Applied Mathematics and Computation 431 (2022) 127310

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Importance of Numerical Implementation and Clustering Analysis in Force-Directed Algorithms for Accurate Community Detection

Alessandra M.M.M. Gouvêa[a,*], Nicolás Rubido[b,c], Elbert E.N. Macau[a], Marcos G. Quiles[a]

[a] Federal University of São Paulo,1201, Av. Cesare Monsueto Giulio Lattes,São José dos Campos,São Paulo,12247-014,Brazil
[b] University of Aberdeen, King's College, Institute for Complex Systems and Mathematical Biology,Aberdeen,AB24 3UE, United Kingdom
[c] Universidad de la República, Instituto de Física de Facultad de Ciencias,Iguá 4225, Montevideo 11400 Uruguay

ARTICLE INFO

Article history:
Received 16 March 2022
Revised 26 May 2022
Accepted 5 June 2022
Available online 23 June 2022

Keywords:
Complex Networks
Community Detection
Force-Directed Algorithms
Clustering Analysis

ABSTRACT

Real-world networks show community structures – groups of nodes that are densely intra-connected and sparsely inter-connected to other groups. Nevertheless, Community Detection (CD) is non-trivial, since identifying these groups of nodes according to their local connectivity can hold many plausible solutions, leading to the creation of different methods. In particular, CD has recently been achieved by Force-Directed Algorithms (FDAs), which originally were designed as a way to visualize networks. FDAs map the network nodes as particles in a D-dimensional space that are affected by forces acting in accordance to the connectivity. However, the literature on FDA-based methods for CD has grown in parallel from the classical methods, leaving several open questions, such as how accurately FDAs can recover communities compared to classical methods. In this work, we start to fill these gaps by evaluating different numerical implementations of 5 FDA methods and different clustering analyses on state-of-the-art network benchmarks – including networks with or without weights and networks with a hierarchical organisation. We also compare these results with 8, well-known, classical CD methods. Our findings show that FDA methods can achieve higher accuracy than classical methods, albeit their effectiveness depends on the chosen setting – with optimisation techniques leading over numerical integration and distance-based clustering algorithms leading over density-based ones. Overall, our work provides detailed information for any researcher aiming to apply FDAs for community detection.

1. Introduction

Many disciplines have been successfully representing real-world data sets as networks. A network is a topological model of the inter-relationships between the elements in a data-set [4], i.e., links and nodes, respectively. However, with growing and evolving data-sets [24], network analysis is becoming rapidly intractable. A possible solution to this problem is to group nodes into subsets that share some common properties, such as into communities [21].

* Corresponding author.
E-mail address: alessandra.marli@unifesp.br (A.M.M.M. Gouvêa).

https://doi.org/10.1016/j.amc.2022.127310
0096-3003/

The concept of a community (or a module) is intuitive: a subset of densely connected nodes but sparsely inter-connected to other nodes (or communities). However, a universally accepted definition is still missing. For example, the initial notions started from cliques [35] and k-plexes [61] and evolved to the notion of strong and weak communities [54]. These concepts require discriminating links for any set of nodes in terms of internal – intra-community – and external – inter-community – links. The resultant communities are dense sub-graphs that can be separated from each other. Recently, the classical approach has been challenged by more realistic concepts [19,27,28,58,60], like overlapping communities – where a node can belong to more than one community – and hierarchy – where nodes can be arranged in different groups at different levels. These modern concepts define communities through probabilities, which implies finding the proportion of shared links within different sets such that nodes have different participation probabilities across the resultant communities [19,27,32,60]. As a result of these broad definitions, there are various methods to tackle Community Detection (CD).

Classically, CD has been accomplished by using edge betweenness [21], spectral partitions [40], modularity optimisation [8], finding random-walks trapped locations [50], seeking to maximise information compression in the network [59], via synchronization [36], or by random relabelling of the nodes [55], to name a few of the most popular methods. In particular, maximising modularity is similar to solving the graph partitioning problem, i.e., cutting the least amount of links to partition the network into disconnected components.

In the early 2000s, Force-Directed Algorithms (FDAs) started to be successfully used for CD. FDAs are a class of methodologies initially developed to visualise networks with unknown structural properties, providing a solution to the graph drawing problem employing particle-like simulations [5,7,29]. Namely, by setting a force model, where nodes are set as particles on a D-dimensional space that evolve according to the forces, the resultant layout delivers a visualisation of the network. The choice of forces for network visualisation may seek to minimise the number of crossing edges, maximise the distance between non-adjacent nodes, and maximise the symmetries in the drawing [51], with the goal of producing an insightful visual aesthetic. Importantly, with the right force model, FDAs can organise nodes into communities [6,10,33,34,43,44,46,48,52,53,59,67,70], holding intuitive layouts and having simple algorithmic implementations.

The development of FDAs for CD has grown rapidly but in parallel to the classical CD methods and literature – which scarcely mentions FDAs [22]. This rapid growth and decoupled development left several open questions. For example, we are unaware of how accurately FDAs can recover communities compared to the classical methods, whether one can increase the FDA's accuracy by changing their numerical implementation, the space dimension (D), or clustering algorithm, and how robust this accuracy is to varying degrees of community mixing.

