) either match or surpass SNF's performance across both the SSC data and the three TCGA datasets (BRCA, KIPAN, and LGG). SNF encounters challenges with partial data. In SNF, the absence of a node within a single modality influences its neighbours in other modalities during the diffusion process. When SNF successfully incorporates partial data, its performance is quite strong – for example, SNF Mean Pair on SSC using correlation (Table 4.7) and SNF Mean Mod on TCGA BRCA (4.5). Chapter 3 highlights a critical threshold where SNF's clustering performance drops abruptly in partial data. This phenomenon is visible in both the TCGA and SSC data. For instance, in partial BRCA data, SNF Mean Pair performance plummets dramatically compared to SNF Mean Mod (Figure 4.2). In the SSC data, the choice of similarity metric plays a role; SNF performance is poor in methods using Euclidean distance. Although SNF occasionally

performs on par or even outperforms other methods – for example, SNF Mean Mod on partial LGG data showed the highest Max AMI score (Table 4.7) – this inconsistency poses a significant drawback in unsupervised clustering endeavours.

The KNN step in NEMO integration serves as a filtering mechanism when aggregating similarity scores across modalities. This proves advantageous when dealing with nodes that consistently emerge as each other's nearest neighbours across most modalities but exhibit some noisy pairwise similarities in others. The aggregation of KNN networks helps avoid the incorporation of outlier high dissimilarities within a single modality, enabling a more effective calculation of overall similarity. However, its effectiveness diminishes when the nearest neighbours included in the KNN across modalities are not consistent. If the set of nearest neighbours in each modality is different, after constructing the KNN, the mean calculation does not have access to the similarity values of these nodes in other modalities. This leads to a more random nearest neighbour selection process, reducing performance. In such instances, Mean Max excels by aggregating all similarity values without disregarding a node's pairwise similarity in any modality. This contrast is evident in the results of Chapter 3 on *Merged* clusters (Table 3.3), where within-cluster similarity remains consistent across modalities, yet nearest neighbours selected in one modality might belong to other clusters. This likely explains the variance in NEMO's performance between the KIPAN and LGG datasets, where Mean Max performs well.

From the performance of the clustering algorithms on the TCGA data, we can see that the optimal choice of clustering algorithm heavily depends on the problem application. The number of clusters predicted by each method is consistent across the datasets used in this chapter and the previous chapters. For example, Spectral tends to perform a high level split of the data and predicts very few clusters, only two or three. In settings with limited number of ground truth clusters, it performs quite well as can be seen in the *Equal 3* clustering problem in Chapter 2, the TCGA data shown here and the SSC cohort. Leiden typically predicts c.5-10 clusters. It scores a high AMI score in Chapters 2 and 3 as a result of problems with 10 ground truth clusters. In contrast, SBM often predicts a far higher number of clusters. While all methods are consistent in their approach, the optimal choice depends on the specific objectives. If the aim is to discover or gain an understanding of a subpopulation, an algorithm that perform a larger number of splits such as Leiden or SBM may be more suitable.

The performance of single-modality networks highlights the advantages of multi-modal integration in tumour subtype analysis. Although single-modality networks occasionally perform comparably—such as the performance of miRNA on KIPAN—multi-modal clustering consistently surpasses the performance of individual modality approaches (Table 4.6). Moreover, the advantages of multi-modal integration become even more apparent with partial data. Single-modality networks are restricted to nodes with observations in that particular modality[10]. Multi-modal analysis can leverage partial data that might be absent from a number of modalities.

The results on the SSC dataset align with those on TCGA. Again NEMO emerges as a highly beneficial integration method that produces networks the enable accurate clustering. On the complete data the differences between SNF, NEMO and Mean S_i[11] are negligible. The differences on partial data is highly significant. NEMO and SNF Mean Pair emerge as significantly more accurate. SNF Mean Pair outperforming SNF Mean Mod is noticeable given the reverse was true on TCGA data. Similar to Chapter 3, Mean ignoring NaN was highly accurate and outperformed Mean Max.

A key benefit of multi-modal network construction is the ability to improve prediction models through the inclusion of partial data. While imputation strategies exist for *item non-response*, *unit non-response* and partially complete data is typically omitted from analysis. K-nearest neighbour imputation is well established Troyanskaya et al. (2001) for handling missing feature values. Here, I show the benefit of KNN imputation for *unit non-response* using constructed networks. Figure 4.4 shows a clear improvement in prediction performance from my integrated networks over a model trained on the complete dataset. Even more remarkable, this benefit is not limited to prediction, community detection with partial included can improve the clustering of the complete data. Even if partial data is omitted from final analysis, the inclusion of partial data in network construction and model training offers clear benefits. Figures 4.9 & C.1 both display improvements in the complete cluster detection (y — *P & C*), while Figures 4.10 & 4.13 display clear improvements in complete cluster prediction (y — *P & C*). The accuracy of cluster detection in partial data can vary significantly, this is particularly clear in the TCGA dataset, but the prediction benefits are far more consistent. While doubts may remain over the inclusion of partial data in formal models, this analysis suggests it can provide significant benefits.

SNF shows consistent struggles with partial data. In Chapter 3, I hypothesised poor performance was a result of a conservative imputation strategy. This was partially true. SNF showed significant improvements with both Mean Mod and Mean Pair imputation approaches displaying comparable performance to NEMO, Mean ignoring NaN and Mean Max across the TCGA and SSC datasets. However, the instability visible in Section 3.5.3 remains. At different points both SNF Mean Mod and SNF Mean Pair have shown complete collapse in clustering

10. In this analysis, I enforce an extra limitation by restricting modalities to nodes in the complete dataset.
11. Mean Max and Mean ignoring NaN are identical without partial data.

performance (SNF Mean Mod on TCGA BRCA in Figure 4.2 and SNF Mean Pair on SSC in Figure 4.11). On SSC using euclidean metric both imputation approaches failed to embed cluster information in the network (Table 4.7). It is not clear, *a priori*, when one imputation strategy is preferable to the other. This is a significant drawback when ground truth labels are unavailable and the performance of each strategy cannot be evaluated — a common scenario in unsupervised clustering. This inability to incorporate partial data is unsurprising. The imputed value is not isolated to the individual missing from a modality, it spreads to its nearest neighbours in other modalities, affecting the wider neighbourhood. One of the key benefits of multi-modal network construction is the ability to incorporate partial data and SNF inability to do so is a significant drawback.

4.6.1 Limitations

The TCGA analysis presented here is significantly hampered by the absence of feature selection. In many multi-omic pipelines, common practice involves leveraging techniques like differential expression or lasso regression to identify features associated with a particular target feature. My objective, however, was to assess unsupervised clustering. This prevented the use of ground truth labels as a target, as incorporating them would artificially inflate clustering performance. The current extensive number of features, especially in the DNAm and mRNA modalities, raises concerns on the biological relevance of the produced clusters. There is a danger the clusters are associated with random noise. Additionally, a consequence of the lack of feature selection is the use of PCA in my ground truth prediction models. Effective feature selection would enhance my prediction model by allowing the incorporation of biologically relevant features, benefiting downstream tasks such as feature importance analysis. The challenge would be the identification of an informative target feature.

An additional limitation in the presented analysis is the reliance on the predictability of a cluster as an indirect measure of successful cluster detection. As shown, the predictability of a cluster is strongly correlated with the number of clusters detected. Without a detailed exploration of the features assigned to each cluster, it becomes challenging to assert that the clustering algorithms are identifying meaningful clusters. This concern applies to both Spectral, which detects a minimal number of clusters, and SBM, which detects a high number of clusters. The success of Spectral may be potentially overestimated, while SBM's performance could be underestimated due to this indirect evaluation approach.

As mentioned, my (feature importance) analysis was conducted on PCA features. Extracting biological meaning from PCA features is challenging, given that each PCA feature is derived from a linear combination of the original features. Moreover, the utilisation of the origin of important features as a statistic has significant limitations. It neglects the varying levels of importance associated with each feature, merely counting the original modality of the top 10% ranked features. For instance, consider a scenario where one feature from *Modality A*

alone distinguishes between clusters with 100% accuracy. If the remaining features in the top 10% originate from *Modality B*, the second modality will appear as more important even if the *Modality B* features cannot distinguish between clusters at all. While this statistic can provide an indication of the relative importance of modalities within a model, it should not be solely relied upon for drawing definitive conclusions.

A final limitation in this analysis is the use of the *Unaffected Sibling* control group in SSC. Subtyping within a condition is a far more challenging clustering problem than differentiation between individuals with a condition and individuals without. As seen within the TCGA, subtyping of BRCA is far more challenging than subtyping KIPAN which comprised of subtypes with significant histological differences. A further limitation imposed by the lack of ground truth subtype targets in SSC is limitations in available data. The control group is limited to a subset of the measurements undertaken by the proband. The most significant diagnostic measures absent are the ADOS and ADI-R (see Table C.1). Both questionnaires form key components of an ASD diagnosis and observations by a clinician are ultimately required.

4.6.2 Future Work

Within the SSC, the probands have a much more comprehensive set of measurements available (see Table C.1). Measurements of the *Unaffected Sibling* control group were limited to a subset of the recorded diagnostic questionnaires. An in depth exploration of the wider set of proband data could be highly rewarding. As discussed in Section 4.2.2, prior studies have predominantly concentrated on summary scores or single diagnostic surveys. Given the substantial levels of partial data within the SSC, adopting a network integration approach that harnesses its capability to incorporate partial data could prove highly successful. A promising future direction of study would involve comparing complete clusters to clusters derived from partial data.

In conjunction with such efforts, conducting an in-depth analysis of cluster factors becomes crucial. The preliminary exploration presented here is insufficient, particularly in the case of SSC. Although factors have been identified, their distributions across clusters have not been examined, and the reasons for their increased importance remain unidentified. For clusters to be clinically relevant, a rigorous and interpretable analysis of cluster factors is essential.

The graph-based imputation method showcased in this study demonstrated significant success in enabling accurate prediction of disease subtypes in TCGA and diagnosis of individuals with ASD using tabular features. Graph-based learning has proven highly effective at leveraging network structure in conjunction with node features across a diverse range of applications Zhou et al. (2020). An intriguing avenue for further investigation would be to

explore graph learning techniques using partial complete multi-modal data. While subtype prediction and ASD diagnosis exemplify types of problems that could benefit from graph-based learning (node prediction), other applications such as the addition of entities to the network (edge prediction) present interesting and potentially fruitful future directions.

Chapter 5

Discussion

5.1 Discussion

In this thesis, I assess components of the network construction process by quantifying their influence on community detection performance. To facilitate this evaluation, I develop a data generation framework featuring diverse data distributions, cluster compositions, and inter-modality relationships. My findings are validated on two pertinent biomedical datasets — discerning tumour subtypes in the Cancer Genome Atlas (TCGA) and distinguishing individuals with Autism Spectrum Disorder (ASD) in the Simons Simplex Collection (SSC). The key outcomes of this thesis are outlined below:

Threshold sparsification proves ineffective in generating networks with meaningful community structures. A global approach to sparsification, it fails to embed the local information that form clusters within a network. Notably, it faces challenges in handling clusters with varying density in the feature space. In Chapter 2, I demonstrated a consistent decline in clustering performance on Threshold networks compared to K-nearest neighbour (KNN) networks, across various distributions and clustering problems. In Chapter 3 and Chapter 4, Extreme Mean, a variant of thresholding, consistently exhibited the poorest performance among all integration methods. Thresholding is less scalable than KNN networks, lacking an equivalent to approximate KNN methods. Hyperparameter selection is notably more challenging, requiring normalisation or substantial fine-tuning. Threshold networks typically produce disconnected components, isolated nodes, or necessitate significant increases in density improve the connectivity of the network. Based on these findings, I recommend that KNN networks be preferred for all community detection applications.

