



# OPEN A comprehensive framework for solution space exploration in community detection

Fabio Morea<sup>1</sup> & Domenico De Stefano<sup>2</sup>✉

Community detection algorithms are essential tools for understanding complex networks, yet their results often vary between runs and are affected by node input order and the presence of outliers, undermining reproducibility and interpretation. This paper addresses these issues by introducing a framework for systematic exploration of the solution space, obtained through repeated runs of a given algorithm with permuted node orders. A Bayesian model assesses convergence, estimates solution probabilities, and provides a defensible stopping rule that balances accuracy and computational cost. Building on this process, we propose a taxonomy of solution spaces that offers clear diagnostics of partition reliability across algorithms and a shared vocabulary for interpretation. Applied to a real-world network, the approach shows that different algorithms produce various types of solution space, highlighting the importance of systematic exploration of the solutions before drawing scientific conclusions.

**Keywords** Network analysis, Community detection, Solution space, Outliers, Horizon project networks

Networks are widely used for representing complex systems where a set of nodes are connected by edges and are ubiquitous across diverse scientific fields<sup>1</sup>, including social sciences<sup>2</sup>, biology<sup>3</sup>, science of science<sup>4,5</sup>, neuroscience<sup>6,7</sup>, and innovation studies<sup>8,9</sup>.

Networks often exhibit the property of having a community structure, that is, their nodes are organised into groups, called communities, clusters, or modules<sup>10</sup>. Therefore, nodes can be partitioned into either disjoint or overlapping sets of vertices such that the number of edges within a set exceeds the number of edges between any two sets by some reasonable amount<sup>11</sup>. Discovering such tightly knit groups of nodes is crucial in many applications<sup>12</sup>. For example, in innovation diffusion networks, identifying significant and stable communities helps reveal knowledge flows between organisations or regions, shedding light on the underlying aggregation mechanism at play<sup>13</sup>. Community detection methods are often used for such task, and the key requirements are that the results must be independent of contingent factors (such as software implementation or ordering of the input data) and tested for validity.

Community detection typically assumes that a network can be partitioned into distinct communities and that a single optimal solution can be identified. It is clear that different algorithms often produce varying results due to methodological differences. However, under specific conditions, an algorithm may yield diverse outcomes depending on their inherent stochastic approach.

The present paper explores the importance of examining the solution space in the most commonly used community detection algorithms, highlighting its role in achieving reliable results when dealing with real-world networks. In this setting there are three issues that are still not extensively explored in the complex networks literature, namely: multiplicity of solutions, outliers handling, and input ordering bias.

In networks with fuzzy or complex community structures, significant variability in results can occur both across different algorithms and within repeated runs of the same algorithm.

Outliers, nodes that do not clearly belong to any community, are a critical yet under-explored aspect of community detection. Many algorithms either disregard outliers or assign them to existing communities, potentially distorting the detected structure. Proper identification and treatment of outliers are essential for generating accurate partitions and understanding their role in network dynamics.

Input ordering bias, where the sequence of nodes and edges affects algorithmic output, poses a significant challenge in community detection. Ideally, results should depend solely on the network's intrinsic structure, not on an arbitrary processing order. However, several popular community detection algorithms are sensitive

<sup>1</sup>Area Science Park, Padriciano 99, Trieste, Italy. <sup>2</sup>University of Trieste, Piazzale Europa 1, Trieste, Italy. ✉email: ddestefano@units.it

to this bias, leading to inconsistencies and affecting the reliability of the results. Mitigating this issue is essential to ensure that detected communities accurately reflect the network's inherent topology rather than artifacts of computation.

To address these challenges, this study proposes a comprehensive methodology for exploring the solution space of community detection algorithms, coupled with a taxonomy to classify different types of solution spaces. This paradigm allows for a deeper understanding of the variability, biases, and anomalies present in network partitions, paving the way for more robust and meaningful analyses.

The remainder of this paper is structured as follows: Section 2 discusses related works addressing the quality issues in community detection results; Section 3 presents the methodological framework for solution space exploration, including the proposed taxonomy and outlier detection approach; Section 4 illustrates the methodology through synthetic and real-world network experiments; Section 5 concludes with a discussion of findings and future research directions.

## Related works

Several methods have been proposed to detect meaningful community structures based on the density of their internal connections<sup>14–16</sup>. This principle is quite general although other attachment and aggregation mechanisms are possible in social networks<sup>17</sup>. The main strategies for identifying an optimal partition include the detection of actors or edges with high centrality<sup>18</sup>, optimisation-based algorithms<sup>19</sup>, genetic algorithms<sup>20,21</sup>, statistical inference using stochastic block models<sup>22</sup>. Furthermore, a new class of community detection methods has emerged that exploits node semantics or node attributes in addition to network topology. According to the taxonomy proposed by<sup>23</sup>, these include graphical model-based community detection, deep learning-based community detection, as well as node embeddings<sup>24</sup>.

Although many of these methods focus on partitioning networks into non-overlapping communities, there is a diverse range of variants, including hierarchical clustering<sup>25,26</sup>, which captures structures at different scales, overlapping communities<sup>27–29</sup> and mixed-membership communities<sup>30</sup>, where a node can belong to more than one community, as well as a combination of overlapping and non-overlapping communities<sup>31</sup>.

However, probably due to their ability to produce easily interpretable results, optimisation methods and modularity based approaches that generate non-overlapping partitions are still widely used.

Despite their importance in practical settings, few studies focused on the issues discussed in the present paper or on the exploration of the solution space determined by the stochastic nature of such approaches.

Solution quality issues have been discussed in terms of the degeneracy of modularity maximisation algorithms<sup>32</sup> and resolution limit<sup>33</sup>. Other papers used topological information to measure the quality of community structure identified in large scale networks<sup>34</sup>. More recent works have also delved into the multiplicity of solutions arising from community detection methods, mapping out how solutions can vary across different algorithm runs and parameter settings<sup>35,36</sup>. A number of contributions, focused on exploiting such characteristic, proposed the use of consensus clustering based approaches<sup>37,38</sup>.

An additional focus is to examine the behaviour of community detection algorithms when confronted with outlier nodes, which are nodes that display significantly different behaviour compared to the rest of the network<sup>39</sup>. Previous studies have focused on anomaly and outlier detection<sup>40</sup>, especially in the context of online social networks<sup>41,42</sup>. In our framework, we explicitly define outliers as nodes whose community membership changes frequently across repeated runs of community detection algorithms. However, the way such cases are handled in many widely used community detection algorithms remains under explored.