Here, we fill this gap by testing 5 different FDAs under two state-of-the-art network benchmarks and comparing their accuracy with that of 8 classical CD methods. In these tests, we analyse different numerical implementations for the FDAs, spatial dimensions, and clustering algorithms. We also extend these analyses to weighted networks with communities and networks with a hierarchical organization.

In summary, to the best of our knowledge, our work is the first one to detail FDA's accuracy dependence on the force model, its numerical implementation, choice of dimensionality, and clustering technique. Importantly, we show that FDAs can outperform classical methods. For example, we find that the Edge Linear-Logarithmic Model [45] implemented by the conjugate gradient method in a 3-dimensional space (with the optimal result if $D > 4$) and a distance-based clustering technique is enough to rival or outperform the 8 classical methods. Thus, our work shows that FDA methods are powerful tools for community detection and network analysis.

2. Methods

2.1. Validation Process

The search for an optimal community detection method happens on two fronts: (i) decreasing computational complexity and (ii) getting the greatest possible accuracy. Validating a new method in terms of computational complexity is a well-defined problem that can be derived analytically or from simulations. In contrast, validating accuracy is more challenging since it means dealing with the weak concept of community [31]. It implies analysing the community structures recovered by the proposed algorithm from a network where such structures are known *a priori*.

2.1.1. Benchmark Networks

There is a large amount of real-world data with known community structures and artificial networks generated with desired community structures that can be used as benchmarks [17,21,23,30,31,56,69].

The first and most famous network benchmark was introduced by Girvan and Newman [21] (GN), which we use in this work. The GN benchmark describes undirected and unweighted random networks with $N = 128$ nodes organized into 4 groups of 32 nodes whose average degree is 16. Mixing of communities in GN networks [21] is carried by reshuffling edges, controlled by a mixing parameter μ that indicates the expected ratio of edges connecting nodes from different communities with respect to the total number of edges. That is, if $\mu = 0$, nodes from different communities do not share edges; hence, communities are completely isolated. If $\mu = 0.5$, half of the edges connect intra-community nodes and the other half inter-community nodes, which is typically when community detection methods start to lose accuracy [17].

Lacichinetti, Fortunato, and Radichi (LFR) networks [31] have broad degree and community distributions, following power laws with exponents γ and β, respectively. For example, typical real-world networks show values within a range of $2 \leq \gamma \leq$

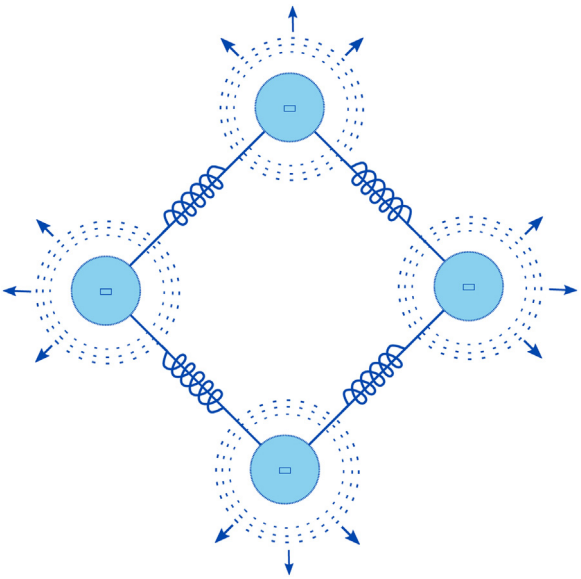

Fig. 1. Force model where nodes are represented as positive charges that magnetically repel each other and edges are represented as springs that attract adjacent nodes.

3 and $1 \leq \beta \leq 2$ [31]. In this work, we generate LFR networks with $N = 1000$ nodes, $\gamma = 2$, $\beta = 1$, $\langle k \rangle = 20$, and $\max(k) = 50$, but under 2 different scenarios: i) small communities, where $\min_C = 10$ and $\max_C = 50$ nodes, and ii) big communities, where $min_C = 20$ and $max_C = 100$ nodes. Parameters are chosen so that we can compare our results with previous reports [30,31,53]. Moreover, we generate weighted LFR (LFRw) networks [30] by setting a power-law distribution of edge weights with exponent 2 (the same as the degree distribution). Edge weights represent the intensity of the relationship between two nodes, which in real-world network analysis, is of paramount importance.

Noack [42] defined a hierarchical network with 16 cliques (all-to-all connected) of 50 nodes, i.e., $N = 16 \times 50 = 800$ nodes, which we also use to test the community detection algorithms' accuracy. Hierarchy is set by arranging these cliques such that they hold 4 communities with 4 cliques each. We do this by fixing the probability of connecting two nodes from different communities to $p = 0.16$, and if the two nodes are within the same community to $p = 0.32$.

Among the real-world networks used in the CD literature, Zachary's Karate Club is one of the most popular. This network models friendship data constructed by Zachary [71] from a variety of measures to estimate the bond of friendship between individuals [21]. The network comprises 34 nodes whose IDs represent the karate club members, whereas the edges among them describe their social interactions outside the club. Here, we use the standard unweighted version described by Girvan and Newman [21] to illustrate some FDA results.

2.1.2. Accuracy Quantification

The accuracy of an algorithm is estimated by a quantitative measure of similarity between the algorithm's outcome and the ground truth. A list of measures can be found in Fortunato [17]. Here, we use the Normalised Mutual Information (NMI) [11], $0 \leq \text{NMI} \leq 1$, where $\text{NMI} = 1$ means 100% accuracy – the recovered community structure matches perfectly the benchmark's communities – and $\text{NMI} = 0$ means there is complete mismatch.