Similarity Network Fusion (SNF) B. Wang et al. (2014) does not consistently demonstrate advantages over simpler integration methods such as mean similarity score (Mean S_i) and NEighborhood-based Multi-Omics clustering (NEMO) Rappoport and Shamir (2019). Using our data generation framework, in Chapter 3, I highlighted several data scenarios where SNF clustering performed significantly worse than Mean S_i, notably in data problems where clusters are merged in different modalities. SNF exhibited notable success in scenarios where clusters are split into subclusters across modalities and when incorporating random modalities. Addi-

tionally, as the number of modalities increased, SNF emerged as significantly superior. In my synthetic data problems, SNF consistently outperformed NEMO integration. However, on real-world datasets presented in Chapter 4, NEMO showed significant improvements over SNF networks, particularly on the more challenging TCGA datasets — BRCA and LGG. While the suitability of SNF varied across complete datasets, its performance on partial data was consistently poor. NEMO proved to be more resilient to partial data, producing networks that enable accurate clustering. With improved imputation, SNF can successfully incorporate partial data, as seen in Chapter 4, but it is highly sensitive and frequently experiences a collapse in network structure, failing to embed communities successfully.

Multi-modal integration offers significant benefits. As observed in Chapter 3 and Chapter 4, multi-modal integration consistently outperforms single modality networks for community detection. The most notable advantage of multi-modal network construction lies in its ability to facilitate the *inclusion of partial modalities*. Partially complete data provides significant benefits. Not only does it significantly reduce data wastage but both clustering and prediction problems can benefit form the inclusion of partial data. I demonstrated the ability of multi-modal networks to offer improved imputation techniques for unit non-response partial data. As observed in both TCGA and SSC data in Chapter 4, clustering and prediction of complete data improved with the inclusion of partial data. While the accuracy on partial data can vary, the benefits on complete data are evident. With a more comprehensive coverage of the feature space, clustering and prediction algorithms demonstrate improved performance.

5.2 Future Work

In this thesis, I concentrated on the impact of network construction on community detection performance. Graph representation learning, a rapidly growing field in network analysis, aims to apply the success of deep learning network algorithms to network data. Successful approaches in this area enable effective node, edge, and graph-based predictions, making them highly valuable. An in-depth exploration of the effects of network construction components on graph learning performance would be valuable. While the findings of this work are likely to remain relevant, the flexibility of graph learning might benefit from alternative approaches.

In Chapter 1, I noted that similarity network construction involves two key steps: similarity estimation and edge sparsification[1]. While I have focused on the latter in this thesis, the former remains under-explored. The impact of the scaled exponential kernel on Extreme Mean performance in Chapter 3 highlighted that similarity measures can compensate for

1. With multi-modal data, an additional step is introduced — similarity integration.

drawbacks in a particular method. Could threshold sparsification improve in performance with alternative metrics? Are certain metrics more suitable for specific integration approaches? While application-specific metrics are common, a potential future research direction would be to explore the choice of similarity estimation in typical data scenarios.

As discussed in Chapter 2 and Chapter 3, an expansion of the data generation framework would be highly beneficial. The full range of experiments possible with the current implementation has not yet been fully explored. With improvements in the range of distributions offered and types of cluster mappings provided, more complex and in-depth experiments could be conducted. For example, exploring hyperparameter selection when ground truth data is unavailable or examining the effect of pairwise similarity consistency in more complex scenarios. The current data generation framework, as well as the network construction and clustering algorithms, can be found in the `simnetpy` package: https://github.com/amarnane/simnetpy.

A key avenue of research is a comprehensive subtype analysis of the probands within the SSC dataset. The success of partial data integration opens the possibility for comparing a wider set of individuals and diagnostic measurements with more granular detail. A key restriction in the analysis of developmental disorders such as ASD, is the use of different modules based on age and language ability in key diagnostic measures. Comparisons across age ranges are restricted by an inability to include these questionnaires due to partially complete data. With the evidence presented here, an in-depth exploration of subtypes within SSC, comparing both complete and partial data, and focusing on factors driving clusters could lead to clinical implications.

5.3 Conclusion

In summary, this thesis illuminates the critical role of network construction. Through formal demonstrations of its impact on node class prediction and clustering, I have shed light on an often-overlooked aspect of current biomedical analysis pipelines. By providing comprehensive guidance on network construction, this work offers valuable insights applicable across various fields of study. I have identified key considerations for both single and multi-modal use cases and illustrated the application of these principles on real world datasets. These findings not only enhance our understanding of network analysis in biomedicine but also provide a practical foundation for future research.

Appendix A

Sparsification

A.1 Evaluation using AMI

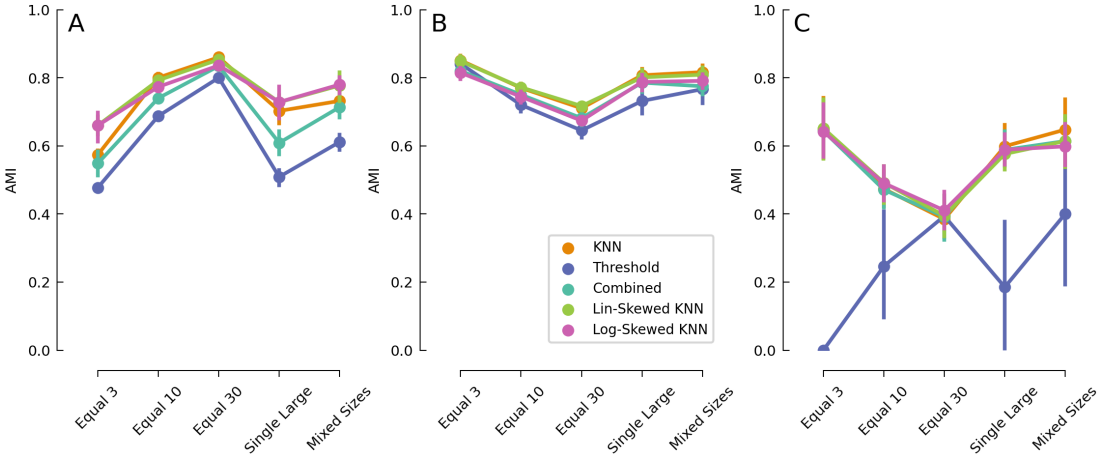

Figure A.1: AMI Performance of Sparsification Methods Across Different Clustering Algorithms on Mixed Gaussian Data. The AMI performance of the sparsification methods using **A** SBM, **B** Leiden and **C** Spectral for mixed Gaussian data is shown. 10 instances of data are evaluated using the optimal parameter identified for each clustering algorithm on each sparsification method. The differences in performance form cluster problem to cluster problem is not as significant. AMI does not punish incorrect prediction of the number of clusters as severely and gap between SBM and Leiden clustering is reduced compared to ARI.

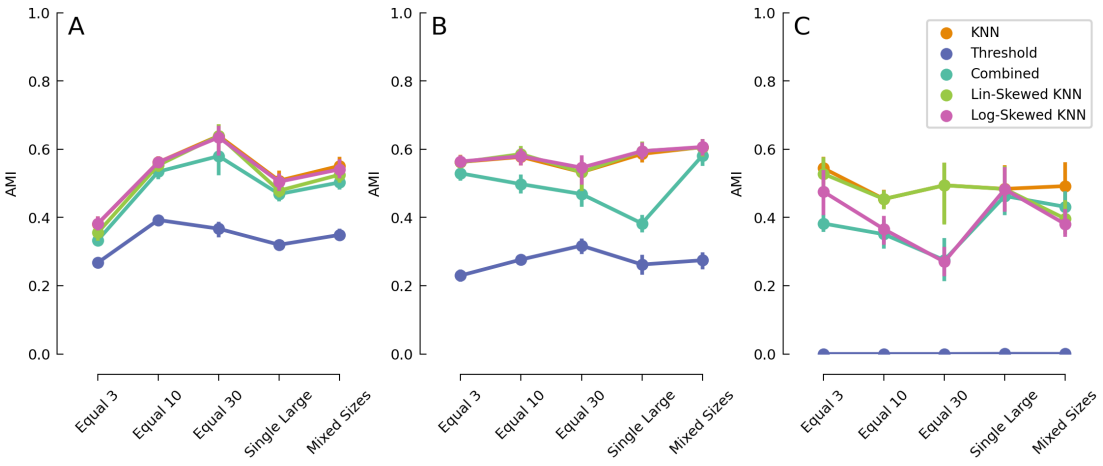

Figure A.2: AMI Performance of Sparsification Methods Across Different Clustering Algorithms on Mixed Student's-t Data. The AMI performance of the sparsification methods using **A** SBM, **B** Leiden and **C** Spectral for mixed Student's-t data is shown. 10 instances of data are evaluated using the optimal parameter identified for each clustering algorithm on each sparsification method. The differences between Linear-Skewed KNN and KNN seen in ARI evaluation (Figure 2.14) disappear. Threshold network performance, while still the worst performing, is not as poor when evaluated with AMI.

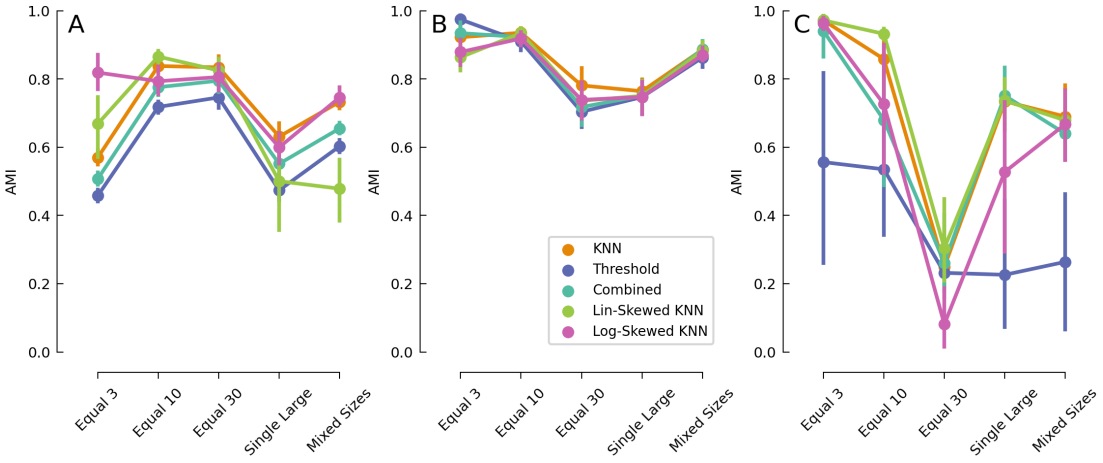

Figure A.3: AMI Performance of Sparsification Methods Across Different Clustering Algorithms on Categorical Data. The AMI performance of the sparsification methods using **A** SBM, **B** Leiden and **C** Spectral for mixed Student's-t data is shown. 10 instances of data are evaluated using the optimal parameter identified for each clustering algorithm on each sparsification method.

A.2 Additional Figures

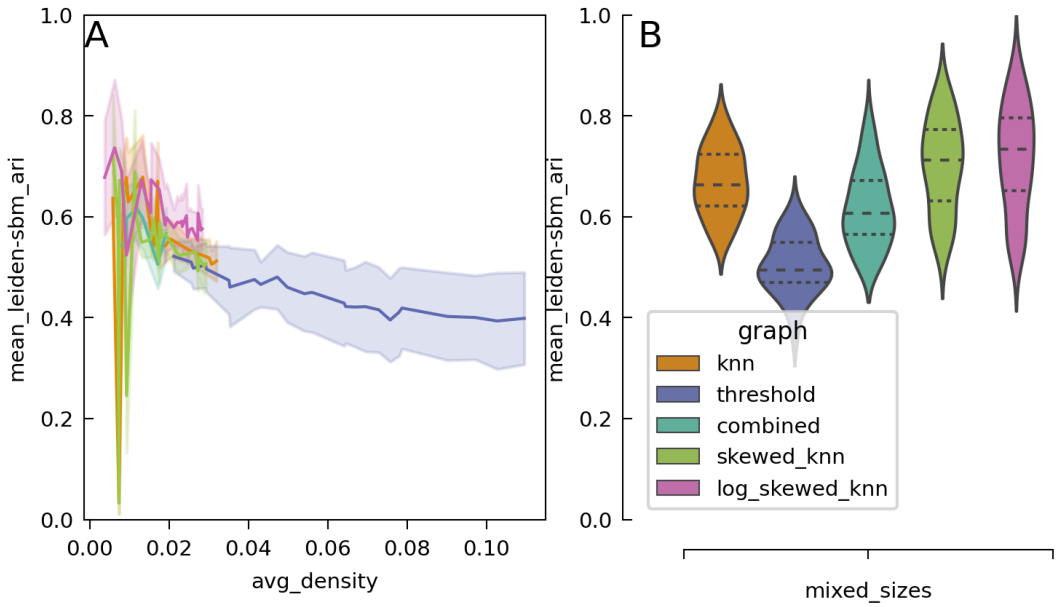

Figure A.4: Hyperparameter Search and Performance Evaluation of Sparsification Methods using mean SBM and Leiden ARI. The mean SBM and Leiden clustering ARI of the five sparsification methods across all five cluster settings of mixed Gaussian data is shown using euclidean distance as a metric. Panel **A** shows the change in performance for different hyperparameter choices. To fairly compare the different parameters, we plot ARI vs graph density. Panel **B** shows the distribution of mean SBM and Leiden ARI across 10 instances. Hyperparameters are selected which result in the highest mean Leiden and SBM ARI score on each method.