Input ordering bias has been discussed in the literature, notably by<sup>32,43,44</sup> focusing mainly on modularity-based methods.

In this paper, we aim to generalise the results on the above discussed issues to any algorithm, and to devise a procedure that mitigates particularly the input-ordering bias, while improving the stability and reliability of results.

## Methodology

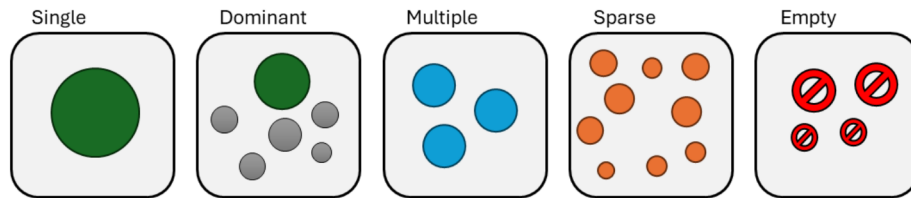
Let  $G = (V, E)$  be a graph with  $n_v = |V|$  vertices and  $n_e = |E|$  edges (for a list of key symbols and notations used throughout the manuscript, see the Supplementary Table S1). A community  $C$  is a subnetwork of  $G$  where nodes are more densely connected to each other than to the rest of  $G$ . A partition  $P$  is a set of  $k$  disjoint subnetworks  $C_1, \dots, C_k$  whose union is equal to  $G$ . A community detection algorithm  $\mathcal{A}(G, \rho) \rightarrow P$  is a function that takes as input a graph  $G$  and one or more parameters  $\rho$ , and returns a partition  $P$ . Ideally, each community detection algorithm discussed in Section 2 should yield a single valid partition when applied with a given set of parameters. However, in practice, for large and dense networks,  $\mathcal{A}$  may produce different partitions,  $P_i \neq P_j$  in repeated runs.

### Solution space definition and characterisation

The solution space  $\mathbb{S} = \{P_1, P_2, \dots, P_{n_s}\}$  is the set of all  $n_s$  unique partitions that  $\mathcal{A}$  produces in  $t$  trials or runs.

Our research introduces the notion that  $\mathbb{S}$  can be characterised through an iterative exploration process. In this process,  $\mathcal{A}$  is applied repeatedly under random permutations of the graph's input ordering—permutations that preserve the topology and information content—and the resulting partitions are modelled probabilistically. We also propose a taxonomy to classify  $\mathbb{S}$  into different categories (depicted in Fig. 1), based on the number of unique solutions, their probabilities and the convergence of the process.

The exploration process, detailed in Algorithm 1, consists of repeated trials guided by the convergence of a Bayesian model  $\mathbb{M}$ , based on the Dirichlet-Multinomial distribution.



**Fig. 1.** Taxonomy of solution spaces generated by community detection algorithms. The framework classifies partitions into five categories based on posterior probabilities and convergence behaviour: Single (only one stable partition), Dominant (one prevailing solution with probabilistic separation from others), Multiple (several comparable solutions with no clear separation), Sparse (many low-probability partitions, often incoherent), and Empty (degenerate or non-valid partitions).

The model maintains both counts  $c = (c_1, \dots, c_k)$  tracking how often each solution  $P_i$  is observed and their corresponding probability estimates  $\hat{p} = (\hat{p}_1, \dots, \hat{p}_k)$ , that quantify the likelihood of each partition. The marginal Beta distributions provide credible intervals  $[\hat{p}_i^l, \hat{p}_i^u]$  that quantify the uncertainty in each probability estimate.

The discovery process consists of repeated trials, whose independence is ensured through algorithmic stochasticity and permutation of network inputs (Algorithm 1, step 4). However, model  $\mathbb{M}$  relies on the exchangeability of observations, a condition that is justified by the same mechanisms, allowing effective sampling from the posterior distribution.

At the beginning, the model is initialised with a single dimension and a weakly informative prior, to reflect the belief that at least one solution exists. At each trial, the algorithm returns a partition  $P_t$  that is compared to all previously observed solutions using Normalised Mutual Information (NMI), a measure of partition similarity that is invariant to community label permutations. If  $P_t$  matches an existing solution ( $\text{NMI}(P_t, P_j) = 1$  for some  $j$ ), the corresponding count is incremented. When  $P_t$  is novel ( $\text{NMI}(P_t, P_j) < 1$  for all existing  $P_j$ ), the model is expanded to track this new solution.

After  $t$  trials, the posterior distribution over solution probabilities is given by:

$$p | c \sim \text{Dirichlet}(\gamma_0 + c_1, \dots, \gamma_0 + c_k)$$

where  $c_i$  is the number of times solution  $P_i$  has been observed, and  $\gamma_0$  is the prior count assigned equally to all solutions.

The exploration process is terminated either at convergence or upon reaching the maximum number of trials  $t_{\max}$ .

Convergence occurs when the posterior renders the solution space statistically unambiguous under one of two conditions: (i) *Stabilisation*, when  $\max_i (p_i^u - p_i^l) \leq \delta$ , where  $\delta > 0$  is a tolerance threshold controlling the required precision of probability estimates; or (ii) *Separation*, when  $\exists i^* : p_{i^*}^l > \max_{j \neq i^*} p_j^u$ , indicating that one solution is unambiguously dominant.

If neither condition is met within  $t_{\max}$  trials, the process stops in an intermediate regime where new solutions may continue to emerge sporadically. The thresholds  $\delta$  and  $t_{\max}$  balance computational cost against the benefit of further exploration. In this study, we set  $\delta = 0.1$  and  $t_{\max} = 50$ , a choice supported by empirical evidence, as most cases converge earlier and non-convergent ones are *Sparse* (see Fig. 1 and Section [Taxonomy](#) for the definition) where additional trials would not alter the solution space classification.