2.2. Force-Directed Algorithms

Let a network, $\mathcal{G}$, be an ordered pair of sets, $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is a set with N nodes and $\mathcal{E}$ is a set with M edges connecting the nodes. Force-Directed Algorithms (FDA) treat nodes as physical particles (masses or bodies) in a D-dimensional space, $\mathbb{R}^D$, and edges as forces (interactions) between nodes. Consequently, FDAs require choosing a force model, setting D, and finding the stable steady-state, which gives a particle layout that can be used for community detection by applying clustering analysis.

Natural analogies can be drawn when a force model has a physical meaning. For example, Fig. 1 shows a model where nodes are described as magnetic particles that repulse each other (regardless of whether they share an edge). Edges are replaced with springs that attract the particles. FDAs can also be defined by non-physical laws, without any physical analogy [51].

2.2.1. General (a, r)-Energy Models

Most force models [22,43] are defined from defining an attractive function $f(r_{ij})$ and a repulsive function $g(r_{ij})$, where $r_{ij} = \left\|\vec{x}_i - \vec{x}_j\right\|$ is the Euclidean distance between the particles (nodes) i and j located at positions $\vec{x}_i \in \mathbb{R}^D$ and $\vec{x}_j \in \mathbb{R}^D$, respectively. The (instantaneous) potential energy of the system, $U(t)$, at any given time t is given by

$$U(t) = \sum_{i>j=1}^{N} f(r_{ij}(t)) + \sum_{i>j=1}^{N} g(r_{ij}(t)), \tag{1}$$

where the symmetric terms are unaccounted because generally $f(r_{ij}(t)) = f(r_{ji}(t))$ and $g(r_{ij}(t)) = g(r_{ji}(t))$. Importantly, U is at a minimum in the steady-state.

The models where $f(r_{ij})$ and $g(r_{ij})$ can be written as

$$f(r_{ij}) = C_1 \frac{\left\|\vec{x}_i - \vec{x}_j\right\|^{(a+1)}}{a+1} = C_1 \frac{r_{ij}^{(a+1)}}{a+1},$$
$$g(r_{ij}) = -C_2 \frac{\left\|\vec{x}_i - \vec{x}_j\right\|^{(r+1)}}{r+1} = -C_2 \frac{r_{ij}^{(r+1)}}{r+1},$$

are known as (a, r)-energy models [43]; $C_1, C_2 > 0$ being constants possibly having some information about the edge (i, j) or the nodes degrees of particles i and j. These models define conservative forces from the negative gradient of U. Namely, the attractive, $\vec{F}_A$, and repulsive, $\vec{F}_R$, forces acting on particle i due to the other particles are given by

$$\vec{F}_A(i) = -\nabla_i \sum_{j=1}^{N} f(r_{ij}) = -\sum_{j=1}^{N} C_1 \left\|\vec{x}_i - \vec{x}_j\right\|^{a} \hat{r}_{ij}, \tag{2}$$

$$\vec{F}_R(i) = -\nabla_i \sum_{j=1}^{N} g(r_{ij}) = \sum_{j=1}^{N} C_2 \left\|\vec{x}_i - \vec{x}_j\right\|^{r} \hat{r}_{ij}, \tag{3}$$

where $\nabla_i = (\partial/\partial x_i, \partial/\partial y_i, \partial/\partial z_i, \ldots)$ is the D-dimensional gradient acting on the i-th particle's position and $\hat{r}_{ij} = \left(\vec{x}_i - \vec{x}_j\right)/\left\|\vec{x}_i - \vec{x}_j\right\|$ defines a unit vector going from particle j to particle i (force direction).

2.2.2. Fruchterman-Reingold Model

The Frucherman-Reingold (FR) model [66] is a $(2, -1)$-energy model that considers nodes as magnetic particles usually in a $D = 2$ space with forces given by

$$\vec{F}_A(i) = -\sum_{j=1}^{N} \frac{A_{ij}}{\kappa} r_{ij}^2 \hat{r}_{ij} \text{ and } \vec{F}_R(i) = \sum_{j=1}^{N} \kappa^2 r_{ij}^{-1} \hat{r}_{ij}, \tag{4}$$

where $\kappa = C\sqrt{area/N}$ is defined as the optimal distance between the nodes, which is typically found experimentally, C being the experimental constant and *area* being the size of the two-dimensional space available to place the particles.

An example of the resultant drawing from this model on a 500 node network with 4 communities is shown in Fig. 2. We note that attraction happens solely on adjacent nodes. However, repulsion acts on all nodes – most works using drawing graph techniques to detect communities in vanilla are based on this model [22].

2.2.3. Linear-Logarithmic Energy Model

The Linear-Logarithmic (LinLog) energy model was proposed by Noack [42] to draw clustered small-world networks, where $(a, r) = (0, -1)$. It was subsequently improved [43–45], to the point of being the first method to reveal clusters in undirected networks [22]. Like Fruchterman-Reingold Model, it uses magnetic-like particles as nodes and links as spring-like forces. The following energy equation gives the motion

$$U = \sum_{\{i,j\}\in\mathcal{E}} \left\|\vec{x}_i - \vec{x}_j\right\| - \sum_{\{i,j\}\in\mathcal{V}^2} \log \left\|\vec{x}_i - \vec{x}_j\right\|, \tag{5}$$

where $\mathcal{V}^2 = \mathcal{V} \times \mathcal{V}$ means that the addition is over all unordered pairs of nodes and $\mathcal{E}$ is the edges set, meaning the addition only happens for adjacent nodes.