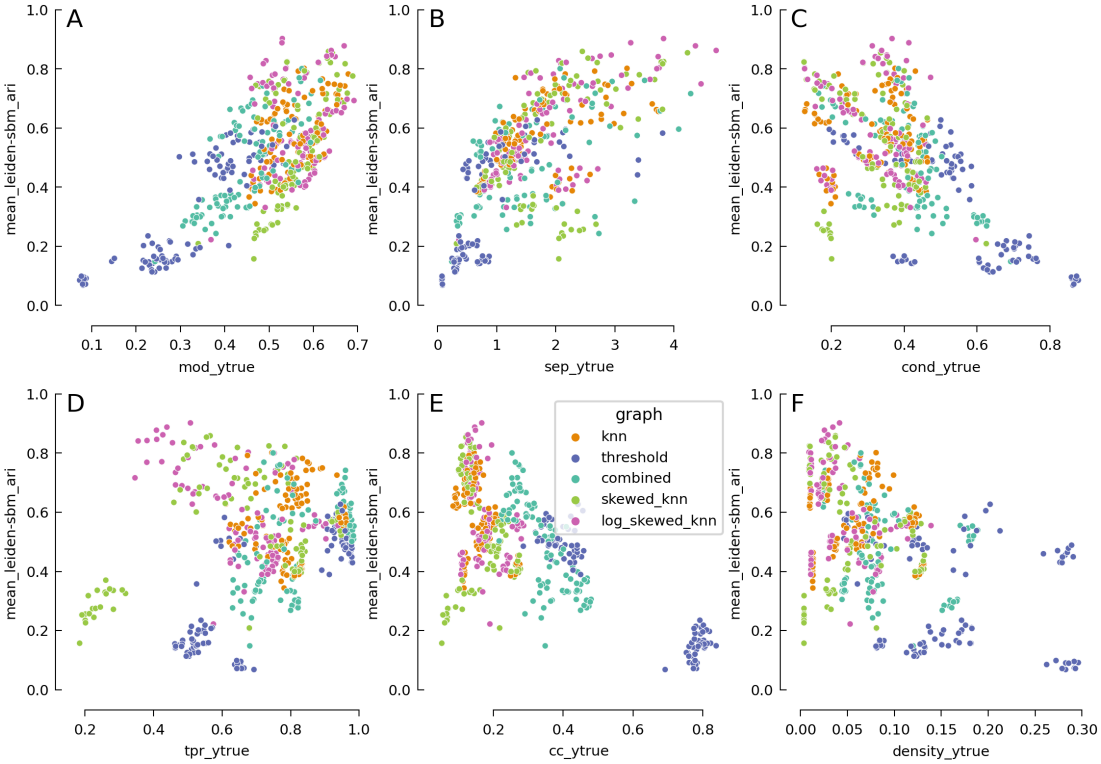

Figure A.5: Relationship between Mean ARI and Clustering Quality Measures of Sparsification methods on Mixed Gaussian and Student's-t Data. Average ARI for Leiden and SBM methods for ground truth cluster quality score for Gaussian and Student's-t distributed data across all clustering problems vs. **A** Modularity, **B** Separability, **C** Conductance, **D** TPR, **E** Clustering Coefficient and **F** average density. We can see quality of the true clusters is positively correlated for **A** & **B** and negatively correlated for **C**, **E**, and **F**.

Modularity	Separability	Conductance	TPR	Clustering Coefficient	Density
0.857	0.715	-0.844	0.09	-0.816	-0.760

Table A.1: Correlation Coefficient between ARI and Clustering Quality Measures of Sparsification methods on Mixed Gaussian and Student's-t Data. Correlation values for Panels A-F in Figure A.5 between ground truth cluster y quality score and mean ARI of Leiden and SBM clustering for both Gaussian and Student-t distributed data.

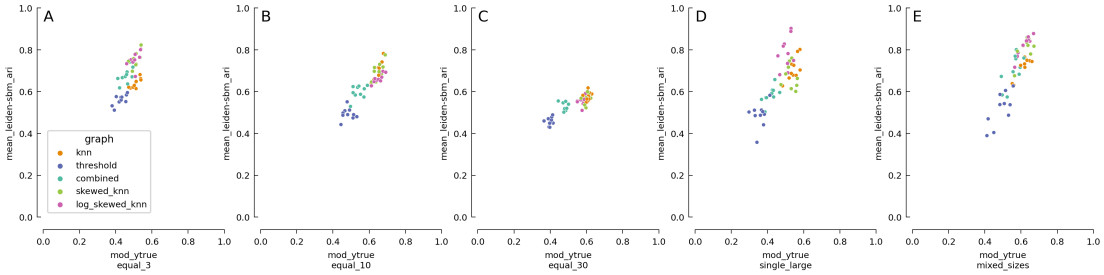

Figure A.6: Relationship between Ground Truth Modularity and Mean ARI on Mixed Gaussian Data. Ground truth cluster y modularity scores for mixed Gaussian data on the five clustering problems. Ground truth modularity is strongly correlated with mean ARI of Leiden and SBM clustering methods. We can see threshold based methods (Threshold & Combined) consistently produce networks with lower modularity compared to the KNN-based methods. Log-Skewed KNN creates networks with higher modularity than KNN in settings with large clusters. Surprisingly, no method produces clusters with a modularity above 0.7 across all problems.

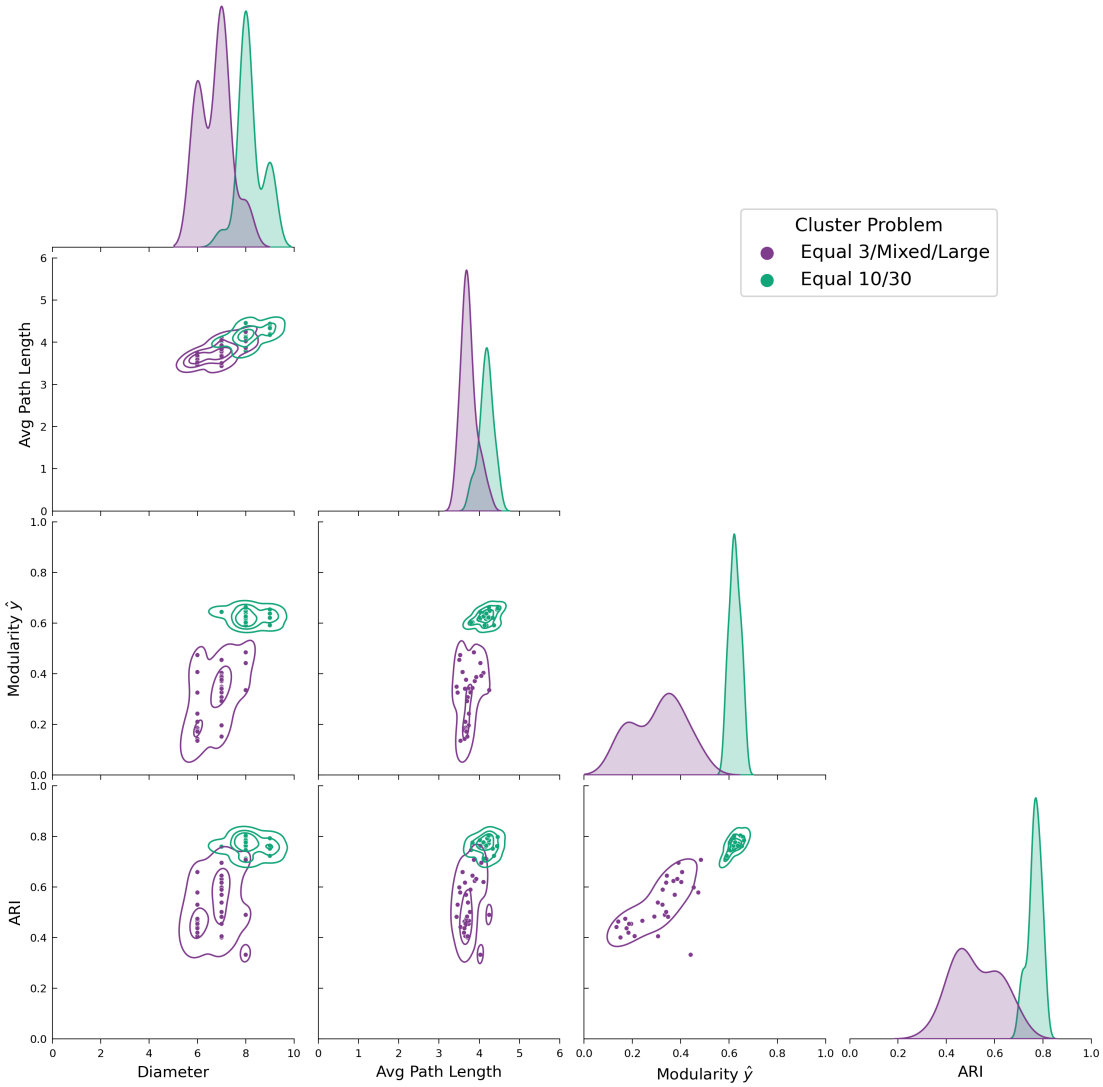

Figure A.7: Pairwise Distributions of Network Metrics and SBM ARI for KNN networks by Cluster Problem on Gaussian Data. Difference in structure of KNN networks between problems with a low number of large clusters and problems with a high number of smaller clusters. Large clusters have a smaller diameter, lower average path length and significantly lower predicted cluster modularity. This lower predicted cluster modularity corresponds strongly to lower ARI performance.

