Computational Statistics
https://doi.org/10.1007/s00180-024-01536-8

ORIGINAL PAPER

INet for network integration

Valeria Policastro[1,2] · **Matteo Magnani**[3] · **Claudia Angelini**[2] · **Annamaria Carissimo**[2]

Received: 12 March 2024 / Accepted: 5 August 2024

Abstract

When collecting several data sets and heterogeneous data types on a given phenomenon of interest, the individual analysis of each data set will provide only a particular view of such phenomenon. Instead, integrating all the data may widen and deepen the results, offering a better view of the entire system. In the context of network integration, we propose the `INet` algorithm. `INet` assumes a similar network structure, representing latent variables in different network layers of the same system. Therefore, by combining individual edge weights and topological network structures, `INet` first constructs a `Consensus Network` that represents the shared information underneath the different layers to provide a global view of the entities that play a fundamental role in the phenomenon of interest. Then, it derives a `Case Specific Network` for each layer containing peculiar information of the single data type not present in all the others. We demonstrated good performance with our method through simulated data and detected new insights by analyzing biological and sociological datasets.

Keywords Network · Integration · Consensus network · Case specific networks · Algorithm · Multilayer network

The work has been started at Department of Environmental Biological and Pharmaceutical Sciences and Technologies, University of Campania "Luigi Vanvitelli", 81100 Caserta, Italy.

✉ Valeria Policastro
valeria.policastro@unina.it

[1] Department of Political Science, University of Naples Federico II, 80138 Naples, Italy

[2] Istituto per le Applicazioni del Calcolo "Mauro Picone", Consiglio Nazionale delle Ricerche (CNR), 80131 Naples, Italy

[3] Department of Information Technology, Division of Computing Science, Uppsala University, Uppsala, Sweden

Published online: 04 September 2024

1 Introduction

Networks are becoming a prevalent approach to describe complex relationships among units in complex systems and to extract information at different levels, Barabási (2016); Pržulj (2019); Vulliard and Menche (2021). Modeling a complex system as a network constitutes a flexible representation of the insights since nodes can represent either statistical units or observed variables according to specific needs, with edges representing distance metrics or correlation between units or probabilistic relationships among variables, such as in the graphical model framework. Given a representation of the system in terms of a graph, it is possible to study the network topology using different measures, identify the communities, and understand diffusion phenomena. In the literature, there are several approaches to build networks from observed datasets (Váša et al. 2018; Korhonen et al. 2021).

Nowadays, vast amounts of data are being gathered across diverse domains, encompassing biomedicine, sociology, and the physical sciences. When dealing with multiple data sets associated with a phenomenon of interest, the individual analysis of each data set yields only a limited perspective on such phenomenon. In contrast, integrating all the data may widen and deepen the results, giving a more global view of the entire system. Therefore, when data are represented as networks, integrating networks from different data sources on the same nodes allows for a better representation of shared structure and understanding of case-specific peculiarity.

The interest in network integration arises from several applied contexts. For instance, in biomedical analysis, patients affected by a particular disease can be described using networks according to specific biological data. However, diseases are not always caused by a single factor but might occur due to different factors. Therefore, data analysis of a single component only allows a partial understanding of patient similarities. In sociological analysis, the study of the online relationships between people should not be limited to their connections on a single social networking website such as Facebook or Twitter. One must consider different contexts to understand their relations' broad and complex structure. Similarly, in text analysis, analyzing a topic (e.g., co-occurrence of terms) only from one source, for example, one journal/book or one website, can give only a specific view of it, while merging different sources will give a more exhaustive view of the entire topic.

The advantages of network integration are shown in different papers. In the biomedical field, for example, similarity network fusion (SNF) (Wang et al. 2014) integrates patient networks obtained from different omics. Then, it performs community detection on the resulting network. The integrated analysis led to the identification of significantly different survival profiles in five cancer datasets that could not be identified with the analysis of the individual omics data. In a similar framework, iCell (Malod-Dognin et al. 2019) presented a method that better captures enriched clusters in cancers than the single data analysis. In the textual field, Rondinelli et al. (2020) constructed an a-priori aggregation of the books to analyze the relationships between biblical names'. Despite these diverse approaches, a standard methodology for integrating networks from different sources considering the network structure, is lacking.

This work presents our novel statistical method, INet, for data integration, which is peculiar in that it uses a multilayer network structure and the network topology as a basis for integration. According to Interdonato et al. (2020), our method can be defined as a “specific” case of flattening where we not only consider the edges, but we introduce, as a novelty, a significant role to the neighborhood contributing to the robustness of the relationship between pairs of nodes. INet generate a Consensus Network capable of pulling out important information about the phenomenon under study, assuming that the structure underneath the different layers has some similarity that we want to emerge. Moreover, INet generates something not present in other works, the Case Specific Networks, one for each layer, which captures the unique information exclusive to that specific layer, distinct from all the others. Overall, the method is versatile and flexible in handling the integration across samples and variables, which yielded analytical enhancements across diverse fields. For example, we uncovered patient similarities undetectable through single data type analysis and revealed different political scenarios by analyzing the Case Specific Networks, demonstrating the method’s broad applicability. Finally, the availability of the algorithm INet as the R-package INetTool contributes to its usage in practical applications.

This paper is structured as follows. Section 2 presents the theoretical background underpinning this research, Sect. 3 presents our method, Sect. 4 describes the simulation study and the performance of our method, and Sect. 5 provides the results of an application of INet to two case studies. Concluding remarks are given in Sect. 6.

2 Integration via network

Nowadays, data integration is a widespread research area that demonstrates the capacity to fill the gap between producing a huge amount of data and extracting information for their interpretation. Given $X^{(1)}, X^{(2)}, \dots, X^{(N)}$, $N \geq 2$ datasets, where each data set $X^{(i)}$ is a matrix $X_{(n_i \times p_i)}$ with n_i denoting the samples and p_i the variables or features, one can define the *n-Integration* as the integration of the same $n_i = n$ samples measured across the N datasets collecting p_i different type of variables or the *p-Integration* as the integration of the same $p_i = p$ variables observed over N different sets of n_i samples, see for example (Rohart et al. 2017). Moreover, such integrative approaches allow for a better consideration of the individual matrices’ heterogeneity, which can also contain different data types, and the simple data merging will destroy such specificity. For instance, when analyzing omics data (i.e., genomic, transcriptomic, and epigenomic) on the same patients, the *n-Integration* methods can allow significant aspects of the disease to emerge across the different omics. Moreover, the *n-Integration* integration allows the proper handling of transcriptomic data as quantitative (real or count-value) representing gene expression, methylation data as values in the [0, 1] interval providing the amount of methylation, and genomic data as binary variables coding genome mutations. Conversely, when performing a meta-analysis of the same clinical or omic variables using data sets collected in different laboratories, hospitals, or studies, the *p-Integration* method can improve the power of the analysis and better capture the shared information. Although, in this case, the individual matrices are of the same data type, they can have different probability distributions. These integrative approaches can also be referred to as horizontal and vertical integration as in Argelaguet et al. (2021).

Network integration is an approach that allows both *n*- and *p-Integration* since instead of working directly on the data domain $X^{(i)}$, it is applied after the construction of a network of relationships. Given a dataset $X^{(i)}$, a network can be inferred regardless of whether the nodes are variables or samples. Specifically, a network is a mathematical representation of relationships among the units of a complex system and can be represented by a graph. A *graph* $G = (V, E)$ consists of a collection of V vertices (or nodes) corresponding to the individual units of the observed system (e.g., people, patients, genes, words, documents) and a collection of E edges (or links), indicating some relations between them. When the nodes represent variables, the edges are often obtained using probability measures such as correlations, partial correlations, or mutual information; instead, when nodes are statistical units, the edges can be obtained as a similarity measure or others. Hence, the specific construction of the network from a given dataset $X^{(i)}$ depends on the types of data encoded and the scientific focus.