The LinLog model has 2 extensions. One is called Node LinLog energy model and includes information of the network's weights, which is done by multiplying each attractive (first) term in Eq. (5) by the link weight, W_{ij}. Another is called Edge LinLog energy model and includes information of the node's neighbourhood, which is done by multiplying each repulsive (second) term in Eq. (5) by the node degree products, $k_i \times k_j$. In this work, we test both extensions.

In contrast to other models, the LinLog energy model draws distances with a topological interpretation [43]. The particle positions in $\mathbb{R}^D$ are such that the distance between any 2 disjoint sets of nodes, $\mathcal{V}_1$ and $\mathcal{V}_2$, is inversely proportional to their

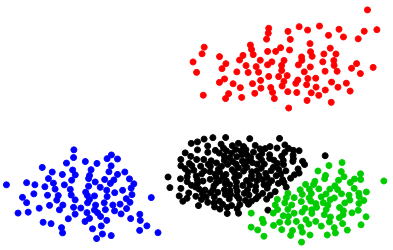

Fig. 2. Network drawing by Frucherman-Reingold model. This network has 500 nodes and 4 modules. A central module with 200 nodes and a probability of inter-connection $P(i, j) = 0.064$ (black), and other 3 with 100 nodes each, where $P(i, j) = 0.016$ (green), 0.008 (blue), and 0.004 (red) when i belongs to the central module and j to the peripheral ones.

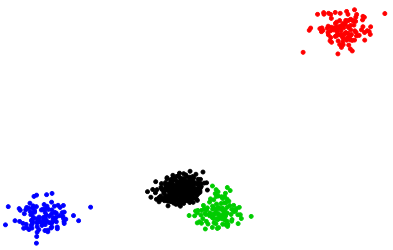

Fig. 3. Drawing of the network in Fig. 2 by the Edge Linear-Logarithmic energy model. Here the probability of inter-connection $P(i, j)$ is better represented by the distance of the groups of nodes on the plane.

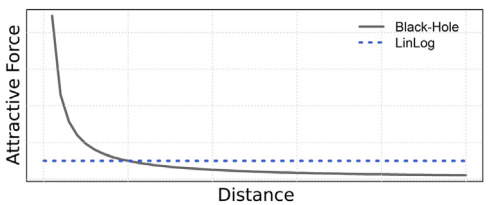

Fig. 4. Attractive forces defined by Lim et. al. [33] and Noack [42–45] in the Black-Hole and LinLog energy models, respectively.

coupling, $c(\mathcal{V}_1, \mathcal{V}_2)$, which is defined as $c(\mathcal{V}_1, \mathcal{V}_2) = \frac{E[\mathcal{V}_1, \mathcal{V}_2]}{|\mathcal{V}_1| \times |\mathcal{V}_2|} = \frac{M_{12}}{N_1 N_2}$, where M_{12} is the number of links between the sets and $N_1 N_2$ is the number of node pairs between the sets (and maximum possible links).

When comparing the resultant planar drawing from the Frucherman-Reingold model in Fig. 2 with that from the Edge LinLog model in Fig. 3, we note that the 4 modules are better separated by the Edge LinLog model. Although, in both cases, we can see 3 modules of 100 nodes (green, blue, and red) around a central module of 200 nodes (black).

2.2.4. Black-Hole Model

Luce and Perry [33] put forward a modification to Eq. (5): distance dependent attraction forces, making the resultant model an $(a, r) = (-0.95, -1)$ energy model known as Black-Hole model. This modification leas to

$$\vec{F}_A(i) = -\sum_{j=1}^{N} W_{ij} \left\| \vec{x}_i - \vec{x}_j \right\|^{-0.95} \hat{r}_{ij}, \tag{6}$$

where W_{ij} is the link weight between nodes i and j (as in the Node LinLog energy model). The relationship between the force strength and the distance between nodes for both models is shown in Fig. 4. In practice, this aims to place nodes from a community as closely as possible, which allows clustering them by density-based clustering techniques.

2.2.5. QMR Model

The QMR model was proposed by Quiles et al. [53], with forces given by

$$\vec{F}_A(i) = -\sum_{j=1}^{N} \left(\frac{A_{ij}}{k_i} \right) \hat{r}_{ij}, \tag{7}$$

$$\vec{F}_R(i) = \eta \sum_{j=1}^{N} \left(\frac{1 - \delta_{ij} - A_{ij}}{k_i} \right) e^{-\left\| \vec{x}_i - \vec{x}_j \right\|} \hat{r}_{ij}, \tag{8}$$

where $\eta > 0$ is a constant, k_i is the i-th node degree, δ_{ij} is the Kronecker delta function, and A_{ij} is the adjacency matrix. Here, we use $\eta = 0.36$ as set by Quiles et al. [53]. The attraction in Eq. (7) is similar to the LinLog model ($a = 0$), acting only when node i and j share a link (as in most force models). Contrary to other models, Eq. (8) only repels non-adjacent nodes $(1 - \delta_{ij} - A_{ij})$ and cannot be expressed as an energy model because of the repulsive force.