Appendix B

Multi-modal Integration

B.1 Additional Figures

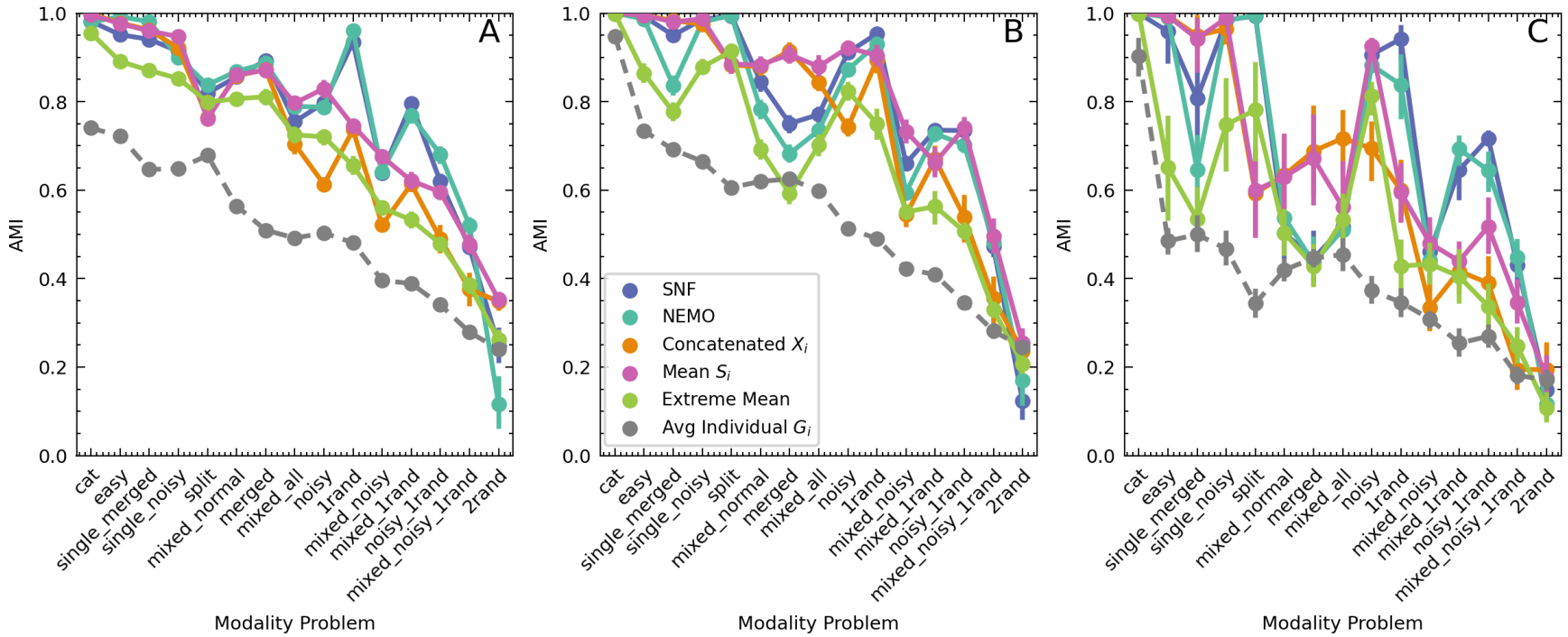

Figure B.1: AMI Performance Comparison of Similarity Integration Methods across Multiple Modalities using Correlation Metric. AMI performance of A) SBM B) Leiden and C) Spectral clustering algorithm on 20 instances of 15 different modality problems using **Correlation** distance are shown for the five similarity integration methods. The gap between Mean S_i and SNF and NEMO on SBM clustering is significantly reduced.

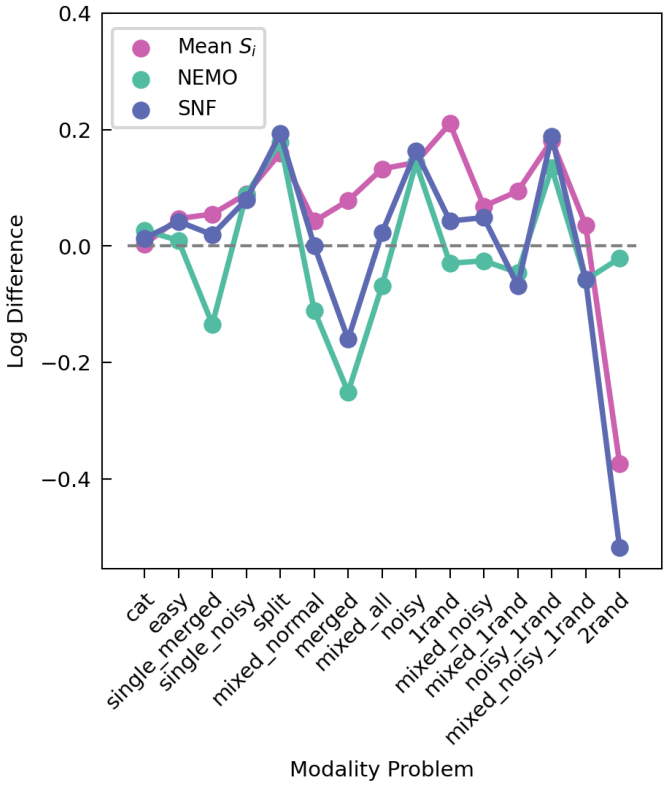

Figure B.2: Comparison of Leiden and SBM clustering by Integration Method across Modality Problems. Log AMI difference between Leiden and SBM clustering on SNF, Mean S_i and NEMO networks on 20 instances of 15 modality problems using both euclidean and correlation metrics. Leiden is always preferable on Mean S_i networks. SBM clustering outperforms Leiden on SNF and NEMO on modalities with Merged clusters. SBM clustering is a preferential choice on NEMO networks on multiple types of modality problems.

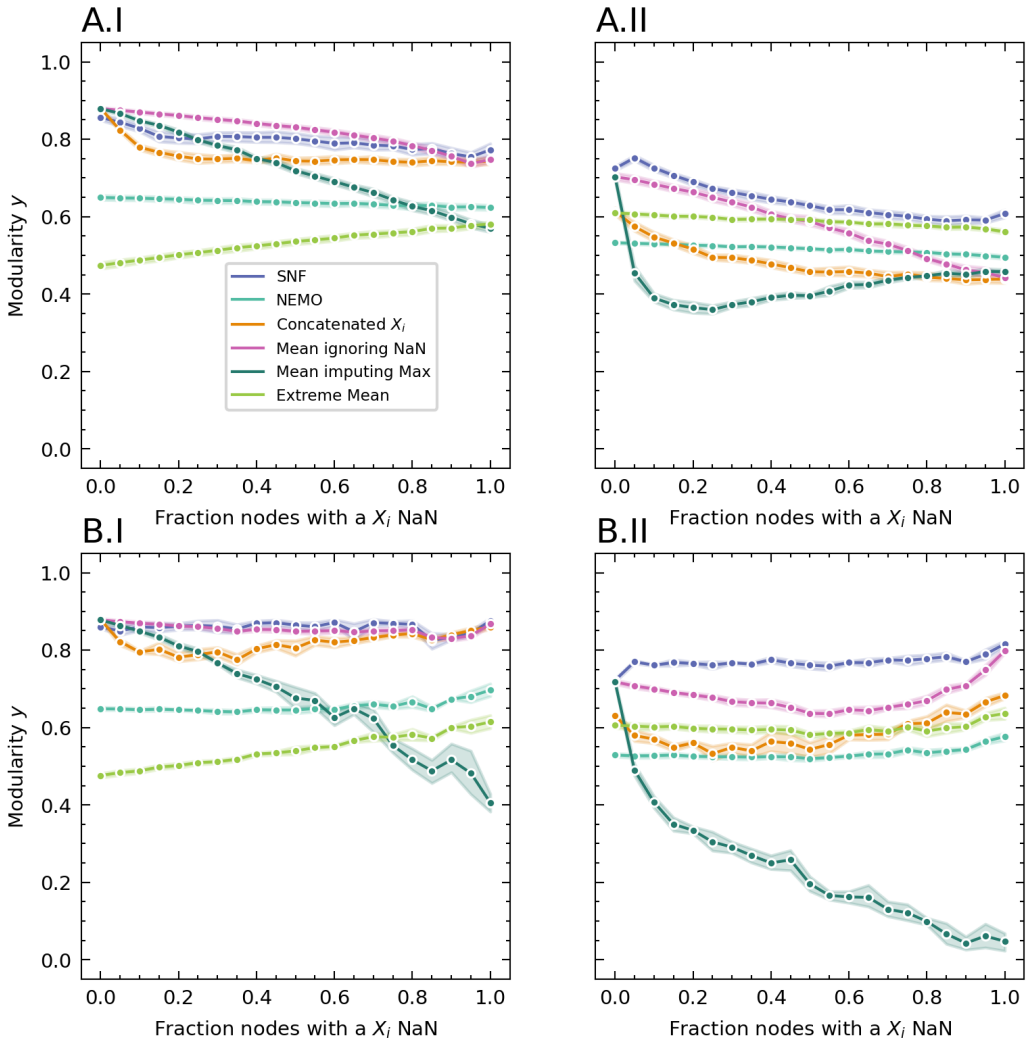

Figure B.3: Change in Ground Truth Modularity with Increasing Partial — *Easy* and *Noisy* Modality Problems. Change in Modularity y for five instances of A) *Random* and B) *Cluster Based* partial data on I) *Easy* and II) *Noisy* modality problems. Mean imputing max experiences a sharp decline in modularity in all cases. NEMO and Extreme Mean are relatively unaffected by partial data both at random and cluster based. The modularity of most methods is less affected by cluster based partial data - the modularity with all nodes having cluster based NaN is higher than no nodes being partial on *Noisy* data.

Appendix C

Applications

C.1 SSC Data Measurements

Table C.1: List of Phenotypic Measures Collected Within the Simon's Simplex Collection (SSC).

ID	Name	Measures	Format
Diagnosis			
ADI-R	Autism Diagnostic Interview-Revised	Parent report of behaviours related to autism phenotype	Direct interview
ADOS	Autism Diagnostic Observation Schedule Modules 1,2,3 and 4.	Observational measure of autism phenotype	Direct examiner observation
ABCL 18-59	Adult Behaviour Checklist for Ages 19 to 59	Problem behaviour	Questionnaire
CBCL 2-5	Child Behaviour Checklist for ages 2 to 5 years	Problem behaviour	Questionnaire
CBCL 6-18	Child Behaviour Checklist for ages 6 to 18 years	Problem behaviour	Questionnaire
SCQ Parent	Social Communication Questionnaire — Parent report	Screen of ASD markers	Questionnaire
SCQ Teacher	Social Communication Questionnaire — Teacher report	Screen of ASD markers	Questionnaire
SRS Adult	Social Responsiveness Scale — Adult Research Version	Autistic traits on a continuous scale	Questionnaire
SRS Parent	Social Responsiveness Scale — Parent report	Autistic traits on a continuous scale	Questionnaire

Continued on next page

Table C.1 – continued from previous page

ID	Name	Measures	Format
SRS Teacher	Social Responsiveness Scale — Teacher report	Autistic traits on a continuous scale	Questionnaire
TRF 6-18	Teacher Report Form	Problem behaviour and school functioning	Questionnaire
CTRF 2-5	Caregiver-Teacher Report Form	Problem behaviour	Questionnaire
Cognitive			
DAS-II	Differential Ability Scales, Second Edition	Cognitive ability	Direct assessment
Mullen	Mullen Scales of Early Learning, AGS Edition	Cognitive ability	Direct assessment
WASI	Wechsler Abbreviated Scale of Intelligence	Cognitive ability	Direct assessment
WISC-IV	Wechsler Intelligence Scale for Children, Fourth Edition	Cognitive ability	Direct assessment
Vineland II	Vineland Adaptive Behaviour Scale-II	Adaptive behaviour	Interview
ABC	Aberrant Behaviour Checklist	Aberrant behaviours	Questionnaire
DCDQ	Developmental Coordination Disorder Questionnaire	Motor delays	Questionnaire
CTOPP-NR	Comprehensive Test of Phonological Processing Non-word Repetition	Speech and memory of sounds and non-words	Direct assessment
PPVT-4	Peabody Picture Vocabulary Test, Fourth Edition	Receptive single-word vocabulary	Direct assessment
Purdue	Purdue Peg Board	Fine-motor dexterity	Direct assessment
Raven's	Raven's Standard Progressive Matrices	Nonverbal problem solving	Direct assessment
RBS-R	Repetitive Behaviour Scale-Revised	Repetitive behaviours	Questionnaire

C.2 Additional Figures

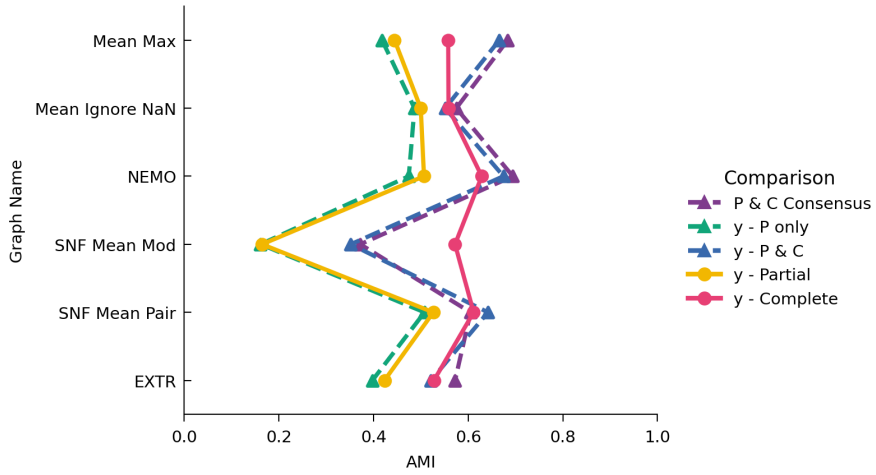

Figure C.1: AMI Clustering Performance for Partial and Complete SSC Data by Nodetype. Breakdown of partial and complete data clustering performance by nodetype through the mean AMI scores generated by SBM, Leiden, and Spectral clustering algorithms on multimodal integration networks on SSC. We show the AMI scores for partial data (*y - Partial*), complete data (*y - Complete*), a breakdown of *y-Partial* based on nodetype, nodes exclusive to partial data (*y - P only*) and nodes present in both partial and complete datasets (*y - P & C*), and the AMI agreement of complete nodes between their partial and complete clusters (*P & C consensus*). Mean Max, NEMO and SNF Mean Pair improve the clustering of the partial-complete nodes when partial data is included.