A *graph* $G = (V, E, W)$ is a weighted structure where W represents the strength of the edge E, i.e., the strength of the relationship. Both weighted and unweighted graphs G can be represented through an adjacency matrix A of dimension $|V| \times |V|$ where |*V*| denotes the number of nodes of G. For unweighted graphs, the adjacency matrix A is a square sparse binary matrix where the entry $a_{i,j}$ is set to 1 when an edge exists between the nodes (i, j) and 0 otherwise. Similarly, for weighted graphs, the entry $a_{i,j}$ is set to the value $w_{i,j} > 0$ in correspondence to the presence of the edge between the nodes (i, j) and zero otherwise. The larger $w_{i,j}$ is, the stronger the relationship.

Different networks can represent the same complex system when multiple data sets are available. In this case, each network can be viewed as a layer in a multilayer network, and an integration of this information is required to synthesize the knowledge. Researchers used many and various approaches for network integration, ranging from the aggregated approach, where edges present in at least one network are retained, to the strict consensus approach, where only edges present in all networks are preserved. The mean approach was also utilized for weighted networks, which associates each edge with the mean of its weights across all networks. We will compare this last approach to our method in Sect. 4.1. Although the advantages of network integration are shown in various papers, no existing methods for multilayer networks are as broad and versatile as `INet`. In the biomedical field, more customized methods have been developed, such as iCell (Malod-Dognin et al. 2019) and SNF Wang et al. (2014). The last one involves an iterative procedure that updates the similarity matrix for each data type, considering local affinity through k-nearest neighbours. In contrast, `INet` operates on a multilayer network rather than a similarity matrix. Additionally, `INet` does not require setting a parameter k for the neighbours, as it considers all the neighbours in the layers. Furthermore, `INet` constructs a network to which any community detection algorithm can be applied, allowing the detection of clusters without the need for a-priori knowledge of the number of clusters, unlike the SNF procedure.

Furthermore, examples of works with a similar aim to detect common information include those by Núñez-Carpintero et al. (2019) and Giordano et al. (2019). The first applies a multilayer community detection method to establish the association between disease and genes, and the second one employs a factorization technique on multilayer networks to construct a compromise space. However, neither of these approaches constructs an integrated network. Another distinguishing aspect of our approach is the construction of the Case Specific Network, which investigates the unique information that remains in the specific layer. Overall, using INet provides significant insights across various fields, as Sect. 5 demonstrates.

3 The proposed method: INet

INet uses a message-passing theory, i.e., starting from a multilayer weighted network, it iteratively modifies the layers, updating the edge weights to make them more similar at each step. In this way, the algorithm constructs a Consensus Network that preserves the edges and the common structures underneath the networks, as entities in the same neighborhood tend to have functional similarities, making them highly relevant to the phenomenon of interest.

Additionally, from the Consensus Network, INet generates also Case Specific Networks, one for each layer.

The idea of our method is illustrated in Fig. 1.

The weight update involves two components that we denominated as the Edge weight component and the Ego network component. The latter term arises from a specific type of network, known as the Ego-centric network or, briefly, Ego network, which comprises a focal node ("ego") and the nodes directly connected to the ego ("alters") along with the connections between them (Biswas and Biswas 2015).

Figure 2 illustrates a simplified representation of our algorithm. In brief, at each iteration, INet tries to update all edge weights in each layer, incorporating the information from the other layers in the new proposal. After this proposal step, INet computes a network distance between the layers to decide whether

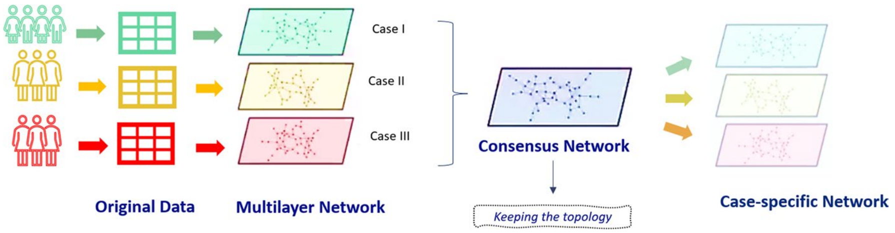


Fig. 1 *Method idea*: From a multilayer weighted network where each layer represents a different dataset (e.g., same features from different patients), we firstly construct a Consensus Network, which preserves the common structure underneath the layers and secondly form a Case Specific Network for each case study

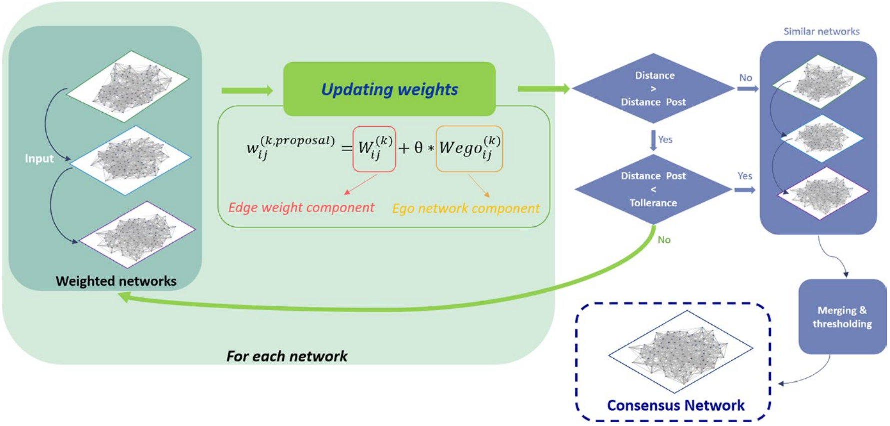


Fig. 2 `INet` algorithm

to accept or reject the proposed weights. The network weights are updated when the proposal is accepted, so the layers get closer. If the distance among the networks is less than a predefined tolerance, denoted as (*tol*), the iteration stops; otherwise, it continues for another update until the distance tolerance or a maximum number of iterations is reached. When `INet` stops, it returns "similar" networks that are combined and subjected to a thresholding process to generate the `Consensus Network`.

The rationale behind incorporating an Ego network component in weight updates comes from the triadic closure property (Simmel 1908), which asserts that nodes sharing several common neighbors are more likely to form connections. Our integrative procedure removes weak connections (low-weight edges) between nodes without common neighbors, reduces the noise in each layer, and preserves relevant connections (high-weight edges) when present in more layers or when nodes share common neighbors.

To be more precise, we provide the following notations and the mathematical formulation of the `INet` algorithm. Consider a multilayer network M consisting of N layers, represented through $A^{(1)}, \dots, A^{(k)}, \dots, A^{(N)}$ adjacency matrices. From here on in the paper:

(i, j) denotes the undirected edge between vertices i and j,
$w_{i,j}^{(k)}$ is the entry of $A^{(k)}$ corresponding to the weight of the edge (i, j) in layer k,
$ne_{(i|j)}^{(k)}$ is the set of neighbours of node i excluding j in layer k,
$Cne^{(k)} = ne_{(i|j)}^{(k)} \cap ne_{(j|i)}^{(k)}$ represents the set of common neighbours of nodes i and j in layer k,
a denotes a generic common neighbour of i and j,
n_a is the number of layers where i and j have node a as a common neighbour.

Then, we first define the weight update proposal and introduce the multilayer network distance; after that, we give the explicit formulation of the algorithm.