2.3. Searching for the stable steady-state

A general goal in the implementation of force models is to reach a stable state as fast as possible, ensuring that the node distances in the final layout allow detecting the communities correctly [22] – or to meet the visual aesthetic in a graph drawing [13,65]. Specifically, once the force model is defined, all nodes (particles) are mapped to some initial position, which is usually set at random; that is $\vec{x}_i(t=0) = \{x_i(0), y_i(0), z_i(0), \ldots\} \in \mathbb{R}^D$ with $i = 1, \ldots, N$. The forces start acting on the particles from this initial position, causing them to change their positions until a stable steady-state is reached. In this state, all forces cancel each other out, and the particles have reached a final position that constitutes a graphical drawing of the network [65], which we refer to as the D-dimensional layout (or simply, the layout).

There are two well-established approaches to find stable states: Numerical Solutions (NS) and Optimisation Techniques (OT). In NS, the forces define a set of coupled differential equations on the particles' positions, such as Quiles et al. [53], $\dot{\vec{x}}_i = d\vec{x}_i/dt = \vec{F}_A(i) + \vec{F}_R(i)$, where the positions $\vec{x}_i(t)$ evolve with time, t, from their initial positions, $\vec{x}_i(0)$, for $i = 1, \ldots, N$. Therefore, NS require discretisation methods like Euler – used by Quiles et al. [53] – or Runge-Kutta. On the other hand, OT use the energy formulation as an objective function to be minimised, such as the potential energy from Eq. (1). Possible OT are the Fruchterman and Reingold, [20] heuristic or conjugate gradients Tunkelang [65,66] – used by Noack [45] and Lim [33].

Because most implementations are not easily adapted to solve other force models, we lack comparative studies on algorithmic implementations. In this work, we present the first set of exhaustive tests on Euler, Fruchterman-Reingold heuristic, and conjugate gradient approach implementations for 5 force models, which are summarised in Table 1 and grouped according to 9 settings, including changes to the number of iterations used to set the numerical convergence.

2.4. Methods for Clustering Analysis

Communities can be extracted from the steady-state solutions of the Force-Directed Algorithm (FDA) by applying Clustering Analysis (CA). CA is a generic term for the process of organising data into groups based solely on data-driven information [16]. The resultant groups are defined in terms of internal cohesion (homogeneity) and external isolation (separation), meaning that similar objects are grouped together. However, CA has the same problem as community detection: having a clear cluster definition. Moreover, given a data-set, different clustering techniques – or even the same technique applied with different input parameters – usually result in different groupings, as can be seen from Fig. 5. Consequently, although there are well-established methods to perform CA [1,16,25,63], choosing one to detect communities requires knowledge of CA methodologies.

We note that community detection has been achieved by using 2 classes of CA [22]: Distance-based or Density-based. Distance-based methods are the simplest and most fundamental; they are generally divided into Partitioning Methods (PM) and Hierarchical Methods (HM). PM organise a data-set with N objects (particles) into K partitions (clusters), requiring background knowledge or a strategy to specify K a priori. Such methods are typically inefficient in finding clusters when the data has complex shapes [25]. HM organise the objects as a tree with varying granularity levels, which do not require background knowledge but are usually limited to small or medium-sized data-sets.

Density-based methods assume that areas of low data density should be treated as noise (not belonging to any cluster). These methods, in principle, can be applied to find clusters of any shape – even when data has irregular or intertwined layouts – and without assumptions about cluster number [63]. However, their use requires answering key design questions: a) how is the density estimated? b) how is the connectivity defined? The answers can directly influence the results. Here, we test both CA classes using the methods shown in Fig. 5.

2.5. Classical Methods

Classical methods for community detection are found in many software packages, such as `igraph`. We use these methods as a reference to compare with FDA's accuracy. However, it is well-documented that there is no gold algorithm that presents accurate results for all networks [17,19,38,58]. In this work, we compared the force-directed algorithms with the following classical methods.

Edge betweenness was proposed by Girvan [21], being the first modern method for community detection [30], with a computational complexity of order $\mathcal{O}(N^3)$, N being the number of nodes. It iteratively removes edges according to their edge betweenness, which quantifies the number of shortest paths going through each edge. The edge with the highest betweenness likely represents a bridge between communities, hence, communities are revealed once all bridges are removed. Iterations stop when any other edge deletion ceases to increase modularity – a quality function that estimates the goodness of the network partition.