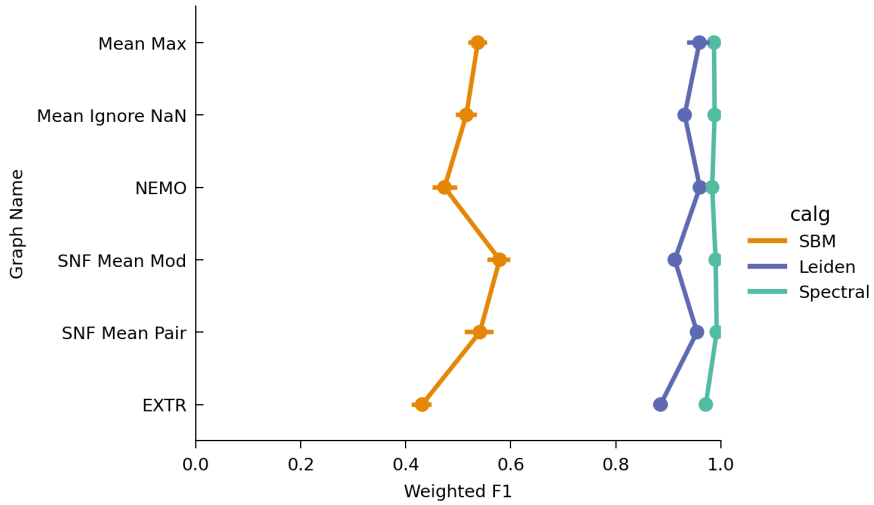

Figure C.2: Predictability of Clusters Detected by SBM, Leiden and Spectral Algorithms on SSC Data. Weighted F_1 scores of the prediction of cluster labels generated by SBM, Leiden, and Spectral clustering algorithms. These scores are derived using 5-fold cross-validated random forest models trained on both partial and complete SSC datasets. Leiden clusters are highly predictable and have high weighted F_1-score on both partial and complete data. Leiden clusters on SSC are highly predictable despite the low AMI. These sub-clusters identified by Leiden (Figure 4.12) may have clinical importance and might worthy of further exploration.

Bibliography

Achenbach, T. M., & Verhulst, F. (2010). *Achenbach system of empirically based assessment (ASEBA)*. Burlington, Vermont.

Acosta, J. N., Falcone, G. J., Rajpurkar, P., & Topol, E. J. (2022, September). Multimodal biomedical AI. *Nature Medicine*, *28*(9), 1773–1784. Retrieved from https://doi.org/10.1038/s41591-022-01981-2 doi: 10.1038/s41591-022-01981-2

Afzalan, M., & Jazizadeh, F. (2019, June). An automated spectral clustering for multi-scale data. *Neurocomputing*, *347*, 94–108. Retrieved from https://www.sciencedirect.com/science/article/pii/S0925231219303108 doi: 10.1016/j.neucom.2019.03.008

Agelink van Rentergem, J. A., Deserno, M. K., & Geurts, H. M. (2021, July). Validation strategies for subtypes in psychiatry: A systematic review of research on autism spectrum disorder. *Clinical Psychology Review*, *87*, 102033. Retrieved from https://www.sciencedirect.com/science/article/pii/S0272735821000763 doi: 10.1016/j.cpr.2021.102033

Ahern, D. J., Ai, Z., Ainsworth, M., Allan, C., Allcock, A., Angus, B., ... Zurke, Y.-X. (2022, March). A blood atlas of COVID-19 defines hallmarks of disease severity and specificity. *Cell*, *185*(5), 916–938.e58. Retrieved from https://www.sciencedirect.com/science/article/pii/S0092867422000708 doi: 10.1016/j.cell.2022.01.012

Allen, J. D., Xie, Y., Chen, M., Girard, L., & Xiao, G. (2012). Comparing statistical methods for constructing large scale gene networks. *PloS one*, *7*(1), e29348. (Publisher: Public Library of Science San Francisco, USA)

Arslanturk, S., Siadat, M.-R., Ogunyemi, T., Killinger, K., & Diokno, A. (2016, March). Analysis of incomplete and inconsistent clinical survey data. *Knowledge and Information Systems*, *46*(3), 731–750. Retrieved 2024-01-13, from https://doi.org/10.1007/s10115-015-0850-7 doi: 10.1007/s10115-015-0850-7

Baltrušaitis, T., Ahuja, C., & Morency, L.-P. (2018). Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, *41*(2), 423–443. (Publisher: IEEE)

Barabási, A.-L., & Márton, P. (2016). *Network Science*. Cambridge University Press. Retrieved from https://books.google.ie/books?id=iLtGDQAAQBAJ

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008, October). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, *2008*(10), P10008. Retrieved 2019-03-02, from http://arxiv.org/abs/0803.0476 (arXiv: 0803.0476) doi: 10.1088/1742-5468/2008/10/P10008

Brannon, A. R., Reddy, A., Seiler, M., Arreola, A., Moore, D. T., Pruthi, R. S., ... Rathmell, W. K. (2010, February). Molecular Stratification of Clear Cell Renal Cell Carcinoma by Consensus Clustering Reveals Distinct Subtypes and Survival Patterns. *Genes & Cancer*, *1*(2), 152–163. Retrieved 2023-12-20, from https://journals.sagepub.com/doi/abs/10.1177/1947601909359929 (Publisher: SAGE Publications) doi: 10.1177/1947601909359929

Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., ... Wiener, J. (2000). Graph structure in the web. *Computer networks*, *33*(1-6), 309–320. (Publisher: Elsevier)

Callaway, D. S., Newman, M. E., Strogatz, S. H., & Watts, D. J. (2000). Network robustness and fragility: Percolation on random graphs. *Physical review letters*, *85*(25), 5468. (Publisher: APS)

Cavalli, F. M., Remke, M., Rampasek, L., Peacock, J., Shih, D. J., Luu, B., ... Taylor, M. D. (2017, June). Intertumoral Heterogeneity within Medulloblastoma Subgroups. *Cancer Cell*, *31*(6), 737–754.e6. Retrieved from https://www.sciencedirect.com/science/article/pii/S1535610817302015 doi: 10.1016/j.ccell.2017.05.005

Chen, J., Fang, H.-r., & Saad, Y. (2009). Fast Approximate kNN Graph Construction for High Dimensional Data via Recursive Lanczos Bisection. *Journal of Machine Learning Research*, *10*(9).

Chen, M., Kuzmin, K., & Szymanski, B. K. (2014, March). Community Detection via Maximization of Modularity and Its Variants. *IEEE Transactions on Computational Social Systems*, *1*(1), 46–65. Retrieved 2024-01-13, from https://ieeexplore.ieee.org/abstract/document/6785984 (Conference Name: IEEE Transactions on Computational Social Systems) doi: 10.1109/TCSS.2014.2307458

Chen, Y., Zhang, X., Zhang, G.-q., & Xu, R. (2015, February). Comparative analysis of a novel disease phenotype network based on clinical manifestations. *Journal of Biomedical Informatics*, *53*, 113–120. Retrieved from https://www.sciencedirect.com/science/article/pii/S1532046414002172 doi: 10.1016/j.jbi.2014.09.007

Clauset, A., Newman, M. E. J., & Moore, C. (2004, December). Finding community structure in very large networks. *Physical Review E*, *70*(6), 066111. Retrieved from https://link.aps.org/doi/10.1103/PhysRevE.70.066111 (Publisher: American Physical Society) doi: 10.1103/PhysRevE.70.066111

Constantino, J. N., & Gruber, C. P. (2012). Social responsiveness scale: SRS-2. (Publisher: Western psychological services Torrance, CA)

Csardi, G., & Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal*, *Complex Systems*, 1695. Retrieved from https://igraph.org

Dai, L., Zhu, H., & Liu, D. (2020). Patient similarity: methods and applications. *arXiv preprint arXiv:2012.01976*.

Davies, D. L., & Bouldin, D. W. (1979). A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*(2), 224–227. (Publisher: IEEE)

Deng, X., Das, S., Kaur, H., Wilson, E., Camphausen, K., & Shankavaram, U. (2023, July). Glioma-BioDP: database for visualization of molecular profiles to improve prognosis of brain cancer. *BMC Medical Genomics*, *16*(1), 168. Retrieved from https://doi.org/10.1186/s12920-023-01593-w doi: 10.1186/s12920-023-01593-w

Dozmorov, M. G. (2018, June). Disease classification: from phenotypic similarity to integrative genomics and beyond. *Briefings in Bioinformatics*. Retrieved 2018-11-30, from https://academic.oup.com/bib/advance-article/doi/10.1093/bib/bby049/5043113 doi: 10.1093/bib/bby049

Eaves, L. C., Wingert, H. D., Ho, H. H., & Mickelson, E. C. (2006). Screening for autism spectrum disorders with the social communication questionnaire. *Journal of Developmental & Behavioral Pediatrics*, *27*(2), S95–S103. (Publisher: LWW)

Elliott, C. D., Salerno, J. D., Dumont, R., & Willis, J. O. (2007). Differential ability scalesSecond edition. *San Antonio, TX*.