3.1 Weight updating

The core of INet consists of the network weight updating during the iterations. At each iteration, INet updates all weights $w_{i,j}^{(k)}$ of the adjacency matrices $A^{(k)}$ for $k = 1, \dots, N$, for (i, j) belonging to the union of the edges from all layers of the multilayer network (denoted in the Algorithm 1 as *Union.Edgelist*). The weights are not updated if the $w_{i,j}^{(k)} = 0$ for all $k = 1, \dots, N$. The update formula considers the N layers individually, updating their weights simultaneously. In the following, INet distinguishes between the network under update and the networks on the other layers since the update formula gives more relevance to the weight of the currently updated layer. More specifically, at each step, the algorithm proposes an update of the weight $w_{i,j}^{(k,proposal)}$ of the edge *(i,j)* in the layer k as follows:

$$w_{ij}^{(k,proposal)} = W_{ij}^{(k)} + \theta * Wego_{ij}^{(k)} \qquad k = 1, \dots, N \tag{1}$$

where $W_{ij}^{(k)}$ and $Wego_{ij}^{(k)}$ represent the Edge weight component and Ego network component, respectively, and θ is a parameter in [0, 1] that gives different relevance to the Ego network component. At the end of each iteration over the N layers, INet computes the goodness of the updated proposals evaluating a network distance. If the update step is successful, the weights are updated as $w_{ij}^{(k)} = w_{ij}^{(k,proposal)}$, otherwise $w_{ij}^{(k)}$ remains unchanged and the algorithm terminates.

The two components in the updated proposal in Eq. (1) are defined as follows:

Definition 1 For a given edge (i,j), the Edge weight component $W_{ij}^{(k)}$ is given by

$$W_{ij}^{(k)} = \frac{w_{ij}^{(k)} + \frac{\sum_{l \neq k} w_{ij}^{(l)}}{N-1}}{2}, \tag{2}$$

i.e., the mean of the edge weight within the k^{th} layer and the mean of the weights of the same edge in the other layers.

Definition 2 For a given edge (i,j), the Ego network component $Wego_{ij}^{(k)}$ is given by

$$Wego_{ij}^{(k)} = \frac{EgoMeasure_{ij}^{(k)} + \frac{\sum_{l \neq k} EgoMeasure_{ij}^{(l)}}{N-1}}{2} \tag{3}$$

i.e., the mean of the ego measure of the k^{th} layer and the mean of the ego measures across all the other layers, where the ego measure is defined as:

$$EgoMeasure_{ij}^{(k)} = \left[\frac{\sum_{a \in Cne^{(k)}} \frac{n_a}{N} \left[w^{(k)}_{(i,a^{(k)})} + w^{(k)}_{(j,a^{(k)})} \right]}{|ne_{(i|j)}{}^{(k)}| + |ne_{(j|i)}{}^{(k)}|} \right]^{\frac{1}{|Cne^{(k)}|}} . \tag{4}$$

3.2 Multilayer network distance

`INet` uses a multilayer network distance to evaluate the update proposal's goodness and decide when to terminate the iterations. The multilayer network distance is obtained from the Jaccard similarity measure (Jaccard 1901), as follows. Given two vectors $\mathbf{x} = (x_1, x_2, \dots, x_n)$ and $\mathbf{y} = (y_1, y_2, \dots, y_n)$ with all real $x_i, y_i \geq 0$, the Jaccard weighted similarity is defined as:

$$J_{\mathcal{W}}(\mathbf{x}, \mathbf{y}) = \frac{\sum_i \min(x_i, y_i)}{\sum_i \max(x_i, y_i)}. \tag{5}$$

Hence, the Jaccard distance, also known as the Soergel distance, is Deza et al. (2009)

$$d_{J\mathcal{W}}(\mathbf{x}, \mathbf{y}) = 1 - J_{\mathcal{W}}(\mathbf{x}, \mathbf{y}). \tag{6}$$

From here, we define the Mean Weighted Jaccard Distance for Multilayer Networks to quantify the difference among the different layers of the multilayer network, and we used this distance in our algorithm. The distance is defined as follows:

Definition 3 The Mean Weighted Jaccard Distance for a Multilayer Networks M is the mean of the weighted Jaccard distances between all pairs of layers:

$$d_{J\mathcal{W}}(M) = \frac{1}{|S|} \sum_{(p,t) \in S} \left[1 - \frac{\sum_{i,j \in V} \min(w_{i,j}^{(p)}, w_{i,j}^{(t)})}{\sum_{i,j \in V} \max(w_{i,j}^{(p)}, w_{i,j}^{(t)})} \right], \tag{7}$$

where S is the set of all the possible pairs of layers (p, t) of the multilayer network and $|S|$ is equal to the binomial coefficient $\binom{N}{2} = \frac{N!}{2!(N-2)!}$.

This distance ranges between 0 and 1, where values closer to 0 indicate high similarity between the layers, while values near 1 suggest dissimilarity.

3.3 Algorithms

Algorithm 1 is the core of `INet`. In each iteration, the algorithm updates the weights within each adjacency matrix of the multilayer network by computing the proposed weights using Eq. (1). It accepts the proposal if the distance $d_{J\mathcal{W}}$ in

Eq. (7) decreases, meaning that the networks will be more similar after the update step. Otherwise, the weights remain unchanged, and the algorithm terminates. At the end of each iteration, the algorithm verifies if the network distance d_{JW} is below a predefined tolerance (*tol*), stopping the procedure if this condition is met. Alternatively, it proceeds with a new iteration. Upon reaching the tolerance, INet computes the mean adjacency matrix whose entries are the means of the weights and constructs the Consensus Network after the thresholding step that retains significant edges.

Algorithm 1 Consensus Network

```
Input: $\mathbf{A}^{(1)}, ..., \mathbf{A}^{(k)}, .., \mathbf{A}^{(N)} = \mathbf{M}$, $\theta$, $tol$, $niter_{\max}$ and $threshold$
step=0

Repeat
$D \leftarrow d_{JW}(\mathbf{M^{current}})$
for $k = 1, \ldots, N$ do
  for $(i, j) \in Union.Edgelist$ do
        $w_{ij}^{(k,proposal)} = W_{ij}^{(k)} + \theta * Wego_{ij}^{(k)}$
  end for
end for
$DPost \leftarrow d_{JW}(\mathbf{M^{proposal}})$
if $D > DPost$ then
  $\mathbf{M^{current}} \leftarrow M^{proposal}$
else
  $Break$ return $\mathbf{M^{current}}$
end if
step=step+1
Until $DPost < tol$ or $step > niter_{\max}$

Define the mean adjacency matrix A such that:
$\mathbf{A}_{i,j} \leftarrow \frac{\sum_k w_{ij}^{(k,current)}}{N}$

Define the Consensus Network such that:
$Consensus.Network_{i,j} \leftarrow \mathbf{A}_{i,j}[\mathbf{A}_{i,j} > threshold]$

Output: $Consensus.Network$
```

Algorithm 1 provides the algorithm's pseudocode. The algorithm takes as input the adjacency matrices of the multilayer weighted network, the parameter θ that determines the relevance assigned to the Ego network component, the tolerance (*tol*) representing the acceptable distance between layers, the maximum number of iterations ($niter_{\max}$), and the *threshold* for constructing the consensus network after merging the similar networks. The output of Algorithm 1 is the Consensus Network.

Algorithm 2 Case Specific Networks

```
Input: $\mathbf{A}^{(1)}, ...\mathbf{A}^{(k)}.., \mathbf{A}^{(N)}$, $\boldsymbol{Consensus.Network}$
for $k = 1, \ldots, N$ do
    $\mathbf{S}^{(k)} \leftarrow \mathbf{A}^{(k)} - \boldsymbol{Consensus.Network}$
    $percSpecificity^{(k)} \leftarrow ecount(\mathbf{S}^{(k)})/ecount(\mathbf{A}^{(k)})$
end for
Output: $\mathbf{S}$, $percSpecificity$
```

While the `Consensus Network` shows a shared network structure among the different layers, it is also crucial to investigate the differences between the layers in several applications. For this purpose, `INet` constructs the `Case Specific Networks` given by the differences in terms of edges between the original networks and the `Consensus Network`. This step is accomplished by using the respective adjacency matrices. A `Case specific network` provides different insights into a specific layer. Additionally, as a global measure, we compute the specificity percentage for each layer, representing the proportion of specific edges compared to the total initial edges.