---

**Require:** Graph  $G$ , algorithm  $\mathcal{A}$ , parameters  $t_{\max}$  and  $\delta$

- 1: Initialise solution space  $\mathbb{S} \leftarrow \emptyset$
- 2: Initialise Bayesian model  $\mathbb{M}$  with a symmetric Dirichlet prior
- 3: **for**  $t = 1$  to  $t_{\max}$  **do**
- 4: Generate a random permutation  $G^*$  of  $G$  preserving topology and information content
- 5: Run  $A(G^*)$  to obtain partition  $P_t$
- 6: **if**  $P_t$  is identical (NMI = 1) to an existing solution in  $\mathbb{S}$  **then**
- 7: Increment the corresponding count in  $\mathbb{M}$
- 8: **else**
- 9: Extend  $\mathbb{M}$  with a new component
- 10: **end if**
- 11: Update posterior distribution in  $\mathbb{M}$
- 12: Compute point estimates  $\hat{p}_i$  for each solution
- 13: Compute credible intervals  $[p_i^l, p_i^u]$  for each solution
- 14: **if** (all intervals are narrower than  $\delta$ ) **or** (one interval lies entirely above all others) **then**
- 15: **break**
- 16: **end if**
- 17: **end for**
- 18: **return** solution space  $\mathbb{S}$  and model  $\mathbb{M}$

---

**Algorithm 1.** Solution space exploration.

---

### Taxonomy

The taxonomy proposed in Fig. 1 can be formally defined as follows:

**Single** The exploration process achieves stabilisation with only one valid partition ( $n_s = 1$ ). In this case, the process terminates at convergence, meeting the stabilisation criterion.

**Dominant** Multiple valid partitions exist ( $n_s > 1$ ), but one exhibits a decisive separation:  $i^* = \arg \max_i \hat{p}_i$  where  $p_{i^*}^l > \max_{j \neq i^*} p_j^u$ . The process terminates at convergence, satisfying the separation criterion.

**Multiple** Several partitions are observed ( $n_s > 1$ ) with comparable probabilities (credible intervals are overlapping). No solution achieves a decisive separation and  $\max_i (p_i^l) < 0.5$ .

**Sparse** A large number of solutions is observed ( $n_s \approx t$ ), each with low individual probability ( $p_i \approx 1/n_s$ ). Typically, the exploration process terminates at  $t_{\max}$  without achieving convergence.

**Empty** No meaningful partition is observed, either because the algorithm produces no valid solutions ( $n_s = 0$ ) or because all solutions violate basic community requirements, such as all nodes grouped in one community ( $k = 1$ ), all nodes as singletons ( $k = n_v$ ), or communities are internally disconnected.

The solution space taxonomy serves as a decision aid for downstream analysis, indicating if a single, representative partition can be credibly reported, and, conversely, if the evidence supports the absence of credible community structure.

In the Single case, the analysis is straightforward, and the unique partition can be regarded as representative. The Dominant case is more nuanced: a prevailing solution coexists with credible alternatives. Whether the most probable partition can be treated as representative is context-dependent and should be justified, for instance, by clear probabilistic separation from the alternatives. When the solution space is classified as Multiple, it is important to check for symmetry: if  $G$  admits  $k$  symmetry equivalent partitions, no single solution is expected to exceed  $\approx 1/k$  posterior probability. When a single representative partition must be reported in Dominant or Multiple solution spaces, a consensus procedure can be used to synthesise information across the competing solutions. A Sparse solution space can arise in two ways. (i) Solutions may be largely similar, differing only in minor details; they then form one or a few coherent clusters (with possible outliers), indicating an underlying structure that can be summarised via consensus. (ii) Partitions are mutually dissimilar, which may signal a lack of a credible community structure. This situation is easily observed when applying community detection algorithms to random graphs: each run returns a partition, yet the set of partitions is collectively incoherent.

In the following we discuss other possible issues affecting the solution space, that are input ordering bias and the identification of outlier nodes in the observed network.

### Input ordering bias

Although graphs are mathematically unordered objects, software implementations necessarily process ordered data structures such as nodes and edge lists, or adjacency matrices. As a result, many community detection algorithms become sensitive to otherwise irrelevant features such as node or edge indexing, order of iteration, or tie-breaking. We refer to this phenomenon as *input-ordering bias*.

Input-ordering bias can be detected by comparing solutions generated by  $\mathcal{A}(G)$  with those generated by  $\mathcal{A}(G^*)$  where  $G^*$  is an information preserving permutation of  $G$  obtained only by reordering nodes and edges. If  $\mathcal{A}$  is order invariant, the outcomes should be statistically indistinguishable.

If input ordering bias is not addressed, runs performed on a fixed node order may undersample the solution space: fewer distinct partitions are discovered, and the apparent dominance of some solutions may be overstated. Our exploration procedure implements systematic, information preserving permutations at every trial (Algorithm 1, Step 4). As already mentioned, this double randomisation—algorithmic stochasticity plus independent permutation—mitigates order dependent artifacts, expands coverage of the solution space, and justifies the exchangeability assumption underlying our statistical model.

### Outliers

Our approach to outlier detection is coherently grounded in the data and framework generated through solution space exploration, providing a straightforward method for identifying node outliers. The first step is to compute a *Pairwise Agreement Matrix*  $\Gamma$ , an  $n_v \times n_v$  symmetric matrix with entries:

$$\gamma_{uv} = \sum_{i=1}^{n_s} \hat{p}_i \cdot \chi_i(u, v), \quad \Gamma = [\gamma_{uv}]$$

where  $\chi_i(u, v) = 1$  if nodes  $u$  and  $v$  belong to the same community in partition  $P_i$ , and 0 otherwise;  $n_s$  denotes the number of unique partitions in the solution space  $\mathbb{S}$ , and  $\hat{p}_i$  is the estimated probability of partition  $P_i$ .

To assess whether a node  $v$  is an outlier, we examine its *pairwise agreement profile*, defined as the  $v$ -th row of  $\Gamma$ , which is a vector of length  $n_v$ . Its entries  $\gamma_{uv}$  quantify the frequency-weighted co-occurrence of  $v$  with each other node in the network. Entries equal to zero indicate pairs that never co-occur in the same community and can be excluded from the calculation.

The analysis of  $\gamma_{uv}$  offers a straightforward means of quantifying outlier tendencies without extensive algorithmic sophistication, making it particularly valuable for exploratory analysis and initial outlier screening. For example,  $\bar{\gamma}_v^{(+)}$  as mean of the non-zero entries of  $\gamma_{uv}$  is a single number associated to each node. Stable (i.e., non-outlier) nodes have  $\bar{\gamma}_v^{(+)}$  values close to 1.0 and consistently appear with the same neighbours, while low values of  $\bar{\gamma}_v^{(+)}$ , indicates nodes that are assigned to different communities within the solution space.