Erdös, P., & Rényi, A. (1959). On random graphs I. *Publicationes Mathematicae Debrecen*(6), 290–297. doi: doi:10.5486/PMD.1959.6.3-4.12

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In (Vol. 96, pp. 226–231). (Issue: 34)

Falkmer, T., Anderson, K., Falkmer, M., & Horlin, C. (2013, June). Diagnostic procedures in autism spectrum disorders: a systematic literature review. *European Child & Adolescent Psychiatry*, *22*(6), 329–340. Retrieved from https://doi.org/10.1007/s00787-013-0375-0 doi: 10.1007/s00787-013-0375-0

Feldner-Busztin, D., Firbas Nisantzis, P., Edmunds, S. J., Boza, G., Racimo, F., Gopalakrishnan, S., ... de Polavieja, G. G. (2023, January). Dealing with dimensionality: the application of machine learning to multi-omics data. *Bioinformatics*, *39*(2), btad021. Retrieved 2024-01-13, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9907220/ doi: 10.1093/bioinformatics/btad021

Feliciano, P., Daniels, A. M., Green Snyder, L., Beaumont, A., Camba, A., Esler, A., ... Chung, W. K. (2018, February). SPARK: A US Cohort of 50,000 Families to Accelerate Autism Research. *Neuron*, *97*(3), 488–493. Retrieved 2022-04-22, from https://doi.org/10.1016/j.neuron.2018.01.015 (Publisher: Elsevier) doi: 10.1016/j.neuron.2018.01.015

Fischbach, G. D., & Lord, C. (2010, October). The Simons Simplex Collection: A Resource for Identification of Autism Genetic Risk Factors. *Neuron*, *68*(2), 192–195. Retrieved 2019-10-04, from http://www.sciencedirect.com/science/article/pii/S0896627310008305 doi: 10.1016/j.neuron.2010.10.006

Flores, J. E., Claborne, D. M., Weller, Z. D., Webb-Robertson, B.-J. M., Waters, K. M., & Bramer, L. M. (2023). Missing data in multi-omics integration: Recent advances through artificial intelligence. *Frontiers in Artificial Intelligence*, *6*. Retrieved 2024-01-12, from https://www.frontiersin.org/articles/10.3389/frai.2023.1098308

Fortunato, S., & Barthélemy, M. (2007, January). Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, *104*(1), 36–41. Retrieved 2022-12-07, from https://doi.org/10.1073/pnas.0605965104 (Publisher: Proceedings of the National Academy of Sciences) doi: 10.1073/pnas.0605965104

Fortunato, S., & Hric, D. (2016, November). Community detection in networks: A user guide. *Community detection in networks: A user guide*, *659*, 1–44. Retrieved from https://www.sciencedirect.com/science/article/pii/S0370157316302964 doi: 10.1016/j.physrep.2016.09.002

Fortunato, S., & Newman, M. E. J. (2022, August). 20 years of network community detection. *Nature Physics*, *18*(8), 848–850. Retrieved from https://doi.org/10.1038/s41567-022-01716-7 doi: 10.1038/s41567-022-01716-7

Fowlkes, E. B., & Mallows, C. L. (1983). A method for comparing two hierarchical clusterings. *Journal of the American statistical association*, *78*(383), 553–569. (Publisher: Taylor & Francis)

Gene Ontology Consortium. (2004, January). The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Research*, *32*(90001), 258D–261. Retrieved 2019-07-31, from https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gkh036 doi: 10.1093/nar/gkh036

Girvan, M., & Newman, M. E. J. (2002, June). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, *99*(12), 7821–7826. Retrieved 2024-08-17, from https://doi.org/10.1073/pnas.122653799 (Publisher: Proceedings of the National Academy of Sciences) doi: 10.1073/pnas.122653799

Good, B. H., de Montjoye, Y.-A., & Clauset, A. (2010, April). Performance of modularity maximization in practical contexts. *Physical Review E*, *81*(4), 046106. Retrieved from https://link.aps.org/doi/10.1103/PhysRevE.81.046106 (Publisher: American Physical Society) doi: 10.1103/PhysRevE.81.046106

Gotham, K., Risi, S., Pickles, A., & Lord, C. (2007). The Autism Diagnostic Observation Schedule: revised algorithms for improved diagnostic validity. *Journal of autism and developmental disorders*, *37*, 613–627. (Publisher: Springer)

Graham, J. W. (2009, January). Missing Data Analysis: Making It Work in the Real World. *Annual Review of Psychology*, *60*(1), 549–576. Retrieved 2024-01-12, from https://doi.org/10.1146/annurev.psych.58.110405.085530 (Publisher: Annual Reviews) doi: 10.1146/annurev.psych.58.110405.085530

Greaves-Lord, K., Eussen, M. L. J. M., Verhulst, F. C., Minderaa, R. B., Mandy, W., Hudziak, J. J., ... Hartman, C. A. (2013, August). Empirically Based Phenotypic Profiles of Children with Pervasive Developmental Disorders: Interpretation in the Light of the DSM-5. *Journal of Autism and Developmental Disorders*, *43*(8), 1784–1797. Retrieved from https://doi.org/10.1007/s10803-012-1724-4 doi: 10.1007/s10803-012-1724-4

Grossman, R. L., Heath, A. P., Ferretti, V., Varmus, H. E., Lowy, D. R., Kibbe, W. A., & Staudt, L. M. (2016). Toward a shared vision for cancer genomic data. *New England Journal of Medicine*, *375*(12), 1109–1112. (Publisher: Mass Medical Soc)

Hall, M. K., Kea, B., & Wang, R. (2019). Recognising bias in studies of diagnostic tests part 1: patient selection. *Emergency Medicine Journal*. (Publisher: BMJ Publishing Group Ltd and the British Association for Accident ...)

Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive Representation Learning on Large Graphs. In I. Guyon et al. (Eds.), *Advances in Neural Information Processing Systems* (Vol. 30). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7ebea9-Paper.pdf

Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*.

Hao, Y., Hao, S., Andersen-Nissen, E., Mauck, W. M., Zheng, S., Butler, A., ... Satija, R. (2021, June). Integrated analysis of multimodal single-cell data. *Cell*, *184*(13), 3573–3587.e29. Retrieved from https://www.sciencedirect.com/science/article/pii/S0092867421005833 doi: 10.1016/j.cell.2021.04.048

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770–778). doi: 10.1109/CVPR.2016.90

Holland, P. W., Laskey, K. B., & Leinhardt, S. (1983, June). Stochastic blockmodels: First steps. *Social Networks*, *5*(2), 109–137. Retrieved from https://www.sciencedirect.com/science/article/pii/0378873383900217 doi: 10.1016/0378-8733(83)90021-7

Huang, L., Luo, H., Li, S., Wu, F.-X., & Wang, J. (2021, July). Drug–drug similarity measure and its applications. *Briefings in Bioinformatics*, *22*(4), bbaa265. Retrieved 2023-05-07, from https://doi.org/10.1093/bib/bbaa265 doi: 10.1093/bib/bbaa265

Hubert, L., & Arabie, P. (1985, December). Comparing partitions. *Journal of Classification*, *2*(1), 193–218. Retrieved from https://doi.org/10.1007/BF01908075 doi: 10.1007/BF01908075

J. Wen, Z. Zhang, L. Fei, B. Zhang, Y. Xu, Z. Zhang, & J. Li. (2022). A Survey on Incomplete Multiview Clustering. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 1–14. doi: 10.1109/TSMC.2022.3192635

Jeub, L. G., Sporns, O., & Fortunato, S. (2018). Multiresolution consensus clustering in networks. *Scientific reports*, *8*(1), 3259. (Publisher: Nature Publishing Group UK London)

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, *349*(6245), 255–260. (Publisher: American Association for the Advancement of Science)

Kamiński, B., Prałat, P., & Théberge, F. (2023, May). Artificial benchmark for community detection with outliers (ABCD+o). *Applied Network Science*, *8*(1), 25. Retrieved from https://doi.org/10.1007/s41109-023-00552-9 doi: 10.1007/s41109-023-00552-9

Karrer, B., & Newman, M. E. J. (2011, January). Stochastic blockmodels and community structure in networks. *Physical Review E*, *83*(1), 016107. Retrieved from https://link.aps.org/doi/10.1103/PhysRevE.83.016107 (Publisher: American Physical Society) doi: 10.1103/PhysRevE.83.016107

Keeling, M. J., & Eames, K. T. (2005). Networks and epidemic models. *Journal of the royal society interface*, *2*(4), 295–307. (Publisher: The Royal Society London)

Khan, B. S., & Niazi, M. A. (2017). Network community detection: A review and visual survey. *arXiv preprint arXiv:1708.00977*.

Kim, T., Chen, I. R., Lin, Y., Wang, A. Y.-Y., Yang, J. Y. H., & Yang, P. (2019, November). Impact of similarity metrics on single-cell RNA-seq data clustering. *Briefings in Bioinformatics*, *20*(6), 2316–2326. Retrieved 2024-01-13, from https://doi.org/10.1093/bib/bby076 doi: 10.1093/bib/bby076

Kiselev, V. Y., Andrews, T. S., & Hemberg, M. (2019, May). Challenges in unsupervised clustering of single-cell RNA-seq data. *Nature Reviews Genetics*, *20*(5), 273–282. Retrieved from https://doi.org/10.1038/s41576-018-0088-9 doi: 10.1038/s41576-018-0088-9

Kiselev, V. Y., Kirschner, K., Schaub, M. T., Andrews, T., Yiu, A., Chandra, T., ... Green, A. R. (2017). SC3: consensus clustering of single-cell RNA-seq data. *Nature methods*, *14*(5), 483–486. (Publisher: Nature Publishing Group US New York)

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*.

Kumpula, J. M., Saramäki, J., Kaski, K., & Kertész, J. (2007). Limited resolution in complex network community detection with Potts model approach. *The European Physical Journal B*, *56*, 41–45. (Publisher: Springer)

Köhler, S., Doelken, S. C., Mungall, C. J., Bauer, S., Firth, H. V., Bailleul-Forestier, I., ... Robinson, P. N. (2014, January). The Human Phenotype Ontology project: linking molecular biology and disease through phenotype data. *Nucleic Acids Research*, *42*(D1), D966–D974. Retrieved 2019-03-04, from https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gkt1026 doi: 10.1093/nar/gkt1026

Lai, M.-C., Lombardo, M. V., & Baron-Cohen, S. (2014, March). Autism. *The Lancet*, *383*(9920), 896–910. Retrieved 2019-03-03, from https://linkinghub.elsevier.com/retrieve/pii/S0140673613615391 doi: 10.1016/S0140-6736(13)61539-1

Lancichinetti, A., & Fortunato, S. (2009, November). Community detection algorithms: A comparative analysis. *Physical Review E*, *80*(5), 056117. Retrieved from https://link.aps.org/doi/10.1103/PhysRevE.80.056117 (Publisher: American Physical Society) doi: 10.1103/PhysRevE.80.056117

Lancichinetti, A., & Fortunato, S. (2012, March). Consensus clustering in complex networks. *Scientific Reports*, *2*(1), 336. Retrieved from https://doi.org/10.1038/srep00336 doi: 10.1038/srep00336

Lancichinetti, A., Fortunato, S., & Radicchi, F. (2008, October). Benchmark graphs for testing community detection algorithms. *Physical Review E*, *78*(4), 046110. Retrieved from https://link.aps.org/doi/10.1103/PhysRevE.78.046110 (Publisher: American Physical Society) doi: 10.1103/PhysRevE.78.046110

Lane, N. D., Xu, Y., Lu, H., Hu, S., Choudhury, T., Campbell, A. T., & Zhao, F. (2014, February). Community Similarity Networks. *Personal and Ubiquitous Computing*, *18*(2), 355–368. Retrieved from https://doi.org/10.1007/s00779-013-0655-1 doi: 10.1007/s00779-013-0655-1

Langfelder, P., & Horvath, S. (2008, December). WGCNA: an R package for weighted correlation network analysis. *BMC Bioinformatics*, *9*(1), 559. Retrieved from https://doi.org/10.1186/1471-2105-9-559 doi: 10.1186/1471-2105-9-559

Li, L., Cheng, W.-Y., Glicksberg, B. S., Gottesman, O., Tamler, R., Chen, R., ... Dudley, J. T. (2015, October). Identification of type 2 diabetes subgroups through topological analysis of patient similarity. *Science Translational Medicine*, *7*(311), 311ra174–311ra174. Retrieved 2024-01-20, from https://doi.org/10.1126/scitranslmed.aaa9364 (Publisher: American Association for the Advancement of Science) doi: 10.1126/scitranslmed.aaa9364

Li, M. M., Huang, K., & Zitnik, M. (2022, December). Graph representation learning in biomedicine and healthcare. *Nature Biomedical Engineering*, *6*(12), 1353–1369. Retrieved from https://doi.org/10.1038/s41551-022-00942-x doi: 10.1038/s41551-022-00942-x

Li, S.-Y., Jiang, Y., & Zhou, Z.-H. (2014). Partial multi-view clustering. In (Vol. 28). (Issue: 1)

Lord, C., Charman, T., Havdahl, A., Carbone, P., Anagnostou, E., Boyd, B., ... McCauley, J. B. (2022, January). The Lancet Commission on the future of care and clinical research in autism. *The Lancet*, *399*(10321), 271–334. Retrieved 2022-08-03, from https://doi.org/10.1016/S0140-6736(21)01541-5 (Publisher: Elsevier) doi: 10.1016/S0140-6736(21)01541-5

Lord, C., Elsabbagh, M., Baird, G., & Veenstra-Vanderweele, J. (2018, August). Autism spectrum disorder. *The Lancet*, *392*(10146), 508–520. Retrieved 2019-03-03, from https://linkinghub.elsevier.com/retrieve/pii/S0140673618311292 doi: 10.1016/S0140-6736(18)31129-2

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In (Vol. 1, pp. 281–297). Oakland, CA, USA. (Issue: 14)

Malta, T. M., Sokolov, A., Gentles, A. J., Burzykowski, T., Poisson, L., Weinstein, J. N., ... Gevaert, O. (2018). Machine learning identifies stemness features associated with oncogenic dedifferentiation. *Cell*, *173*(2), 338–354. (Publisher: Elsevier)

Manikonda, L., Hu, Y., & Kambhampati, S. (2014). Analyzing user activities, demographics, social network structure and user-generated content on Instagram. *arXiv preprint arXiv:1410.8099*.