R package The R package `INetTool` implements the above-described algorithms and contains other additional routines valid for pre or post-analysis. All the functions are listed in Supplementary Table 3. The method implementation and a vignette for the usage are available on the GitHub page https://github.com/ValeriaPolicastro/INet-Tool.

4 Simulations

To assess the performance of our method, we used simulations generating different networks using the LFR benchmark (Lancichinetti et al. 2008), with different parameters, as detailed in Table 1. This new class of benchmarks consists of graphs in which node degree and community size distributions are power laws with tunable exponents. Hence, they allow us to model essential features of real networks.

To investigate the impact on the number of nodes and the modularity, we generated networks of different sizes N (100, 500, and 1000 nodes), and for each case, we

Table 1 Simulation settings

Parameters	Definitions	Settings
N	Number of nodes	100-500-1000
k	Average degree	5-25-50
maxk	Maximum degree	10-50-100
ρ	Mixing parameter	0.01,0.05,0.1,0.15,0.2
minc	Minimum community sizes	1
maxc	Maximum community sizes	20-100-200

chose different mixing parameters ρ (0.01, 0.05, 0.1, 0.15, 0.2) to allow different levels of modularity. A lower mixing parameter corresponds to higher modularity. We repeated the simulation 10 times. These networks constitute the ground truth networks. From each of these networks, we generated three perturbed ones varying the percentages of edge perturbation as 0%, 5%, 10%, and 15%. These three perturbed networks represent those we want to integrate to recover the ground truth. Finally, we used a Beta distribution, whose values range between [0,1], to generate the network weights. We adopted the formulation of the Beta distribution characterized by the two parameters $\alpha > 0$ and $\beta > 0$ from which the mean μ is given by $\frac{\alpha}{\alpha+\beta}$ and the variance σ^2 is $\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$. In particular, following Ji et al. (2005), we first sampled the weights from two different Beta distributions, i.e., from a $B_{(8,2)}$ for assigning substantial weights to the edges present in the network and from a $B_{(2,8)}$ for smaller weights to all the other edges, leading to a beta mixture. After that, we fitted a beta mixture and set all the weights with a posterior probability less than 0.01 to zero to obtain sparse networks.

In this way, the simulated networks differ depending on the different perturbation levels, the weights distributions, and the cutting point of the weights. For these reasons, the differences with the ground truth networks are greater than the original percentage of perturbations.

Computational aspects For the generation of the network, we used the C++ *Package 1* of the LFR benchmark graphs (Lancichinetti et al. 2008) from the website https://www.santofortunato.net/resources. For the perturbation strategy, we used the `rewire` function of `igraph` package with the `keeping_degseq` option that preserves the degree distribution of the network. For the random generation of the weights from the Beta distribution, we used the function `rbeta`. To fit the mixtures of beta regression models via maximum likelihood with the EM algorithm, we applied the function `betamix` from the package `betareg`.

4.1 Performance evaluation

To evaluate the performance of our method, we compared our `Consensus Network` and the classical mean approach, defined as the Baseline method from here on, with the network representing the ground truth. To ensure comparability between the methods, we applied a threshold of 0.5 for the edge weights in both cases. To this purpose, we measured the true positive edges (TP), the true negative edges (TN), the false positive edges (FP), and the false negative edges (FN). Furthermore, we also used the F-score and the Accuracy, where

$$F_1 = \frac{\mathrm{TP}}{\mathrm{TP} + \frac{1}{2}(\mathrm{FP} + \mathrm{FN})}, \tag{8}$$

and

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{9}$$

Figure 3 shows the F_1 score performances of our method in comparison to the Baseline method on the simulated networks with 100 nodes. The results corresponding

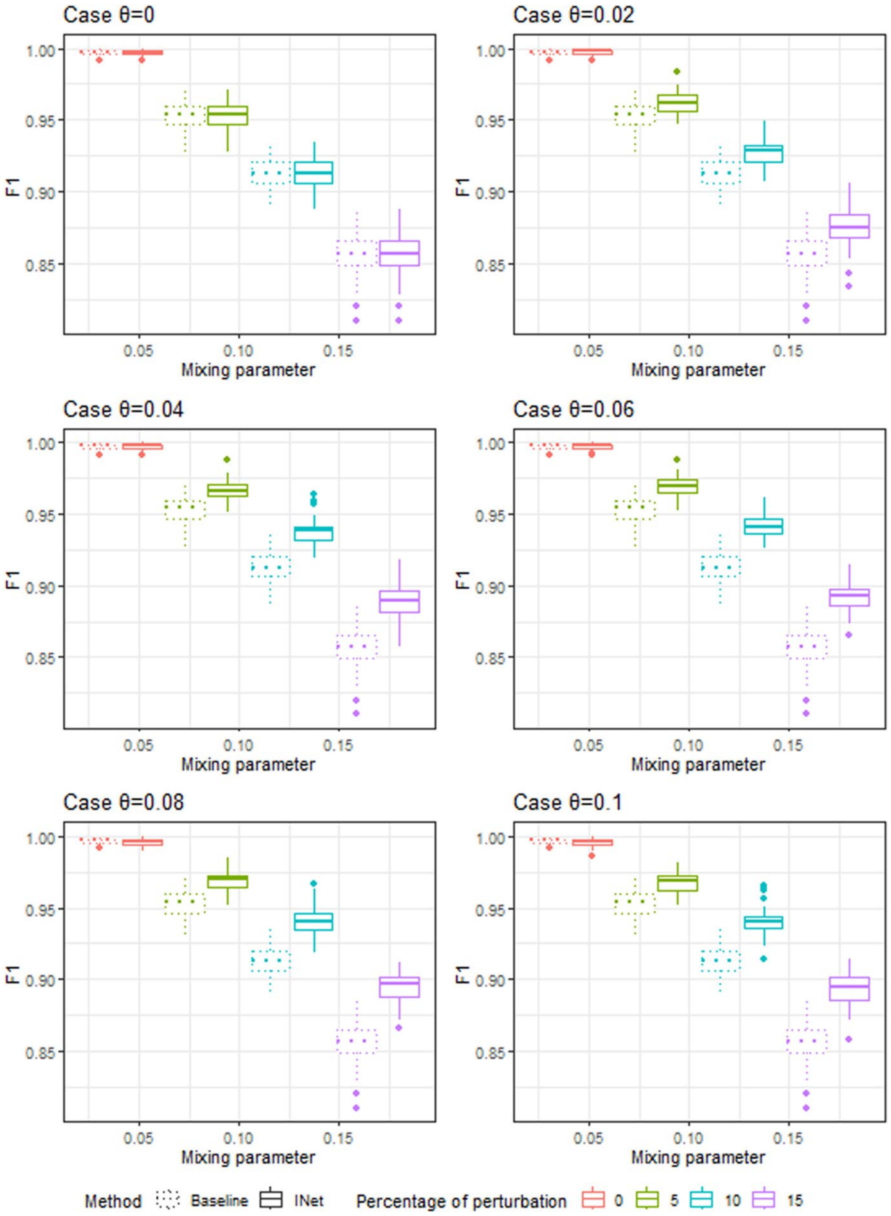


Fig. 3 The F-score is associated with the simulation networks containing 100 nodes. Each panel corresponds to a distinct choice of θ; in each panel, on the x-axis, we plot the mixing parameter (ρ) for both methods: `INet` and Baseline represented with dotted and continuous lines. The different box-plot colours are related to the different percentages of perturbation (color figure online)

to networks with 500 and 1000 nodes are in Supplementary Figs. 9 and 10 respectively. Comparing the box plots with no perturbation among the edges, we observe that our method behaves similarly to the Baseline, and both report excellent performance with an F_1 score close to 1. However, when increasing the perturbation, our method performs significantly better than the Baseline. As expected, the F_1 score decreases when differences among the networks increase; however, the loss in the F_1 score of our method is much less than that of the Baseline. The Accuracy performances are shown in Supplementary Figs. 16, 17 and 18. For the sake of completeness, we compared INet also with the union and intersection approach in Supplementary Figs. 19 and 20. In this case, we also show that INet performs better than the competitors.