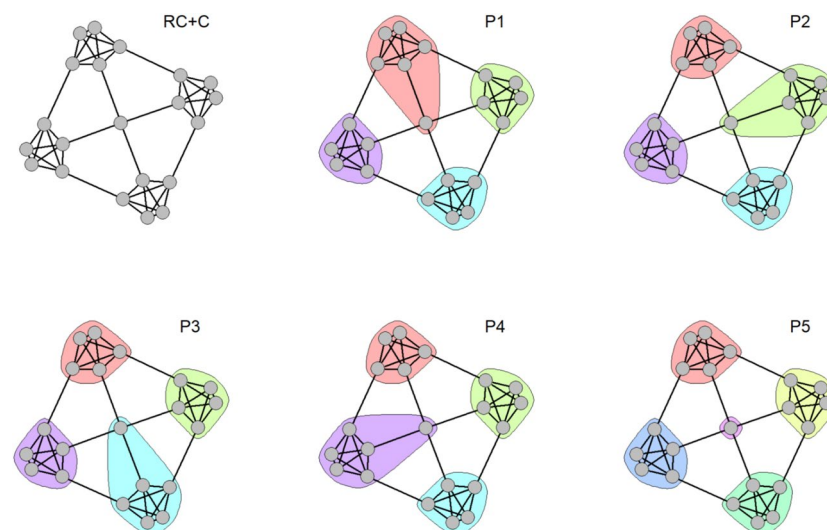
## Results and discussion

### Experimental design and test networks

To test the proposed methodology, a specific test network is used: a Ring of Cliques with a central outlier ( $RC+C$ ), shown in Fig. 2. A Ring of Cliques (RC) is a network composed of  $n_c$  cliques of size  $c_s$ , i.e., fully connected groups of nodes that naturally form distinct communities, arranged in a circular pattern. Each clique is isolated in terms of connectivity, making the community structure clear and straightforward. The  $RC+C$  builds on the basic RC structure by introducing an outlier: a “central” node, connected to each clique.

Although the community structure is visually obvious to human observers, algorithms operate without prior knowledge of the network structure. As a result, they may produce partitions that overlook the symmetry of the central node. Additionally, some algorithms can produce different solutions with each independent run and may exhibit an input ordering bias.

Most algorithms are likely to generate partitions such as  $P_1$ ,  $P_2$ ,  $P_3$ , or  $P_4$  shown in Fig. 2, where the outlier is incorporated into one of the cliques. However, these solutions are inherently flawed, as they unfairly favour



**Fig. 2.** Ring of cliques with central outlier ( $RC+C$ ) and resulting partitions. The  $RC+C$  network comprises four cliques and a central outlier node connected to all cliques. Partitions  $P_1$  through  $P_4$  assign the outlier to one clique, introducing imbalance. Partition  $P_5$  isolates the outlier as a single-node community, a configuration not all algorithms can produce.

one clique over the others. Alternatively, a solution like  $P_5$ , where the outlier forms a separate community, may appear more equitable. Nevertheless, not all algorithms are capable of producing such a partition, as it requires accepting the assumption that a single node can constitute a valid community.

The tests were carried out using six algorithms: Edge Betweenness (EB), Louvain (LV), Leiden (LD), Label Propagation (LP), Infomap (IM) and Walktrap (WT), selected for their widespread use in applications, ready availability in the `iGraph`<sup>45</sup> library for the R programming language<sup>46</sup>, and their representation of different approaches to community detection.

In the following, we briefly describe the main characteristics of such algorithms.

**EB** The Edge Betweenness (EB)<sup>18</sup> algorithm assumes that edges bridging different communities exhibit high betweenness centrality, defined as the frequency with which an edge lies on shortest paths between pairs of nodes. At each step, the edge with the highest score is removed and values are recalculated, and the process yields a hierarchical dendrogram of partitions. The final partition is typically chosen according to an external optimality criterion, though no unique rule is prescribed. EB is computationally demanding and does not scale to large networks.

**LV** The Louvain (LV) algorithm<sup>47</sup> is an agglomerative method that optimises modularity using a greedy approach. Initially, each node is assigned to a separate community; nodes are then iteratively moved to the community of one of their neighbours, maximising modularity, until no further improvement can be made. A resolution parameter ( $r$ ) controls the granularity of partitions:  $r > 1$  produces smaller and more numerous communities, while  $r < 1$  produces larger and fewer ones. LV is fast and scales efficiently to large graphs, but it has the drawback that it may produce communities that are internally disconnected. The algorithm is stochastic, as nodes are examined in random order, and it converges only to local rather than global maxima of modularity; consequently, each run can yield a different partition.

**LD** The Leiden (LD) algorithm<sup>48</sup> is an improvement of LV that addresses internally disconnected communities. After the initial greedy optimisation phase, LD adds a refinement step that splits any disconnected communities, ensuring all detected communities remain internally connected. Like LV, LD is stochastic due to random node examination, and can yield different partitions across runs.

**IM** The Infomap algorithm<sup>49,50</sup> detects communities by minimising the description length of a random walker's trajectory under the map equation. The optimisation employs stochastic search heuristics, so results vary across runs.

**WT** Walktrap<sup>51</sup> is a hierarchical clustering algorithm based on the assumption that nodes within a community are likely to be connected by shorter random walks. The algorithm begins with each node in its own community and iteratively merges pairs of communities based on a distance measure derived from random walks, producing a hierarchical dendrogram of partitions. A user-defined parameter  $s$  defines the length of the random walk to be performed, controlling the resulting community size.

**LP** Label Propagation (LP) relies on the notion of proximity or neighbourhood relationships, as discussed in<sup>52</sup>. Initially, each node is assigned a unique community label, then the nodes are iterated through in a random sequential order, and each node adopts the label that is most prevalent among its neighbours. This process continues until each node shares the label of the majority of its neighbours. The algorithm is stochastic, as both the sequence of node updates and the resolution of label ties are random, so different runs often yield different partitions.

## Results on artificial networks (Ring of Cliques)

We analyse the behaviour of the selected algorithms regarding their tendency to generate multiple solutions, their strategies for managing outliers, and their susceptibility to input ordering bias. The results are presented in Fig. 3 and summarised in Table 1.