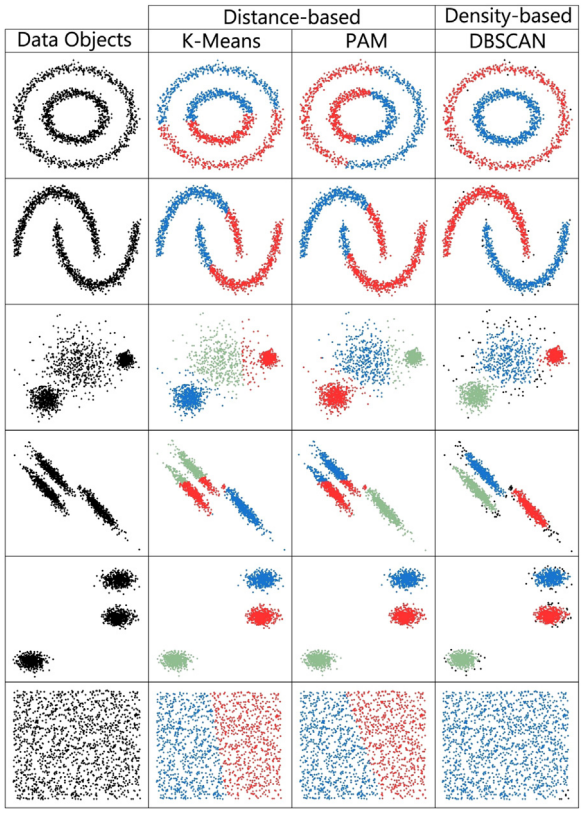

Fig. 5. Resultant clustering of data points (colours) according to distance-based (*K*-means or Partition Around Medoids) and density-based (Density-Based Spatial Clustering of Applications with Noise) algorithms. The initial two-dimensional layouts are shown in the left column panels. This Figure is inspired by scikit-learn 1.0.2 developed by Pedregosa et. al. [49].

Fast Greedy was proposed by Clauset et al. [8]. It is a greedy approach to optimise modularity globally, whose fast version is of order $\mathcal{O}(N \log^2 N)$. The method starts from the network's isolated nodes and iteratively adds its edges, only accepting edge additions that produce an increase in modularity.

Blondel method Blondel [3] (also called the Louvain method or fast modularity optimization) is a greedy approach to optimise modularity locally. Initially, it assigns a community to each node of the network. Nodes then aggregate to the community of their neighbours, producing an increase in modularity. All nodes within a community are replaced by a super-node, reducing the network's size. Aggregation continues until the modularity stops increasing (like Edge Betweenness), resulting in a complexity of order $\mathcal{O}(Nlog(N))$.

Leading Eigenvalue was proposed by Newman [40]. It is a spectral method looking to optimise modularity through the eigenvalues and eigenvectors of the adjacency matrix. It is of order $\mathcal{O}(N^2)$ on sparse networks.

InfoMap was proposed by Rosvall and Bergstrom [59], taking community detection as a problem of compressing information on a network's structure. The best compression is achieved by optimising a quality function that measures the code length describing random walks in the network (contrary to optimising modularity). The algorithm is of order $\mathcal{O}(M)$, M being the number of edges in the network.

Label Propagation was proposed by Raghavan et al. [55]. Inspired by message passing paradigms, it assumes that nodes likely share the community of their neighbors. It starts by labeling each node, and then a random order is created to determine the sequence of relabelling nodes. According to this sequence, each node has its label updated by the label of the majority of its neighbors. The sequential relabelling is repeated until nodes have the same label as the majority of their neighbors. The algorithm is of order $\mathcal{O}(M)$.

WalkTrap is a hierarchical clustering algorithm proposed by Pons and Latapy [50]. It is based on the fact that random walks stay longer within a community. It is of order $\mathcal{O}(N^2 \log N)$ in sparse networks.

Spin Glass method was proposed by Reichardt and Bornholdt [57]. It is based on finding the ground state of a spin glass model on the network [12], which reveals communities by spin clusters. A resolution parameter enables the method to find communities in several scales [30]. The algorithm is of order $\mathcal{O}(N^3)$.

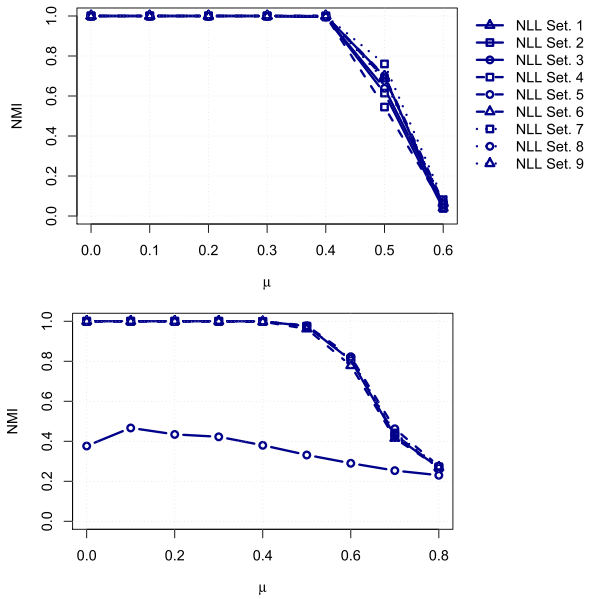

Fig. 6. Normalised Mutual Information (NMI) for 9 settings (i.e., algorithmic implementations) of the Node Linear-Logarithmic (NLL) [44] force model in $D = 3$ dimensions applied to 2 benchmark networks as their communities are mixed by μ. Top [Bottom] panel shows NMI values for Girvan-Newman [21] [Lancichinetti-Fortunato-Radichi [31]] networks with $N = 124$ nodes and 4 equal-sized communities [with $N = 1000$ nodes, $\gamma = 2$, $\beta = 1$, $\langle k \rangle = 20$, $\max(k) = 50$, $\min_C = 10$ and $\max_C = 50$] when using Partition Around Medoids (PAM) to the resultant force layouts. Clustering partitions are set according to the number of communities known from the benchmark and legends signal the settings from Table 1.