Moreover, when evaluating the parameter θ (see Eq. (1)) in the different plots, we can observe that our approach and the Baseline show the same performances with $\theta = 0$; this is an almost obvious result as the differences between our method and the Baseline are primarily due to the Ego network component in the weight update. When $\theta = 0$, this term is not considered, and the weight in Eq. (2) is not the simple mean of the weights in the different layers, as in the Baseline method. When θ increases, our method usually improves. However, the optimal choice of θ requires attention. For example, when the number of nodes increases, a smaller value of $\theta > 0$ is preferable since with a larger value of θ, some performances are similar or lower compared to the Baseline. More details are discussed in Sect. 4.2.

To assess that the F-score performance of our method is statistically better than the Baseline's, we applied the Wilcoxon signed rank test with paired samples (Wilcoxon 1992) and one-side alternative. The results of the Wilcoxon signed rank test are shown in Fig. 4 for the simulated networks of 100 nodes, and in Supplementary Figs. 13 and 14 for respectively 500 and 1000 node networks. The tests are in agreement with the results seen above. All the tests are statistically significant as they are all above the red line, which represents a p-value equal to 0.05 except for $\theta = 0$ and 0 perturbation, where the performances are similar to the baseline method, and for $\theta = 0.1$ only for the 500 and 1000 nodes networks.

Note that our simulation study aimed to investigate the differences between the ground truth network and the Consensus Network regarding structure topology, focusing on the presence and absence of edges. However, other simulations could also be designed to assess the strength of the weights and evaluate the relative root mean square error as suggested in Jin and Xu (2024, 2024).

4.2 Impact of the Ego network component on the performance

From Eq. (1), the parameter θ can be chosen for tuning the relevance of the network structure in updating the weights. Increasing θ gives more relevance to the Ego network component and allows our method to perform better. As noticed before, if $\theta = 0$, the performances of our method are similar to the Baseline method as it will not add the network component in the construction of integrated networks. While increasing the θ, we add information on the neighbours and improve the performance. Figure 5 shows the mean of the F-score that varies with θ (x-axis) for 100

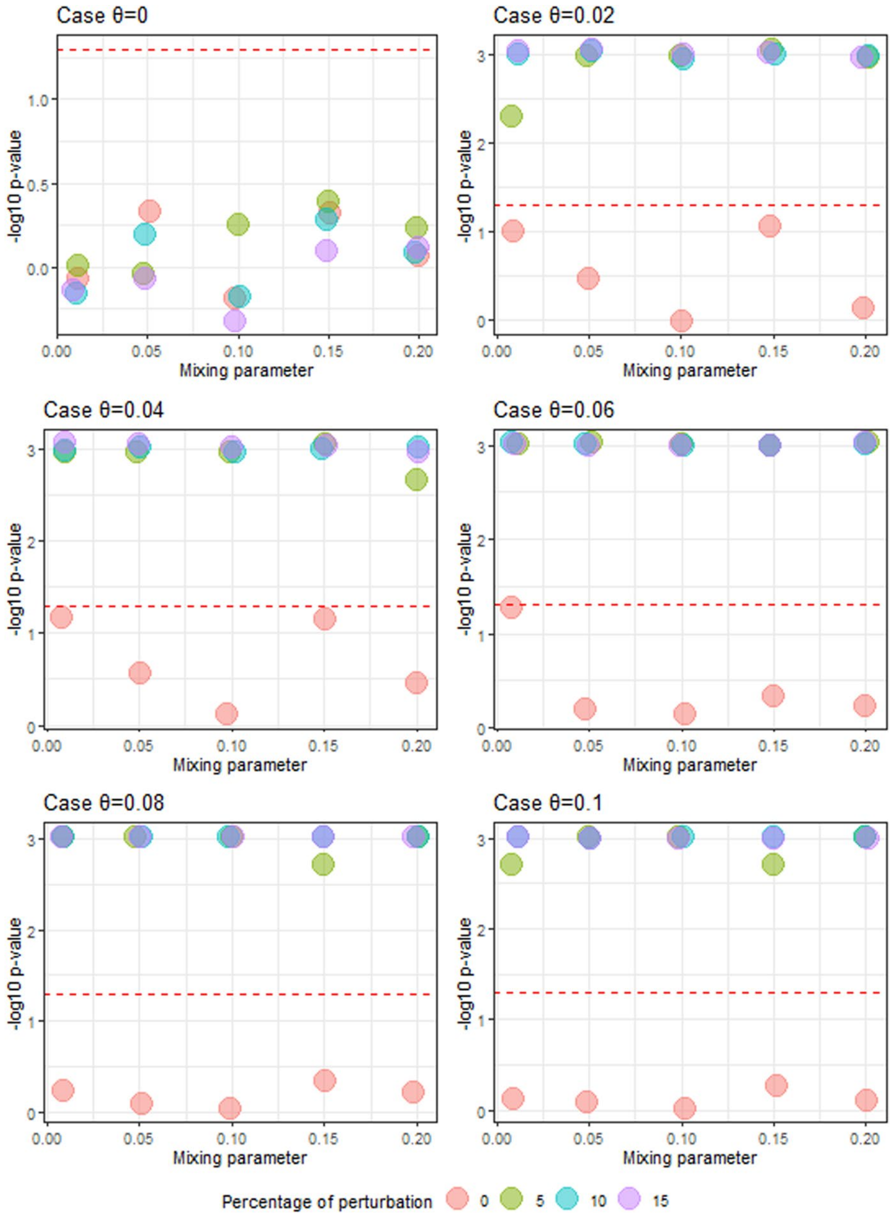


Fig. 4 One-side paired Wilcoxon Test testing the improved performance of `INet` compared with the Baseline approach. The results correspond to the simulation with networks of 100 nodes. Each panel corresponds to a distinct choice of θ; in each panel, on the x-axis, we plot the mixing parameter (ρ), and on the y-axis, the p-values for the test. The red line corresponds to a p-value = 0.05. p-values above the red line correspond to statistically significant tests (color figure online)