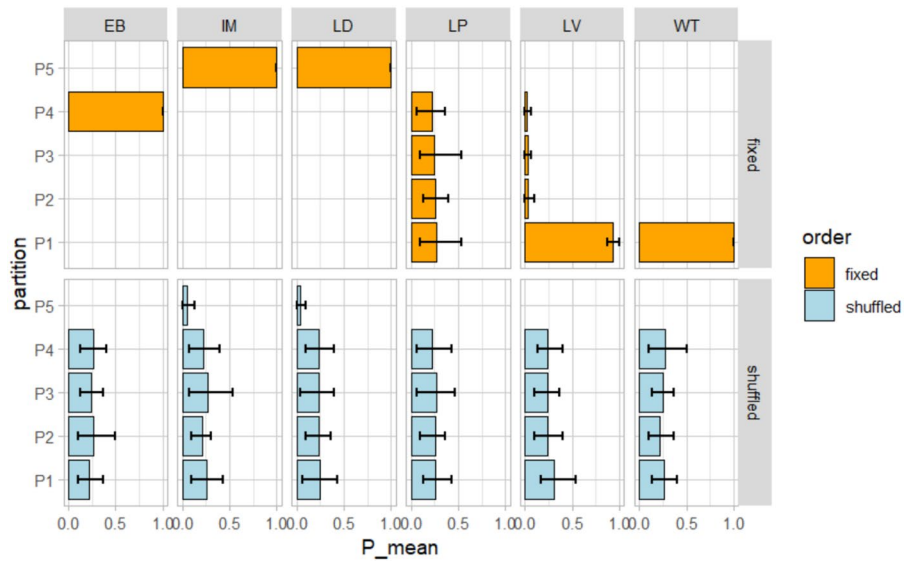
We observe that most algorithms exhibit a noticeable input-ordering bias, with the exception of LP. Specifically, when applied to  $RC+C$ , IM and LD assign the centre to a single-node community ( $P_5$ ). EB and WT always aggregate the center to one of the communities ( $P_4$  and  $P_1$ , respectively), while LV strongly favour the community  $P_1$ . However, when applied to a randomly permuted  $RC+C^*$ , each algorithm produces a less biased result.

As an illustrative example of outliers detection, in the  $RC+C$  network of Fig. 2, all nodes within the cliques always classified as stable communities with  $\bar{\gamma}_v^{(+)} = 1.0$  relative to each other, whereas the central outlier node has  $\bar{\gamma}_v^{(+)} = 0.25$  with any clique node, meaning it determines the actual differences between the solutions.

The solution space exploration methodology was applied to a set of  $RC+C$  networks, with the number of cliques  $nc$  varying between 2 and 25 and the clique size  $cs$  ranging from 3 to 8. This setup tests the ability of algorithms to generate different types of solution spaces. The results are presented in Fig. 4 and summarised in Table 2.

The results indicate that algorithms exhibit a prevalent tendency to generate specific types of solution spaces, but they also reveal inherent limitations. For instance, most diagrams display a Sparse solution space, particularly in cases with 2 or 3 large cliques, as seen in the bottom right corner of Fig. 4. Additionally, when  $nc = 2$ , an extreme case of a graph composed of two cliques connected by one edge, particularly for smaller clique sizes, most algorithms struggle to detect a single solution, with performance improving as clique size increases, highlighting their limitations in handling minimal community structures.

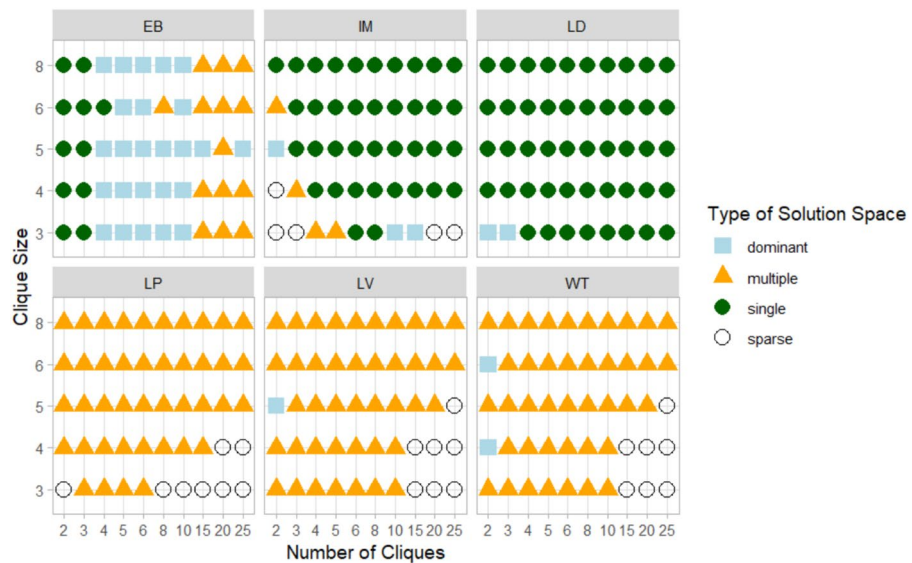
Beyond the Single case, comparing partitions is informative, as it reveals whether multiple solutions differ only marginally or diverge substantially, potentially forming clusters of alternative solutions. Such comparisons can be conducted using similarity measures between partitions, for example the Normalised Mutual Information (NMI), combined with the estimated probabilities of partitions in the solution space. For example, in the  $RC+C$



**Fig. 3.** An illustration of multiplicity of solutions and input-ordering bias. Test carried out on a Ring of Cliques with  $n_c = 4$ ,  $cs = 5$  with a central node. The bars show the frequency (out of 1000 runs) of assigning the central node to each of the 5 possible partitions (P1-P5). The network structure creates 5 equivalent solutions that are equally likely, so frequencies should be uniformly distributed if algorithms are unbiased. Orange bars: fixed node ordering. Blue bars: permuted node ordering at each iteration.

Feature/Method	EB	LV	LD	LP	IM	WT
(1) Multiple solutions	No	Yes	No	Yes	No	No
(2) Single-node community	No	No	Yes	No	Yes	No
(3) Input ordering bias	Yes	Yes	Yes	No	Yes	Yes

**Table 1.** Comparison of community detection algorithms across the identified issues. Three key behavioral characteristics are reported: (1) Multiple solutions—whether the algorithm produces different solutions at each run regardless of input order, (2) capability to assign a single node to its own community, and (3) sensitivity to input node ordering. EB = Edge Betweenness, LV = Louvain, LD = Leiden, LP = Label Propagation, IM = Infomap, WT = Walktrap.



**Fig. 4.** Prevalent type of solution space generated by different algorithms.

Algorithm	EB	LV	LD	LP	IM	WT
Prevalent type	Dominant	Single	Single	Multiple	Multiple	Multiple
Single ( $cs = 5$ )	$nc < 4$	$nc > 3$	Always	Never	Never	Never
Sparse	Never	$cs < 5$	Never	$cs < 6$	$cs < 5$	$cs < 6$

**Table 2.** Comparison of methods across the identified issues.

	WT	LV	LD	LP	IM
Convergence	Stabilisation	Stabilisation	max. trials	stabilisation	max. trials
Trials	18	29	50	29	50
Total solutions	1	29	11	29	13
Valid solutions	1	24	0	20	13
Type of solution space	Single	Sparse	Multiple	Sparse	Multiple

**Table 3.** Summary of solution space exploration on the Horizon network  $G_{2024}$ . The six algorithms yield markedly different outcomes: Walktrap (WT) converges to a Single solution, Label Propagation (LP) and Louvain (LV) generate Sparse solution spaces, producing a different solution at each run. Leiden (LD) results in an Multiple solution space, where all solutions are degenerate. Infomap (IM) produces a Multiple solution space with two solutions of overlapping probabilities clearly separated from the others. Finally, Edge Betweenness (EB) could not deliver any solution due to excessive computational cost. Parameters:  $\delta = 0.1$  and  $t_{max} = 50$ .

network (Fig. 2), the four partitions  $P_1$ – $P_4$  constitute a Multiple solution space, occurring with comparable estimated probabilities ( $\approx 0.25$  each) and differing only in the assignment of the central node.