3. Results

Community detection from unsupervised FDA (i.e., without the presence of decision-makers) requires two phases: (i) a *layout phase*, where one chooses a force model and an algorithmic implementation (numerical simulation or optimisation technique) to achieve the steady-state – where attractive and repulsive forces cancel each other – and (ii) a *clustering phase*, where the community structure is extracted from the positions of the nodes in the D-dimensional layout.

Here, we quantify the accuracy of FDAs (from Sect. 2.2) and classical methods (from Sect. 2.5) on benchmark networks using unweighted, weighted, and hierarchical synthetic networks. In order to test different FDA implementations in the layout phase, we follow the 9 settings from Table 1 and change the space dimensions from $D = 3$ to $D = 10$. For the clustering phase, we use K-Means, Partition around Medoids (PAM), and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) on the resultant force model layout. Our numerical experiments consist of quantifying the similarity between the built-in structure of the benchmarks and the structure recovered after we make changes to both phases – implementation setting, dimensions, and clustering method – by means of the NMI [11]. As a result, we have a total of 54 accuracy tests per force model and benchmark network, which stem from having 2×9 options with regards to the implementations (dimension $\times$ settings) and 3 possible clustering methods.

3.1. Force Models depend on implementation

In order to illustrate how force models depend on the setting from Table 1 and clustering analysis, we focus here on the Node Linear-Logarithmic (NLL) model [44] applied to [21] (GN) and [31] (LFR) networks (with small communities), when using a $D = 3$ dimensional space.

Fig. 6 shows the resultant NMI values of the NLL model for all settings from Table 1 as the mixing parameter, μ, is increased. NMI results come from applying PAM to the layouts of GN networks (top panel) and the layouts of LFR networks (bottom panel). These clustering methods achieve the best NMI in their respective benchmark. We note that in GN networks (top panel in Fig. 6), the choice of the setting only affects the algorithm accuracy when $\mu > 0.4$, reaching the highest NMI (dotted line with squares) when using setting 7 (conjugate gradient approach [65,66] with 100 iterations). On the other hand, we note that accuracy is almost insensitive to the settings in LFR networks (bottom panel in Fig. 6), except for setting 4 (Fruchterman-Reingold Heuristic [20] with 100 iterations), which significantly decreases the resultant NMI.

These results show that different settings can affect accuracy. The NMI results in Fig. 6 also point to a limit on the gain in accuracy that a given FDA can reach by changing its settings. Consequently, in what follows, we only show the highest *NMI* curve obtained from testing all the settings in Table 1 for each force model; in the case of a tie, we choose the setting with the least number of iterations.

Table 1
Different Force-Directed Algorithm (FDA) implementations, classified into 9 settings (bottom row) according to the parameters and method used in this work. FDA models: Black-Hole (BH), Edge Linear-Logarithmic (ELL), Node Linear-Logarithmic (NLL), Fruchterman-Reingold (FR), and QMR. When the model can be adapted to a given setting, the table shows a tick, otherwise, a dashed line. Below each setting, we list lines and symbols used for plotting the corresponding results.

	Euler Method			Fruchterman-Reingold Heuristic			Conjugate Gradient		
	Step 0.1 It.: 10000	Step 1.0 It.: 100	Step 1.0 It.: 1000	Temp.: 1.0 It.: 100	Temp.: 1.0 It.: 1000	Temp.: 1.0 It.: 10000	It.: 100	It.: 1000	It.: 10000
BH	—	—	—	—	—	—	✓	✓	✓
ELL	✓	✓	✓	✓	✓	✓	✓	✓	✓
NLL	✓	✓	✓	✓	✓	✓	✓	✓	✓
FR	—	—	—	✓	✓	✓	✓	✓	✓
QMR	✓	✓	✓	✓	✓	✓	—	—	—
	Set. 1 —△—	Set. 2 —□—	Set. 3 —○—	Set. 4 – – □ – –	Set. 5 – – ○ – –	Set. 6 – –△ – –	Set. 7 - - -□- - -	Set. 8 - - -○- - -	Set. 9 - - -△- - -
	Settings								