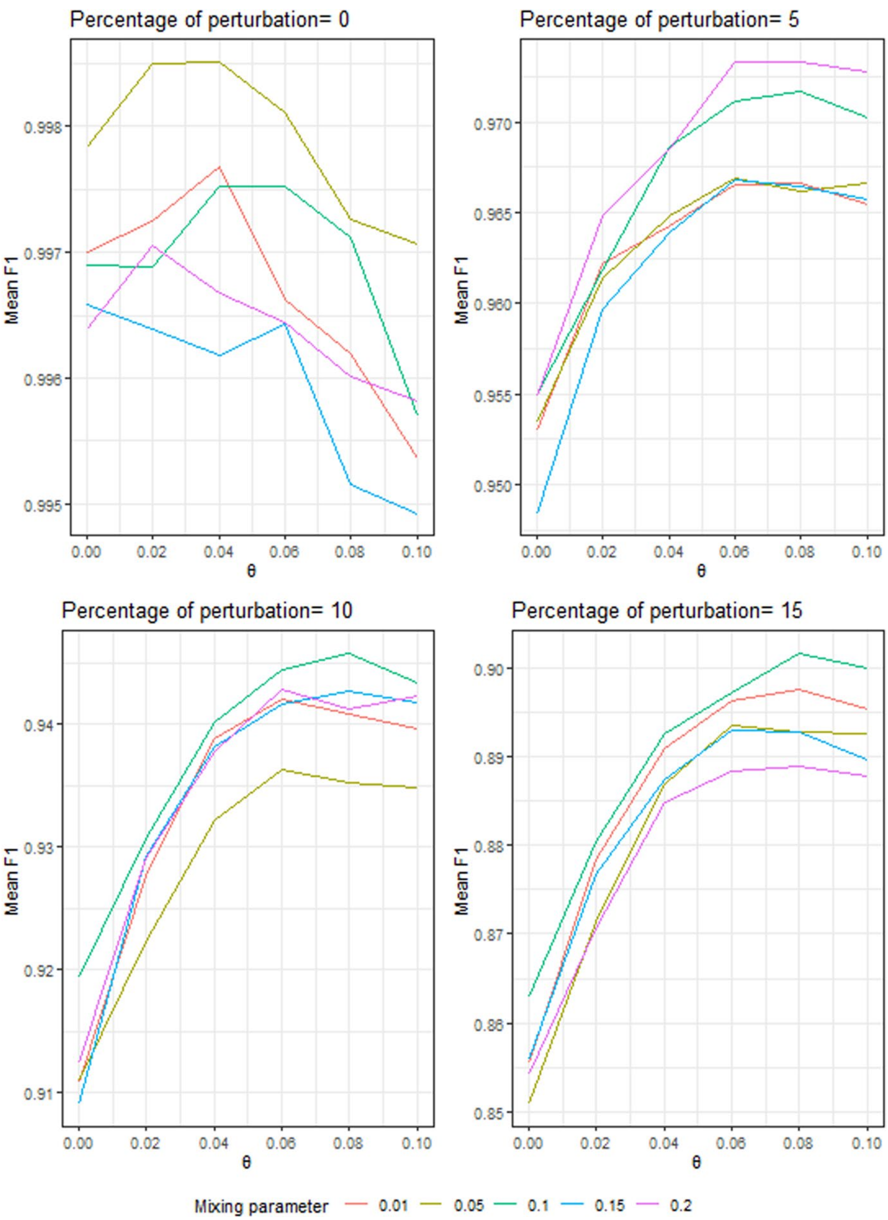


Fig. 5 F-score obtained by INet for different parameter values θ for the simulation with networks of 100 nodes. Each panel represents different percentages of perturbation. Each line colour corresponds to a different mixing parameter ρ (color figure online)

node networks; in Supplementary 11 and 12, we also report the results for the 500 and 1000 nodes networks, respectively. Each figure contains four panels, one for each percentage of perturbation, with five curves, one for each mixing parameter.

We can observe an increase in the performance with the increase of θ until a certain θ^* represents the optimal choice. However, the choice of θ^* varies with the network size. The 100-node networks have a maximum F-score between $\theta = 0.06$ and 0.08, while for the 500 and 1000-node networks, the highest performances are around $\theta = 0.04$.

5 Applications of `INet` to real data

We applied our method to two case studies. The first example concerns integrating multi-omics data to identify the subtypes of glioblastoma multiforme, an aggressive adult brain tumour, and the second is a social analysis of the Italian political debate from social media. For the first data analysis, we downloaded the preprocessed multi-omics data and patient survival data associated with the example *GMB.zip* from the website (http://compbio.cs.toronto.edu/SNF/SNF/Software.html), which stores different TCGA (The Cancer Genome Atlas) datasets used in the paper (Wang et al. 2014). We used Twitter (nowadays named *X*) data for the sociological data, downloading them via Twitter API.

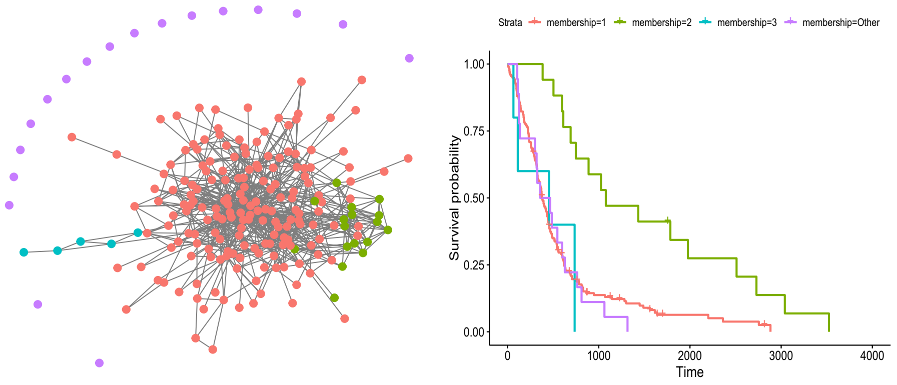
5.1 Multi-omics data analysis

This example consists of three different omics types: DNA methylation, mRNA expression, and miRNA expression from the same 215 patients affected with glioblastoma multiforme (GBM). The DNA methylation matrix contains 1305 probes, the mRNA gene expression contains 12,342 genes, and miRNA expression includes 534 microRNA. The preliminary data preprocessing is described in Wang et al. (2014). Our analysis aims to show that integrating the different omics types allows for better identifying GBM subtypes. The dataset also contains survival data. Similarly to Wang et al. (2014), the survival information is not used for the integration but can be used for validating the identification of the cancer subtypes.

As the first step, we constructed a patients' network for each data type, applying the `constructionGraph` function (See Table 3) of our package, which computes a person correlation, setting the parameter percentile to 0.9 preserving, in this way, we retain only the 10% of the most relevant connections in each network. Therefore, we obtained three patient similarity networks, each consisting of 215 nodes representing patients and 2204 edges connecting them. Then, we used `INet` to construct the `Consensus Network` that integrates all three different data types. The resulting patient `Consensus Network`, shown in Fig. 6a, consisted of 215 patients; out of them, 197 were connected by 847 edges.

To validate the goodness of the `Consensus Network`, we first identified GBM sub-types, and then we showed that each subtype is associated with a statistically significant difference in their prognosis.

To this purpose, given the `Consensus Network`, we first applied the methodology proposed in the `robin` package (Policastro et al. 2021) to determine the appropriate community detection algorithm. We selected the Label propagation



(a) Patient `Consensus Network`

(b) Survival analysis

Fig. 6 **a** The Consensus Network of GBM patients obtained using INet with three different multi-omics data types. The colours of the dots correspond to the cluster. **b** The Kaplan-Meier curves of the groups of patients identified by the community detection algorithm on the Consensus Network. The colours of the lines correspond to the cluster (color figure online)

algorithm as the more robust choice in this case (see Supplementary Fig. 15). Then, we used the Label propagation algorithm on the Consensus Network to cluster the patients corresponding to potential GBM sub-types. We detected three different groups of patients and a few isolated nodes that we labelled as belonging to the fourth cluster.

Finally, we estimated the survival curves of the patients of these four groups using the Kaplan-Meier method (Fig. 6b), and we performed a long-rank test to evaluate the significance of the differences in survival profiles between sub-types. We found a statistically significant difference in survival with a p-value 0.0006 between the groups; see Table 2. Overall, the consensus network resulting by INet gives a much clearer picture of clustering in our patients with GBM, identifying three sub-types with different survival. In particular, the second group of patients identified by INet shows a significantly better survival prognosis than the others. The fourth group identified patients who appeared different from the others (i.e., disconnected in the Consensus Network), with a poor prognosis.