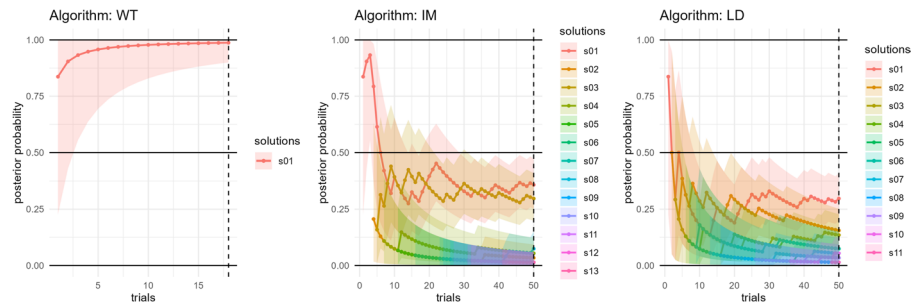
### Application on a real network

To illustrate how the exploration of the solution space on real data allows replicable and interpretable results, we present an example of application to a large, dense network from the Horizon Projects Network dataset. The source of data is CORDIS (Community Research and Development Information Service), the European Commission's primary public repository and portal to disseminate information on all EU-funded research projects and their outcomes (<https://cordis.europa.eu/>). This dataset captures collaboration linkages between organisations participating in EU-funded research projects between 2015 and 2027 in various domains; a comprehensive description of the dataset is provided in<sup>13</sup>. For illustrative purposes, we focus on the collaboration network emerging from projects in the hydrogen energy domain that start in 2024 and retrieved using hydrogen as keyword. We denote this network as  $G_{2024}$ . It is composed of 648 nodes (nodes represent organisations, in their role of partners of one or more Horizon projects on hydrogen) and over 6157 edges denoting collaborative relationships—i.e., two organisations participating in the same project within a given year. Edge weights reflect the funded amount granted to each collaborative project.

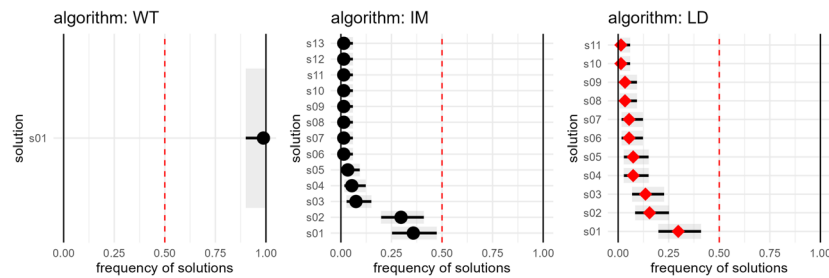
An overly simple approach to the analysis of  $G_{2024}$  would be to apply a community detection algorithm (e.g., Louvain) and directly interpret the resulting partition. However, as discussed above, we argue that each run of an algorithm should be regarded as generating a sample from the solution space; therefore, we applied all the algorithms mentioned above, and the results are summarised in Table 3.

This network represents a challenging case, and each method yields a distinct type of solution space. Walktrap (WT) identifies a single solution, and after 18 trials matches the stability condition. Louvain (LV) and Label Propagation (LP) generate a Sparse solution space, with a different solution at each trial, which reaches the termination condition after 29 trials. Both produce some non-valid solutions (specifically, some communities are internally disconnected). Leiden (LD) results in a Multiple solution space, and the exploration stops after  $t_{max}$  trials. However, in this case all the solutions are degenerate, as the number of inter-community edges is greater than the number of intra-community edges, contradicting the general definition of community. If this is deemed as a relevant issue for the analysis, the solution space may be considered Empty. Finally, Edge Betweenness (EB) could not be computed in a reasonable time, confirming its impracticality on networks of this size.

Figure 5 presents the evolution of posterior probabilities  $\hat{p}_i(t)$  together with credible intervals  $[p_i^l, p_i^u]$ , for three representative algorithms (WT, IM and LD), revealing heterogeneous convergence behaviours. Walktrap (WT) converges rapidly by  $t = 18$ , with shrinking intervals consistent with a Single solution space. Infomap (IM) oscillates among alternatives until two competing partitions of comparable probability emerge, exemplifying a Multiple case. Leiden (LD) remains unstable throughout and does not satisfy the convergence criteria, thus generates a Sparse solutions space; exploration halts at the maximum number of trials,  $t_{max} = 50$ . Figure 6 complements this view with the final distribution of solution probabilities and uncertainties: each point reports  $\hat{p}_i$  with horizontal bars for  $[p_i^l, p_i^u]$ , while the dashed vertical line at  $\hat{p}_i = 0.5$  serves only as a visual reference. These figures further illustrate why characterising the solution space is critical for downstream analyses that often rely on a single representative partition of  $G$ . In this example, a single run is defensible only for Walktrap (WT), which yields a single solution, whereas for Infomap (IM) and Leiden (LD) single-run outputs would



**Fig. 5.** A graphical representation of the successive steps in the exploration of the solution space for the Horizon network  $G_{2024}$  using three representative algorithms. Each curve shows the posterior probability  $\hat{p}_i$  of solution  $P_i$  as a function of the trial number  $t$ , with shaded areas denoting the credible intervals  $[p_i^l, p_i^u]$ .



**Fig. 6.** Distribution of solution frequencies and credible intervals for the Horizon network  $G_{2024}$  obtained with three algorithms (Walktrap, Infomap and Leiden). Each point represents the estimated posterior probability  $\hat{p}_i$  of solution  $P_i$ , with horizontal bars denoting the Bayesian credible intervals  $[p_i^l, p_i^u]$  and the vertical dashed line marking the  $\hat{p}_i = 0.5$  threshold.

overlook the multiplicity and instability revealed by the posterior and risk drawing misleading inferences. When the solution space is Dominant or Multiple and a single partition is nevertheless required, a consensus procedure is advisable, aggregating partitions and optionally weighting them by their posterior probabilities  $\hat{p}_i$ . In Sparse spaces, consensus may be appropriate only when solutions form one or a few coherent clusters; otherwise the evidence may support reporting the absence of a credible community structure, rather than forcing a single partition.