Markello, R. D., Shafiei, G., Tremblay, C., Postuma, R. B., Dagher, A., & Misic, B. (2021, January). Multimodal phenotypic axes of Parkinson's disease. *npj Parkinson's Disease*, *7*(1), 6. Retrieved from https://doi.org/10.1038/s41531-020-00144-9 doi: 10.1038/s41531-020-00144-9

Mathews, J. C., Nadeem, S., Levine, A. J., Pouryahya, M., Deasy, J. O., & Tannenbaum, A. (2019, September). Robust and interpretable PAM50 reclassification exhibits survival advantage for myoepithelial and immune phenotypes. *npj Breast Cancer*, *5*(1), 30. Retrieved from https://doi.org/10.1038/s41523-019-0124-8 doi: 10.1038/s41523-019-0124-8

Matta, J., Zhao, J., Ercal, G., & Obafemi-Ajayi, T. (2018). Applications of node-based resilience graph theoretic framework to clustering autism spectrum disorders phenotypes. *Applied network science*, *3*(1), 38. doi: 10.1007/s41109-018-0093-0

Mirza, B., Wang, W., Wang, J., Choi, H., Chung, N. C., & Ping, P. (2019, January). Machine Learning and Integrative Analysis of Biomedical Big Data. *Genes*, *10*(2). (Place: Switzerland) doi: 10.3390/genes10020087

Mitra, S., Saha, S., & Hasanuzzaman, M. (2020, August). Multi-view clustering for multi-omics data using unified embedding. *Scientific Reports*, *10*(1), 13654. Retrieved from https://doi.org/10.1038/s41598-020-70229-1 doi: 10.1038/s41598-020-70229-1

Molenberghs, G., & Kenward, M. (2007). *Missing Data in Clinical Studies*. John Wiley & Sons. (Google-Books-ID: SuPJkcfdn1YC)

Monti, S., Tamayo, P., Mesirov, J., & Golub, T. (2003, July). Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data. *Machine Learning*, *52*(1), 91–118. Retrieved from https://doi.org/10.1023/A:1023949509487 doi: 10.1023/A:1023949509487

Murugesan, N., Cho, I., & Tortora, C. (2021). Benchmarking in Cluster Analysis: A Study on Spectral Clustering, DBSCAN, and K-Means. In T. Chadjipadelis, B. Lausen, A. Markos, T. R. Lee, A. Montanari, & R. Nugent (Eds.), *Data Analysis and Rationality in a Complex World* (pp. 175–185). Cham: Springer International Publishing. doi: 10.1007/978-3-030-60104-1_20

Nabi, R., Bhattacharya, R., Shpitser, I., & Robins, J. (2022). Causal and counterfactual views of missing data models. *arXiv preprint arXiv:2210.05558*.

Nakagawa, S., & Freckleton, R. P. (2008, November). Missing inaction: the dangers of ignoring missing data. *Trends in Ecology & Evolution*, *23*(11), 592–596. Retrieved from https://www.sciencedirect.com/science/article/pii/S0169534708002772 doi: 10.1016/j.tree.2008.06.014

Newman, M. (2018). *Networks*. Oxford university press.

Newman, M. E. (2003). Mixing patterns in networks. *Physical review E*, *67*(2), 026126. (Publisher: APS)

Newman, M. E. J. (2006a, September). Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, *74*(3). Retrieved 2019-03-02, from http://arxiv.org/abs/physics/0605087 (arXiv: physics/0605087) doi: 10.1103/PhysRevE.74.036104

Newman, M. E. J. (2006b). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, *103*(23), 8577–8582. Retrieved from https://www.pnas.org/content/103/23/8577 doi: 10.1073/pnas.0601602103

Newman, M. E. J., & Girvan, M. (2004, February). Finding and evaluating community structure in networks. *Physical Review E*, *69*(2), 026113. Retrieved from https://link.aps.org/doi/10.1103/PhysRevE.69.026113 (Publisher: American Physical Society) doi: 10.1103/PhysRevE.69.026113

Nie, C.-X., & Song, F.-T. (2018, April). Constructing financial network based on PMFG and threshold method. *Physica A: Statistical Mechanics and its Applications*, *495*, 104–113. Retrieved from https://www.sciencedirect.com/science/article/pii/S037843711731292X doi: 10.1016/j.physa.2017.12.037

Nielsen, T. O., Parker, J. S., Leung, S., Voduc, D., Ebbert, M., Vickery, T., ... Ellis, M. J. (2010, October). A Comparison of PAM50 Intrinsic Subtyping with Immunohistochemistry and Clinical Prognostic Factors in Tamoxifen-Treated Estrogen Receptor–Positive Breast Cancer. *Clinical Cancer Research*, *16*(21), 5222–5232. Retrieved 2024-01-14, from https://doi.org/10.1158/1078-0432.CCR-10-1282 doi: 10.1158/1078-0432.CCR-10-1282

Olopade, O. I., Grushko, T. A., Nanda, R., & Huo, D. (2008, December). Advances in Breast Cancer: Pathways to Personalized Medicine. *Clinical cancer research : an official journal of the American Association for Cancer Research*, *14*(24), 7988–7999. Retrieved 2024-01-14, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4535810/ doi: 10.1158/1078-0432.CCR-08-1211

P. Ni, J. Wang, P. Zhong, Y. Li, F. -X. Wu, & Y. Pan. (2020, June). Constructing Disease Similarity Networks Based on Disease Module Theory. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *17*(3), 906–915. doi: 10.1109/TCBB.2018.2817624

Pagani, G. A., & Aiello, M. (2013, June). The Power Grid as a complex network: A survey. *Physica A: Statistical Mechanics and its Applications*, *392*(11), 2688–2700. Retrieved from https://www.sciencedirect.com/science/article/pii/S0378437113000575 doi: 10.1016/j.physa.2013.01.023

Pai, S., & Bader, G. D. (2018, September). Patient Similarity Networks for Precision Medicine. *Theory and Application of Network Biology Toward Precision Medicine*, *430*(18, Part A), 2924–2938. Retrieved from https://www.sciencedirect.com/science/article/pii/S0022283618305321 doi: 10.1016/j.jmb.2018.05.037

Pai, S., Hui, S., Isserlin, R., Shah, M. A., Kaka, H., & Bader, G. D. (2019, March). netDx: interpretable patient classification using integrated patient similarity networks. *Molecular Systems Biology*, *15*(3), e8497. Retrieved 2022-10-26, from https://doi.org/10.15252/msb.20188497 (Publisher: John Wiley & Sons, Ltd) doi: 10.15252/msb.20188497

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Peixoto, T. P. (2014). The graph-tool python library. *figshare*. Retrieved 2014-09-10, from http://figshare.com/articles/graph_tool/1164194 doi: 10.6084/m9.figshare.1164194

Peixoto, T. P. (2018, January). Nonparametric weighted stochastic block models. *Phys. Rev. E*, *97*(1), 012306. Retrieved from https://link.aps.org/doi/10.1103/PhysRevE.97.012306 (Publisher: American Physical Society) doi: 10.1103/PhysRevE.97.012306

Peixoto, T. P. (2019, November). Bayesian Stochastic Blockmodeling. In *Advances in Network Clustering and Blockmodeling* (pp. 289–332). Retrieved 2023-07-28, from https://doi.org/10.1002/9781119483298.ch11 doi: 10.1002/9781119483298.ch11

Peixoto, T. P. (2021). *Descriptive vs. inferential community detection in networks: pitfalls, myths, and half-truths.* arXiv. Retrieved from https://arxiv.org/abs/2112.00183 (tex.copyright: Creative Commons Attribution Non Commercial Share Alike 4.0 International) doi: 10.48550/ARXIV.2112.00183

Piantadosi, S. (2005). *Clinical Trials: A Methodologic Perspective: Second Edition*. Wiley-Blackwell. doi: 10.1002/0471740136

Pons, P., & Latapy, M. (2006). Computing communities in large networks using random walks. *J. Graph Algorithms Appl.*, *10*(2), 191–218.

Radicchi, F., Fortunato, S., & Vespignani, A. (2011). Citation networks. *Models of science dynamics: Encounters between complexity theory and information sciences*, 233–257. (Publisher: Springer)

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, *66*(336), 846–850. (Publisher: Taylor & Francis)

Rappoport, N., & Shamir, R. (2018, November). Multi-omic and multi-view clustering algorithms: review and cancer benchmark. *Nucleic Acids Research*, *46*(20), 10546–10562. Retrieved 2024-01-14, from https://doi.org/10.1093/nar/gky889 doi: 10.1093/nar/gky889

Rappoport, N., & Shamir, R. (2019, September). NEMO: cancer subtyping by integration of partial multi-omic data. *Bioinformatics*, *35*(18), 3348–3356. Retrieved 2023-05-07, from https://doi.org/10.1093/bioinformatics/btz058 doi: 10.1093/bioinformatics/btz058

Raven, J. (2003). Raven progressive matrices. In *Handbook of nonverbal assessment* (pp. 223–237). Springer.

Reynolds, D. A. (2009). Gaussian mixture models. *Encyclopedia of biometrics*, *741*(659-663). (Publisher: Berlin, Springer)

Rosenberg, A., & Hirschberg, J. (2007). V-measure: A conditional entropy-based external cluster evaluation measure. In (pp. 410–420).

Rosvall, M., & Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal community structure. *Proceedings of the national academy of sciences*, *105*(4), 1118–1123. (Publisher: National Acad Sciences)

Rousseeuw, P. J. (1987, November). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65. Retrieved from https://www.sciencedirect.com/science/article/pii/0377042787901257 doi: 10.1016/0377-0427(87)90125-7

Ruan, J., Dean, A. K., & Zhang, W. (2010, February). A general co-expression network-based approach to gene expression analysis: comparison and applications. *BMC Systems Biology*, *4*(1), 8. Retrieved 2021-03-10, from https://doi.org/10.1186/1752-0509-4-8 doi: 10.1186/1752-0509-4-8

Rubin, D. B. (2018). Multiple imputation. In *Flexible Imputation of Missing Data, Second Edition* (2nd ed.). Chapman and Hall/CRC. (Num Pages: 34)

Rutter, M., Le Couteur, A., & Lord, C. (2003). Autism diagnostic interview-revised. *Los Angeles, CA: Western Psychological Services*, *29*(2003), 30.