Table 2 Survival

	N	Observed	Expected	(O-E)^2/E	(O-E)^2/V
Membership = 1	175	160	143.43	1.915	7.310
Membership = 2	17	16	37.17	12.057	16.831
Membership = 3	5	5	3.85	0.346	0.356
Membership = other	18	18	14.56	0.815	0.889

Chisq= 17.2 on 3 degrees of freedom, $p = $ 6e-04

5.2 Sociological data analysis

We downloaded the data from Twitter. In particular, we extracted the tweets until 2023-01-31, of the 51 most followed Italian politicians with a Twitter account. For each tweet, we considered the hashtags, whether a politician was mentioned or retweeted from another politician. We classify the hashtags of interest according to three different topics: the coronavirus, the Russia-Ukraine war, and Italian internal politics. As hashtags related to coronavirus, we took #Covid, #Coronavirus, #COVID19, #greenpass, #vaccino, #novax, and others. For the one related to the war between Russia and Ukraine, we took hashtags as: #Ucraina, #Russia, #Putin, #NATO, #UkraineRussiaWar, #Kiev etc... . For the last topic, Italian internal politics, some of the hashtags are: #PNRR, #Lavoro, #carobollette, #caroEnergia, #DecretoPNRR #lavoro. We used such information to construct three topic-specific networks with the politicians as nodes. For each of these topics, we constructed a weighted network where the nodes were the politicians, and two politicians were linked if they were present in the same tweet, i.e., if they were mentioned or retweeted. To construct a weight for the edge that varies between [0,1], we defined the weight as the number of tweets of that topic where the pair of politicians are present divided by the maximum weight in that topic.

With this procedure the Coronavirus network was made of 34 nodes (politicians) and 40 edges, the Russia-Ukraina network of 29 politicians and 37 edges and the Internal politics network of 41 politicians and 77 edges. We applied `INet` on this multilayer network to construct a consensus network which gave us the overall idea of the political relationship (Fig. 7).

From the `Consensus Network` in Fig. (7), we can see that the politicians of the Democratic Party (PD) seem to tweet always together, while in the other parties, it seems that only the leader tweets with another politician of the same party, for example, for the Fratelli d'Italia party (FdI) we can see that Giorgia Meloni (the leader at this time) tweets with Daniela Santanchè and for the Italia Viva party Matteo Renzi tweets always with Maria Elena Boschi. Furthermore, we can see that the Movimento 5 Stelle party in the `Consensus Network`, which takes into account all the different topics together, giving a more global view of the entire system, is not present probably because they retweet or mention other people, not politicians. In this field, a union of networks is often used. However, while the `Consensus Network` includes a subset of edges selected based on topology and the strength of weights in the layers, the union network contains all edges present in at least one layer, resulting in a noisier outcome. The union of the different layers is shown in Supplementary Fig. 21, where we can observe connections such as the one between Matteo Renzi and Alemanno, two politicians from opposing parties. This particular edge is weak and noisy and, ideally, should not appear in the integration, as it is present only in the Internal Politics Network, and the two nodes do not share any common neighbors. This behavior is a potential inconsistency our `Consensus Network` aims to face.

Moreover, we applied the `Case Specific Networks` algorithm to construct the three specific networks coronavirus, Russia-Ukraine war, and Italian internal politics, which are shown in Fig. 8.

From the Coronavirus `Specific Network` in Fig. 8a, it stands out that there is no community structure due to the political parties, but it shows the map of the politicians

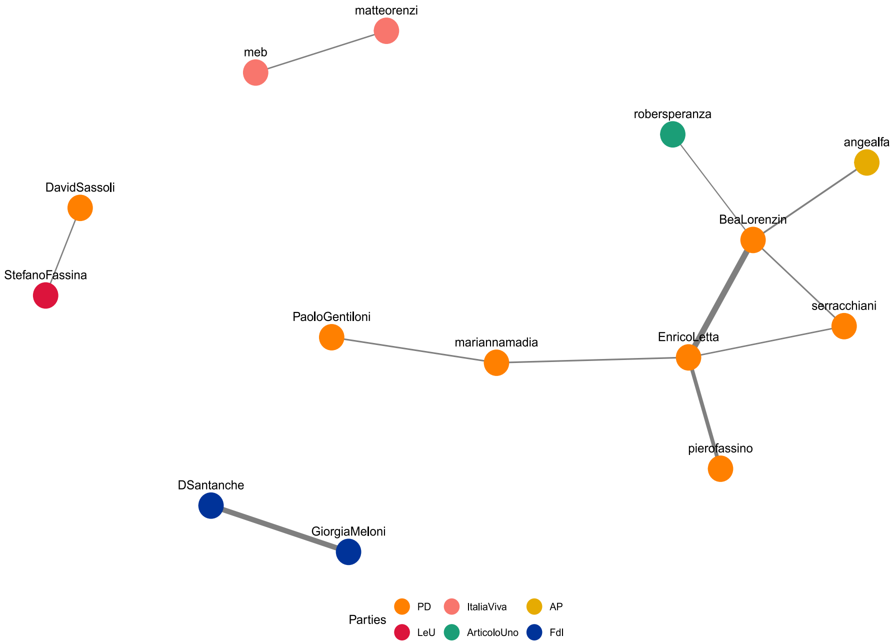
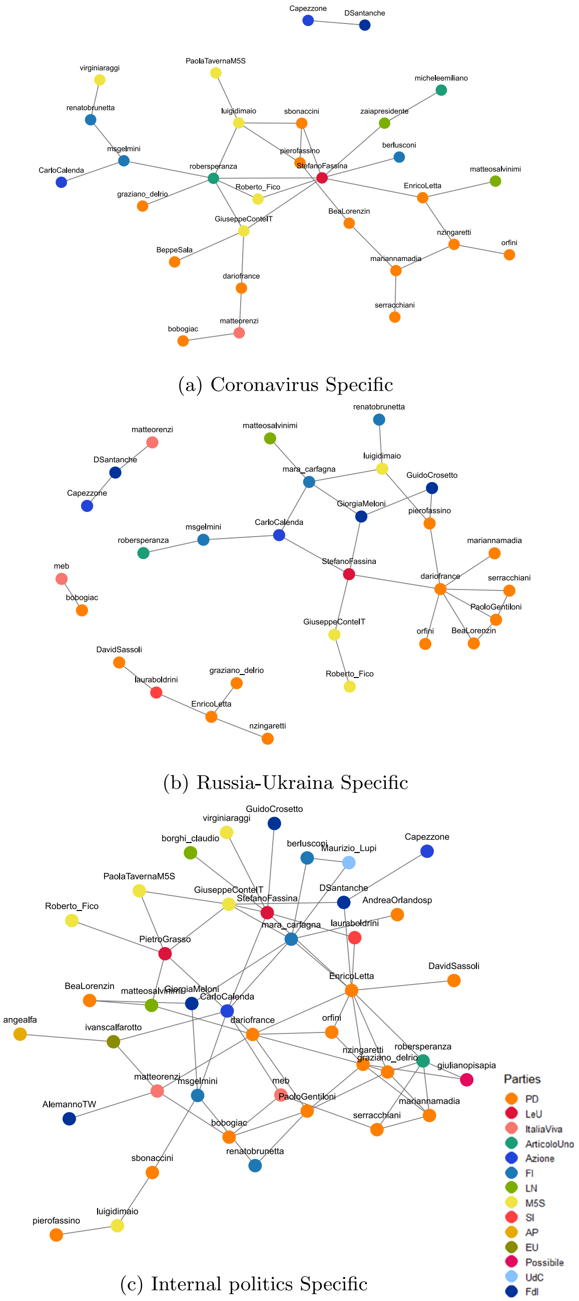


Fig. 7 Politician Consensus Network: the Consensus Network of politicians obtained using INet with the three different topic networks. The colours of the dots correspond to the political parties (color figure online)

involved firsthand in the political decision of the COVID period. We can see Conte, who was President of the Council of Ministers in the COVID period, mostly tweets with Roberto Speranza, who was designated as Minister of Health by Conte at that time, with Giuseppe Sala, who was mayor of Milan (the first Italian city where the virus exploded) and Dario Franceschini Minister of Culture. Moreover, we can see that Meloni, today's President of the Council of Ministers, is not present, maybe because, at that time, she was not so inside the political scenario.

From the Russia-Ukraine war Specific Network in Fig. 8b we can see that it is not a topic covered by a specific party but involves quite all the Italian political parties as they are all present in the network. We can also highlight that PD politicians, also in this case, interact more between them than with the other politicians and that there is a connection between Giorgia Meloni (President of the Council of Ministers) and Guido Crosetto as he is the Defence Minister.