## Conclusions

This study underscores the necessity of exploring the full solution space in community detection, particularly when working with large and complex networks, and introduces a taxonomic framework to classify its structure based on the number of unique solutions, their relative frequencies, and convergence behaviour.

Exploration of solution space proceeds via repeated runs of a given community detection algorithm on the same network, with node order permuted at each run. A Bayesian model assesses convergence and estimates solution probabilities, yielding a defensible stopping rule that balances exploration accuracy and computational cost. The taxonomy classifies solution spaces into five types (Single, Dominant, Multiple, Sparse, and Empty) providing clear diagnostics of partition reliability across algorithms and a shared vocabulary for interpretation.

Our empirical analysis shows that most widely used algorithms are sensitive to input (node) ordering, challenging the assumption that outcomes reflect topology alone. Systematically permuting the input reduces this bias and yields more reliable partitions. More broadly, repeated exploration of the solution space reveals that a single run can miss important structure and even elevate a minority partition to apparent representativeness. The main limitation of the proposed framework is computational: the total runtime scales with the number of trials, making the method impractical at large scale.

Future work should assess how solution space exploration impacts applications across different domains and scales. This involves broadening empirical tests to include additional algorithms, examining the grey area between Multiple and Sparse solution spaces, and integrating quality and similarity metrics (e.g., modularity, NMI) to refine the treatment of cases that, in this paper, were more broadly defined as “degenerate” or “non-valid” solutions.

The methodology should also be tested across a wider range of domains, extending beyond the innovation network case presented here to fields such as biology, neuroscience, and other technological or organisational networks.

## Data availability

All data and code used in this study are available under a CC-BY license. The sample network  $G_{2024}$  is part of the Horizon Projects Network dataset (<https://doi.org/10.5281/zenodo.13765372>). The analysis was conducted using the R programming language, the package igraph (<https://r.igraph.org/>), and our package communities (<https://doi.org/10.5281/zenodo.13594209>).

Received: 2 May 2025; Accepted: 25 September 2025

Published online: 31 October 2025

## References

- Kojaku, S., Radicchi, F., Ahn, Y. Y. & Fortunato, S. Network community detection via neural embeddings. *Nature Communications* **15**(1), 9446 (2024).
- Light, R. & Moody, J. *The Oxford handbook of social networks*. Oxford University Press (2020).
- Junker, B. H. & Schreiber, F. *Analysis of biological networks*, vol. 2. Wiley Online Library (2008).
- Stefano, D., Fuccella, V., Vitale, M. P. & Zaccarin, S. The use of different data sources in the analysis of co-authorship networks and scientific performance. *Social Networks* **35**(3), 370–381 (2013) <https://www.sciencedirect.com/science/article/pii/S0378873313000403>.
- Newman, M. E. J. The structure of scientific collaboration networks. *Proceedings of the national academy of sciences* **98**(2), 404–409 (2001).
- Bassett, D. S. & Sporns, O. Network neuroscience. *Nature neuroscience* **20**(3), 353–364 (2017).
- Bassett, D. S. & Bullmore, E. *Small-world brain networks*. *The neuroscientist* **12**(6), 512–523 (2006).
- Cerina, F., Chessa, A., Pammolli, F. & Riccaboni, M. Network communities within and across borders. *Scientific Reports* **4**(1), 4546 (2014).
- Morea, F. & De Stefano, D. Innovation patterns within a regional economy through consensus community detection on labour market network. *Proceedings of the Statistics and Data Science Conference (2023)*, <https://arts.units.it/handle/11368/3046559>.
- Fortunato, S. & Hric, D. Community detection in networks: A user guide. *Physics Reports* **659**, 1–44 (2016).
- Bedi, P. & Sharma, C. Community detection in social networks. *WIREs Data Mining and Knowledge Discovery* **6**(3), 115–135 (2016) <https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1178>.
- Ding, J. et al. Community evolution prediction based on feature change patterns in social networks. *Scientific Reports* **15**(1), 14608 (2025).
- Morea, F., Soraci, A. & De Stefano, D. Mapping leadership and communities in eu-funded research through network analysis [version 2; peer review: 1 approved, 2 approved with reservations]. *Open Research Europe* **4**(268) (2025).
- Diboune, A., Slimani, H., Nacer, H. & Bey, K. B. A comprehensive survey on community detection methods and applications in complex information networks. *Social Network Analysis and Mining* **14**(1), 93. <https://doi.org/10.1007/s13278-024-01246-5> (2024).
- Fortunato, S. & Newman, M. E. J. 20 years of network community detection. *Nature Physics* **18**(8), 848–850 (2022).
- Khawaja, F. R., Zhang, Z., Memon, Y. & Ullah, A. Exploring community detection methods and their diverse applications in complex networks: a comprehensive review. *Social Network Analysis and Mining* **14**(1), 115. <https://doi.org/10.1007/s13278-024-01274-1> (2024).
- Menardi, G. & Stefano, D. Density-based clustering of social networks. *Journal of the Royal Statistical Society Series A: Statistics in Society* **185**(3), 1004–1029. <https://doi.org/10.1111/rssa.12796> (2022).
- Newman, M. E. J. & Girvan, M. Finding and evaluating community structure in networks. *Phys. Rev. E* **69**, 026113 (2004).
- Danon, L., Díaz-Guilera, A., Duch, J. & Arenas, A. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment* **2005**(09), 9008 (2005).
- Ghoshal, A. K., Das, N., Bhattacharjee, S. & Chakraborty, G. A fast parallel genetic algorithm based approach for community detection in large networks. In: 2019 11th International Conference on Communication Systems & Networks (COMSNETS), pp. 95–101 (2019)
- Ghoshal, A. K., Das, N. & Das, S. Disjoint and overlapping community detection in small-world networks leveraging mean path length. *IEEE Transactions on Computational Social Systems* **9**(2), 406–418 (2022).
- Lee, C. & Wilkinson, D. J. A review of stochastic block models and extensions for graph clustering. *Applied Network Science* **4**(1), 1–50 (2019).
- Jin, D. et al. A survey of community detection approaches: From statistical modeling to deep learning. *IEEE Transactions on Knowledge and Data Engineering* **35**(2), 1149–1170 (2021).
- Škrlj, B., Kralj, J. & Lavrač, N. Embedding-based silhouette community detection. *Machine Learning* **109**, 2161–2193 (2020).
- Clauset, A., Moore, C. & Newman, M. E. Hierarchical structure and the prediction of missing links in networks. *Nature* **453**(7191), 98–101 (2008).
- Perlasca, P. et al. Multi-resolution visualization and analysis of biomolecular networks through hierarchical community detection and web-based graphical tools. *PLOS ONE* **15**(12), 1–28. <https://doi.org/10.1371/journal.pone.0244241> (2020).
- Benati, S., Puerto, J., Rodríguez-Chía, A. M. & Temprano, F. Overlapping communities detection through weighted graph community games. *PLOS ONE* **18**(4), 1–35. <https://doi.org/10.1371/journal.pone.0283857> (2023).
- Palla, G., Derényi, I., Farkas, I. & Vicsek, T. Uncovering the overlapping community structure of complex networks in nature and society. *Nature* **435**(7043), 814–818 (2005).
- Ponomarenko, A., Pitsoulis, L. & Shamshetdinov, M. Overlapping community detection in networks based on link partitioning and partitioning around medoids. *PLOS ONE* **16**(8), 1–43. <https://doi.org/10.1371/journal.pone.0255717> (2021).
- Airoldi, E., Blei, D., Fienberg, S. & Xing, E. Mixed membership stochastic blockmodels. *Advances in neural information processing systems* **21** (2008).
- Moradan, A., Draganov, A., Mottin, D. & Assent, I. Ucode: Unified community detection with graph convolutional networks. *Machine Learning* **112**(12), 5057–5080 (2023).
- Good, B. H., de Montjoye, Y. A. & Clauset, A. Performance of modularity maximization in practical contexts. *Phys. Rev. E* **81**, 046106 (2010).
- Chen, M., Nguyen, T. & Szymanski, B. K. On measuring the quality of a network community structure. In: 2013 International Conference on Social Computing, pp. 122–127 (2013)
- Ghoshal, A. K. & Das, N. On diameter based community structure identification in networks. In: Proceedings of the 18th International Conference on Distributed Computing and Networking. ICDCN '17, Association for Computing Machinery, New York, NY, USA (2017), <https://doi.org/10.1145/3007748.3018285>.
- Calatayud, J., Bernardo-Madrid, R., Neuman, M., Rojas, A. & Rosvall, M. Exploring the solution landscape enables more reliable network community detection. *Phys. Rev. E* **100**, 052308 (2019) <https://link.aps.org/doi/10.1103/PhysRevE.100.052308>.
- Holmgren, A., Bernenko, D. & Lizana, L. Mapping robust multiscale communities in chromosome contact networks. *Scientific Reports* **13**(1), 12979 (2023).