Ryan, B., Marioni, R. E., & Simpson, T. I. (2023). Multi-Omic Graph Diagnosis (MOGDx) : A data integration tool to perform classification tasks for heterogeneous diseases. *medRxiv : the preprint server for health sciences*. Retrieved from https://www.medrxiv.org/content/early/2023/07/09/2023.07.09.23292410 (Publisher: Cold Spring Harbor Laboratory Press tex.elocation-id: 2023.07.09.23292410 tex.eprint: https://www.medrxiv.org/content/early/2023/07/09/2023.07.09.23292410.full.pdf) doi: 10.1101/2023.07.09.23292410

S. Islam, A. Abbasi, N. Agarwal, W. Zheng, G. Doretto, & D. A. Adjeroh. (2021, December). Detecting Drug-Drug Interactions using Protein Sequence-Structure Similarity Networks. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (pp. 3472–3477). (Journal Abbreviation: 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)) doi: 10.1109/BIBM52615.2021.9669858

Saade, A., Krzakala, F., & Zdeborová, L. (2014). Spectral clustering of graphs with the bethe hessian. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, & K. Weinberger (Eds.), *Advances in neural information processing systems* (Vol. 27). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2014/file/63923f49e5241343aa7acb6a06a751e7-Paper.pdf

Santiago-Rodriguez, T. M., & Hollister, E. B. (2021, October). Multi 'omic data integration: A review of concepts, considerations, and approaches. *Neonatal GI: The Intestinal Microbiome*, *45*(6), 151456. Retrieved from https://www.sciencedirect.com/science/article/pii/S0146000521000707 doi: 10.1016/j.semperi.2021.151456

Saria, S., & Goldenberg, A. (2015). Subtyping: What it is and its role in precision medicine. *IEEE Intelligent Systems*, *30*(4), 70–75. (Publisher: IEEE)

Scalfani, V. F., Patel, V. D., & Fernandez, A. M. (2022, December). Visualizing chemical space networks with RDKit and NetworkX. *Journal of Cheminformatics*, *14*(1), 87. Retrieved from https://doi.org/10.1186/s13321-022-00664-x doi: 10.1186/s13321-022-00664-x

Schaub, M. T., Delvenne, J.-C., Rosvall, M., & Lambiotte, R. (2017, February). The many facets of community detection in complex networks. *Applied Network Science*, *2*(1), 4. Retrieved from https://doi.org/10.1007/s41109-017-0023-6 doi: 10.1007/s41109-017-0023-6

Serra, A., Fratello, M., Fortino, V., Raiconi, G., Tagliaferri, R., & Greco, D. (2015, August). MVDA: a multi-view genomic data integration methodology. *BMC Bioinformatics*, *16*(1), 261. Retrieved from https://doi.org/10.1186/s12859-015-0680-3 doi: 10.1186/s12859-015-0680-3

Siegel, M., Smith, K. A., Mazefsky, C., Gabriels, R. L., Erickson, C., Kaplan, D., ... Santangelo, S. L. (2015, December). The autism inpatient collection: methods and preliminary sample description. *Molecular Autism*, *6*(1). Retrieved 2019-10-04, from http://www.molecularautism.com/content/6/1/61 doi: 10.1186/s13229-015-0054-8

Simons VIP Consortium. (2012). Simons Variation in Individuals Project (Simons VIP): a genetics-first approach to studying autism spectrum and related neurodevelopmental disorders. *Neuron*, *73*(6), 1063–1067. (Publisher: Elsevier)

Sokal, R. R., & Rohlf, F. J. (1962). The comparison of dendrograms by objective methods. *Taxon*, 33–40. (Publisher: JSTOR)

Sparrow, S. S., & Cicchetti, D. V. (1989). *The Vineland adaptive behavior scales.* Allyn & Bacon.

Steinley, D. (2004). Properties of the Hubert-Arable Adjusted Rand Index. *Psychological Methods*, *9*(3), 386–396. (Place: US Publisher: American Psychological Association) doi: 10.1037/1082-989X.9.3.386

Sterne, J. A. C., White, I. R., Carlin, J. B., Spratt, M., Royston, P., Kenward, M. G., ... Carpenter, J. R. (2009, June). Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls. *BMJ*, *338*, b2393. Retrieved from http://www.bmj.com/content/338/bmj.b2393.abstract doi: 10.1136/bmj.b2393

Stiglic, G., Kocbek, P., Fijacko, N., Sheikh, A., & Pajnkihar, M. (2019). Challenges associated with missing data in electronic health records: A case study of a risk prediction model for diabetes using data from Slovenian primary care. *Health Informatics Journal*, *25*(3), 951–959. Retrieved from https://doi.org/10.1177/1460458217733288 (_eprint: https://doi.org/10.1177/1460458217733288) doi: 10.1177/1460458217733288

Su, C., Tong, J., Zhu, Y., Cui, P., & Wang, F. (2018, December). Network embedding in biomedical data science. *Briefings in Bioinformatics*. Retrieved 2019-09-13, from https://academic.oup.com/bib/advance-article/doi/10.1093/bib/bby117/5228144 doi: 10.1093/bib/bby117

Sun, Z.-B., Zou, X.-W., Guan, W., & Jin, Z.-Z. (2006, January). The architectonic fold similarity network in protein fold space. *The European Physical Journal B - Condensed Matter and Complex Systems*, *49*(1), 127–134. Retrieved from https://doi.org/10.1140/epjb/e2006-00026-0 doi: 10.1140/epjb/e2006-00026-0

Tomczak, K., Czerwińska, P., & Wiznerowicz, M. (2015). Review The Cancer Genome Atlas (TCGA): an immeasurable source of knowledge. *Contemporary Oncology/Współczesna Onkologia*, *2015*(1), 68–77. (Publisher: Termedia)

Tong, T., Gray, K., Gao, Q., Chen, L., & Rueckert, D. (2017, March). Multi-modal classification of Alzheimer's disease using nonlinear graph fusion. *Pattern Recognition*, *63*, 171–181. Retrieved from https://www.sciencedirect.com/science/article/pii/S0031320316303247 doi: 10.1016/j.patcog.2016.10.009

Traag, V. A., Waltman, L., & van Eck, N. J. (2019, March). From Louvain to Leiden: guaranteeing well-connected communities. *Scientific Reports*, *9*(1), 5233. Retrieved from https://doi.org/10.1038/s41598-019-41695-z doi: 10.1038/s41598-019-41695-z

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., ... Altman, R. B. (2001, June). Missing value estimation methods for DNA microarrays. *Bioinformatics*, *17*(6), 520–525. Retrieved 2024-01-28, from https://doi.org/10.1093/bioinformatics/17.6.520 doi: 10.1093/bioinformatics/17.6.520

Valavanis, I., Spyrou, G., & Nikita, K. (2010, April). A similarity network approach for the analysis and comparison of protein sequence/structure sets. *Journal of Biomedical Informatics*, *43*(2), 257–267. Retrieved from https://www.sciencedirect.com/science/article/pii/S1532046410000134 doi: 10.1016/j.jbi.2010.01.005

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Verhaak, R. G., Hoadley, K. A., Purdom, E., Wang, V., Qi, Y., Wilkerson, M. D., ... Hayes, D. N. (2010, January). Integrated Genomic Analysis Identifies Clinically Relevant Subtypes of Glioblastoma Characterized by Abnormalities in PDGFRA, IDH1, EGFR, and NF1. *Cancer Cell*, *17*(1), 98–110. Retrieved 2022-12-06, from https://doi.org/10.1016/j.ccr.2009.12.020 (Publisher: Elsevier) doi: 10.1016/j.ccr.2009.12.020

Vinh, N. X., Epps, J., & Bailey, J. (2009). Information theoretic measures for clusterings comparison: Is a correction for chance necessary? In *Proceedings of the 26th annual international conference on machine learning* (pp. 1073–1080). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/1553374.1553511 (Number of pages: 8 Place: Montreal, Quebec, Canada) doi: 10.1145/1553374.1553511

Voillet, V., Besse, P., Liaubet, L., San Cristobal, M., & González, I. (2016, October). Handling missing rows in multi-omics data integration: multiple imputation in multiple factor analysis framework. *BMC Bioinformatics*, *17*(1), 402. Retrieved from https://doi.org/10.1186/s12859-016-1273-5 doi: 10.1186/s12859-016-1273-5

von Luxburg, U. (2007, December). A tutorial on spectral clustering. *Statistics and Computing*, *17*(4), 395–416. Retrieved from https://doi.org/10.1007/s11222-007-9033-z doi: 10.1007/s11222-007-9033-z

Vrandečić, D., & Krötzsch, M. (2014, September). Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, *57*(10), 78–85. Retrieved 2024-01-12, from https://dl.acm.org/doi/10.1145/2629489 doi: 10.1145/2629489

Wang, B., Mezlini, A. M., Demir, F., Fiume, M., Tu, Z., Brudno, M., ... Goldenberg, A. (2014, March). Similarity network fusion for aggregating data types on a genomic scale. *Nature Methods*, *11*(3), 333–337. Retrieved from https://doi.org/10.1038/nmeth.2810 doi: 10.1038/nmeth.2810

Wang, C., Machiraju, R., & Huang, K. (2014, June). Breast cancer patient stratification using a molecular regularized consensus clustering method. *Systems Biology with Omics Data*, *67*(3), 304–312. Retrieved from https://www.sciencedirect.com/science/article/pii/S104620231400098X doi: 10.1016/j.ymeth.2014.03.005

Wang, Q., Downey, C., Wan, L., Mansfield, P. A., & Moreno, I. L. (2018). Speaker diarization with LSTM. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 5239–5243). IEEE.

Wang, S., Celebi, M. E., Zhang, Y.-D., Yu, X., Lu, S., Yao, X., ... Tyukin, I. (2021, December). Advances in Data Preprocessing for Biomedical Data Fusion: An Overview of the Methods, Challenges, and Prospects. *Information Fusion*, *76*, 376–421. Retrieved from https://www.sciencedirect.com/science/article/pii/S1566253521001354 doi: 10.1016/j.inffus.2021.07.001

Wells, B. J., Chagin, K. M., Nowacki, A. S., & Kattan, M. W. (2013). Strategies for handling missing data in electronic health record derived data. *EGEMS (Washington, DC)*, *1*(3), 1035. (Place: England) doi: 10.13063/2327-9214.1035

Wiggins, L. D., Tian, L. H., Levy, S. E., Rice, C., Lee, L.-C., Schieve, L., ... Thompson, W. (2017, November). Homogeneous Subgroups of Young Children with Autism Improve Phenotypic Characterization in the Study to Explore Early Development. *Journal of Autism and Developmental Disorders*, *47*(11), 3634–3645. Retrieved from https://doi.org/10.1007/s10803-017-3280-4 doi: 10.1007/s10803-017-3280-4

Wilson, B. N., Crawford, S. G., Green, D., Roberts, G., Aylott, A., & Kaplan, B. J. (2009). Psychometric properties of the revised developmental coordination disorder questionnaire. *Physical & occupational therapy in pediatrics*, *29*(2), 182–202. (Publisher: Taylor & Francis)

Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, *2*(1-3), 37–52. (Publisher: Elsevier)

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2021). A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(1), 4–24. doi: 10.1109/TNNLS.2020.2978386

Xu, J., Li, C., Ren, Y., Peng, L., Mo, Y., Shi, X., & Zhu, X. (2022, June). Deep Incomplete Multi-View Clustering via Mining Cluster Complementarity. *Proceedings of the AAAI Conference on Artificial Intelligence*, *36*(8), 8761–8769. Retrieved 2023-07-05, from https://ojs.aaai.org/index.php/AAAI/article/view/20856 (Section: AAAI Technical Track on Machine Learning III) doi: 10.1609/aaai.v36i8.20856

Y. Yang, & H. Wang. (2018, June). Multi-view clustering: A survey. *Big Data Mining and Analytics*, *1*(2), 83–107. doi: 10.26599/BDMA.2018.9020003

Yang, J., & Leskovec, J. (2012). Defining and evaluating network communities based on ground-truth. In *Proceedings of the ACM SIGKDD workshop on mining data semantics.* New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/2350190.2350193 (Number of pages: 8 Place: Beijing, China tex.articleno: 3) doi: 10.1145/2350190.2350193

Yang, Z., Algesheimer, R., & Tessone, C. J. (2016). A comparative analysis of community detection algorithms on artificial networks. *Scientific reports*, *6*(1), 30750. (Publisher: Nature Publishing Group UK London)

Zahoránszky-Kőhalmi, G., Bologa, C. G., & Oprea, T. I. (2016, March). Impact of similarity threshold on the topology of molecular similarity networks and clustering outcomes. *Journal of Cheminformatics*, *8*(1), 16. Retrieved from https://doi.org/10.1186/s13321-016-0127-5 doi: 10.1186/s13321-016-0127-5

Zhao, H., Ding, Z., & Fu, Y. (2017, February). Multi-View Clustering via Deep Matrix Factorization. *Proceedings of the AAAI Conference on Artificial Intelligence*, *31*(1). Retrieved 2024-01-14, from https://ojs.aaai.org/index.php/AAAI/article/view/10867 (Section: Machine Learning Methods) doi: 10.1609/aaai.v31i1.10867

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., ... Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI open*, *1*, 57–81. (Publisher: Elsevier)