The last Specific Network is the one related to Internal politics (in Fig. 8c), from this network, we can see that all the political parties are involved in this theme. All the politicians in the network interact with other politicians, not considering the political party to which they belong. However, there is still a higher connection between the politicians of the PD party. To summarize, we can say that all four political networks, the Consensus one and the Specific ones, gave us different information and accurately described the Italian political environment with the presence of different parties in different themes, and without a preponderant party in any of them.



(a) Coronavirus Specific

(b) Russia-Ukraina Specific

(c) Internal politics Specific

Fig. 8 `Case Specific Networks` obtained using `specificNet` function (see Table 3). The colours of the dots correspond to the political parties. **a** The coronavirus `Specific Network`. **b** The Russia-Ukraina `Specific Network`. **c** The internal politics `Specific Network` (color figure online)

To conclude, the INet method applied to datasets from two different contexts generated useful insights into the complex system they were describing.

6 Conclusion

In this paper, we presented INet, a novel method to integrate networks based on different data types, showing that such an approach allows the identification of structures not visible with the analysis of a single data set. Since INet works on networks instead of on the data domain, it is capable of handling both *n-* and *p-Integration*, and it captures the structure underneath all the different datasets to understand better the complex system analyzed in a multilayer weighted network. The INet Algorithm constructs a Consensus Network, which captures the complete spectrum of underlying datasets and Case Specific Networks, to highlight the information present only in that network and not in all the others. We assessed our method's performance and advantages using simulated and real data. For the real data, we used two examples: one in the biomedical field, integrating multi-omics data, and the other in the sociological field, integrating social media data. We limited our attention to the network's topology for the simulated data; therefore, we used performance measures to capture the network structure. However, in the future, we could also assess quantitative measures such as the strength of the estimated weights, for example, with the relative root mean square error. In the next works, we plan to expand INet capabilities with more analytical functions and consider the nodes' attributes in the different layers, which is a more complex but promising goal.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00180-024-01536-8.

Acknowledgements This work was partially supported by the project Piattaforma Tecnologica ADVISE funded by Regione Campania, the CNR Short Term Mobility 2020, the PRIN 2022 PNRR P2022BLN38 project funded by European Union—Next Generation EU, the PNC 0000001 D34Health project funded by European Union—Next Generation EU (CUP B53C22006100001). Part of the research activities were carried out at DiSTABiF—University of Campania Luigi Vanvitelli where V.P. was PhD student. C.A., A.C. and V.P. are members of the Gruppo Nazionale Calcolo Scientifico-Istituto Nazionale di Alta Matematica (GNCS-INdAM). The results shown for the GBM dataset are based upon data generated by the TCGA Research Network: https://www.cancer.gov/tcga.

Funding Open access funding provided by Università degli Studi di Napoli Federico II within the CRUI-CARE Agreement.

Declarations

Computational details We uploaded the code of the method in github at https://github.com/ValeriaPolicastro/INet-Tool. The results in this paper were obtained using R 4.1.3 with the packages all packages used are available from the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/.

References

Argelaguet R, Cuomo AS, Stegle O, Marioni JC (5 2021) Computational principles and challenges in single-cell data integration. Nat Biotechnol 39:1202–1215. https://doi.org/10.1038/s41587-021-00895-7, https://www.nature.com/articles/s41587-021-00895-7

Barabási AL (2016) Network Science. Cambridge University Press, Cambridge

Biswas A, Biswas B (2015) Investigating community structure in perspective of ego network. Expert Syst Appl 42:6913–6934. https://doi.org/10.1016/J.ESWA.2015.05.009

Deza E, Deza MM, Deza MM, Deza E (2009) Encyclopedia of Distances. Springer, Berlin

Giordano G, Ragozini G, Vitale MP (2019) Analyzing multiplex networks using factorial methods. Soc Netw 59:154–170. https://doi.org/10.1016/j.socnet.2019.07.005

Interdonato R, Magnani M, Perna D, Tagarelli A, Vega D (2020) Multilayer network simplification: Approaches, models and methods. Comput Sci Rev 36:100246. https://doi.org/10.1016/j.cosrev.2020.100246ï, https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/

Jaccard P (1901) Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bull Soc Vaudoise Sci Nat 37:547–579

Ji Y, Wu C, Liu P, Wang J, Coombes KR (2005) Applications of beta-mixture models in bioinformatics. Bioinformatics 21(9):2118–2122. https://doi.org/10.1093/BIOINFORMATICS/BTI318, https://academic.oup.com/bioinformatics/article/21/9/2118/409220

Jin B, Xu X (2024) Price forecasting through neural networks for crude oil, heating oil, and natural gas. Measur: Energy 1:100001. https://doi.org/10.1016/J.MEAENE.2024.100001

Jin B, Xu X (2024) Forecasting wholesale prices of yellow corn through the Gaussian process regression. Neural Computing and Applications 36. https://doi.org/10.1007/s00521-024-09531-2

Korhonen O, Zanin M, Papo D (2021) Principles and open questions in functional brain network reconstruction. Human Brain Mapp 42(11):3680–3711. https://doi.org/10.1002/hbm.25462

Lancichinetti A, Fortunato S, Radicchi F (2008) Benchmark graphs for testing community detection algorithms. Phys Rev E. https://doi.org/10.1103/physreve.78.046110

Malod-Dognin N, Petschnigg J, Windels SF, Povh J, Hemingway H, Ketteler R, Pržulj N (2019) Towards a data-integrated cell. Nat Commun. https://doi.org/10.1038/s41467-019-08797-8

Núñez-Carpintero I, Cirillo D, Valencia A A (2019) multilayer network approach to elucidate severity in Congenital Myasthenic Syndromes. Tech. rep., https://github.com/imgag/ClinCNV

Policastro V, Righelli D, Carissimo A, Cutillo L, de Feis I (2021) ROBustness in network (robin): an R Package for comparison and validation of communities. R Journal 13(1):292–309. https://doi.org/10.32614/RJ-2021-040

Pržulj N (2019) Analyzing Network Data in Biology and Medicine. Medical, and Computational Scientists, An Interdisciplinary Textbook for Biological

Rohart F, Gautier B, Singh A, Lê Cao KA (2017) mixOmics: An R package for ‘omics feature selection and multiple data integration. PLOS Comput Biol 13(11):e1005752. https://doi.org/10.1371/journal.pcbi.1005752

Rondinelli R, Marmani S, Ficcadenti V (2020) Biblical names’ relationships in the Gospel of Matthew, Mark, Luke, John and Acts of Apostles, pp 1–12. http://arxiv.org/abs/2012.04753

Simmel G (1908) Sociology: Investigations on the Forms of Sociation. Soziologie

Vulliard L, Menche J (2021) Complex networks in health and disease. In: Wolkenhauer O (ed) Systems Medicine. Academic Press, Oxford, pp 26–33. https://doi.org/10.1016/B978-0-12-801238-3.11640-X, https://www.sciencedirect.com/science/article/pii/B978012801238311640X

Váša F, Bullmore ET, Patel AX (2018) Probabilistic thresholding of functional connectomes: Application to schizophrenia. NeuroImage 172:326–340. https://doi.org/10.1016/j.neuroimage.2017.12.043, https://www.sciencedirect.com/science/article/pii/S1053811917310649

Wang B, Mezlini AM, Demir F, Fiume M (2014) Similarity network fusion for aggregating data types on a genomic scale. Nat Methods 11(3):333–337. https://doi.org/10.1038/nmeth.2810, https://pubmed.ncbi.nlm.nih.gov/24464287/

Wilcoxon F (1992) Individual Comparisons by Ranking Methods, pp 196–202. https://doi.org/10.1007/978-1-4612-4380-9_16, https://link.springer.com/chapter/10.1007/978-1-4612-4380-9_16

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.