37. Morea, F. & De Stefano, D. Enhancing stability and assessing uncertainty in community detection through a consensus-based approach. ArXiv preprint (2024), <https://doi.org/10.48550/arXiv.2408.02959>.
38. Tandon, A., Albeshri, A., Thayanathan, V., Alhalabi, W. & Fortunato, S. Fast consensus clustering in complex networks. *Phys. Rev. E* **99**, 042301 (2019) <https://link.aps.org/doi/10.1103/PhysRevE.99.042301>.
39. Akoglu, L., Tong, H. & Koutra, D. Graph based anomaly detection and description: a survey. *Data mining and knowledge discovery* **29**(3), 626–688 (2015).
40. Helling, T. J., Scholtes, J. C. & Takes, F. W. A community-aware approach for identifying node anomalies in complex networks. In: International Conference on Complex Networks and their Applications. pp. 244–255. Springer (2018).
41. Dey, A., Kumar, B. R., Das, B. & Ghoshal, A. K. Outlier detection in social networks leveraging community structure. *Information Sciences* **634**, 578–586 (2023) <https://www.sciencedirect.com/science/article/pii/S0020025523004437>.
42. Win, H. N. & Lynn, K. T. Community and outliers detection in social network. In *Big Data Analysis and Deep Learning Applications* (eds Zin, T. T. & Lin, J. C. W.) 58–67 (Springer Singapore, Singapore, 2019).
43. Chakraborty, T., Park, N., Agarwal, A. & Subrahmanian, V. S. Ensemble detection and analysis of communities in complex networks. *ACM/IMS Trans. Data Sci.* **1**(1) <https://doi.org/10.1145/3313374> (2020).
44. Chakraborty, T., Srinivasan, S., Ganguly, N., Bhowmick, S. & Mukherjee, A. Constant communities in complex networks. *Scientific reports* **3**(1), 1825 (2013).
45. Csardi, G. et al. *igraph: Network analysis and visualization in r* (2023), <https://igraph.org>
46. R Core Team: R: A language and environment for statistical computing (2022), <https://www.R-project.org>
47. Blondel, V. D., Guillaume, J. L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* **2008**(10), P10008. <https://doi.org/10.1088/1742-5468/2008/10/P10008> (2008).
48. Traag, V. A., Waltman, L. & Van Eck, N. J. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports* **9**(1), 5233 (2019).
49. Rosvall, M. & Bergstrom, C. T. An information-theoretic framework for resolving community structure in complex networks. *Proceedings of the national academy of sciences* **104**(18), 7327–7331 (2007).
50. Smiljanic, J. et al. Community detection with the map equation and infomap: Theory and applications. ArXiv preprint (2023). <https://arxiv.org/abs/2311.04036>.
51. Pons, P. & Latapy, M. Computing communities in large networks using random walks. In: Computer and Information Sciences-ISCIS 2005: 20th International Symposium, Istanbul, Turkey, October 26–28, 2005. Proceedings 20. pp. 284–293. Springer (2005).
52. Raghavan, U. N., Albert, R. & Kumara, S. Near linear time algorithm to detect community structures in large-scale networks. *Physical review E* **76**(3), 036106 (2007).

### Author contributions

Both authors contributed equally to the work presented in this paper. D.D.S. and F.M. jointly developed the idea and methodology. F.M. primarily wrote the code and generated the plots. Both authors reviewed the manuscript.

### Funding

This work has been granted by the PRIN project “Models and methods for the analysis of collaboration networks”, CUP J53D23011540006, Principal Investigator (PI) prof. Domenico De Stefano.

### Declarations

### Competing interests

The authors declare no competing interests.

### Additional information

**Supplementary Information** The online version contains supplementary material available at <https://doi.org/10.1038/s41598-025-22046-7>.

**Correspondence** and requests for materials should be addressed to D.D.S.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher’s note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2025