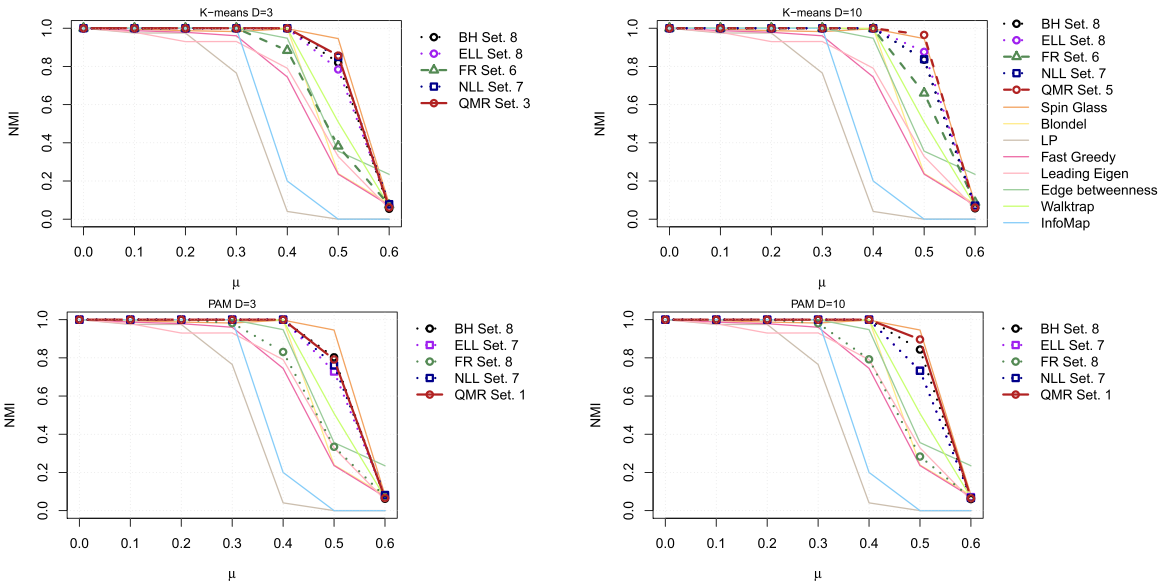

Fig. 7. Best NMI curves from 5 force models (lines with symbols as in Table 1) and 8 classical community detection methods (thin continuous lines) on Girvan-Newman networks [21] as communities are mixed. Force models include the Black-Hole [33] (BH), Edge Linear-Logarithmic [45] (ELL), Node Linear-Logarithmic [44] (NLL), Fruchterman-Reingold [20] (FR), and QMR [53]. The best curve for each force model is found from the NMI results obtained after testing all settings in Table 1, using K-means (top panels) or PAM (bottom panels) as a clustering method with 4 partitions ($K = 4$), and using a 3 (left panels) or a 10 (right panels) dimensional space. Each point is an average over 10 network realisations.

3.2. Benchmark Tests and Clustering Analysis: Force Models and Classical Methods

3.2.1. Unweighted Networks

In unweighted networks, edges represent relationships between nodes without poundage, i.e., without weight. Most community detection algorithms were developed for unweighted networks, which is by itself a challenging task [30]. Here, we present results on unweighted networks.

The best NMI curves for each FDA (lines with symbols) in comparison to the classical community detection methods NMI (thin continuous lines) on GN networks [21] are shown in Fig. 7. The best NMI value for any FDA at a given μ is found from the maximum NMI obtained for the different implementation settings of Table 1. The best setting is abbreviated next to the force model in Fig. 7 as a legend on the right of each panel. Top panels show NMI results using K-means clustering on the resultant layout, where on the left, the layout has 3 dimensions, and on the right, the layout has 10 dimensions. The bottom panels show similar NMI results but using PAM clustering on the resultant layout. Both clustering methods belong to the class of partitioning methods that require setting the expected number of communities, which we set to 4.

NMI values for the classical detection methods are the same for all panels in Fig. 7, where the Spin Glass [12,57] is consistently the most accurate. We also note from Fig. 7, that even the oldest FDA – FR [20] – achieves equivalent NMI values to the best classical detection methods when $D = 10$ (right panels) and K-means (top panels) is used. Overall, we find an increase in accuracy for all FDAs when $D = 10$, and we use K-means to find the communities from the layout. However, the QMR model [53] in the 5-th setting of Table 1 outperforms all FDAs and classical detection methods (top right panel in Fig. 7).

We lack prior knowledge about the ideal number of partitions in most real-world applications. It means that real-world networks require estimations. Here, we test a naive strategy for estimating the communities using the Hartigan index [26] on the $D = 10$-dimensional layouts. Results from this strategy are shown in Fig. 8, where we see a slight loss in the FDA's accuracy. Despite this loss, our naive approach achieves higher NMI values for FDAs than most classical methods; except for the Spin Glass method [57], which requires previous knowledge of the expected number of communities when setting its parameters.

Figs. 9 and 10 show the best NMI-curves achieved for LFR networks with small ($\min_C = 10$ and $\max_C = 50$ nodes) and big ($\min_C = 20$ and $\max_C = 100$ nodes) communities, respectively (details in Sect. 2.1.1). Similarly to the results from Fig. 7 on GN networks, increasing D improves the accuracy of all force models – right panels in Figs. 9 and 10. But the choice of clustering barely impacts on the resultant accuracy (comparing top panels with bottom panels in Figs. 9 and 10), with a slight advantage of PAM over K-means (contrary to the GN results from Fig. 7). Here, the ELL model [45] in its best setting (Set. 8, conjugate gradient) outperforms all other methods, closely followed by the QMR model [53], and then the BH [33] and Spin Glass models [12,57].

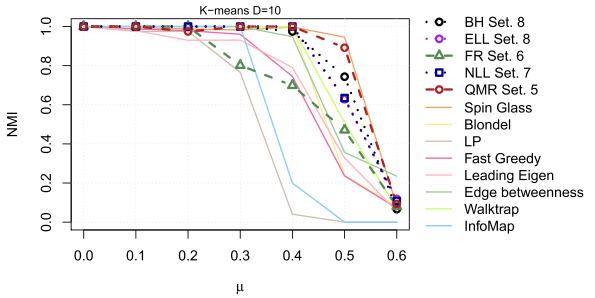

Fig. 8. NMI of classical community detection methods (thin continuous lines) and best NMI values obtained for 5 force models (lines with symbols as in Table 1) from the top right panel in Fig. 7. Here, K is estimated from applying K-means on each force-model resultant layout of the GN network, using the Hartigan index [26] to estimate K.

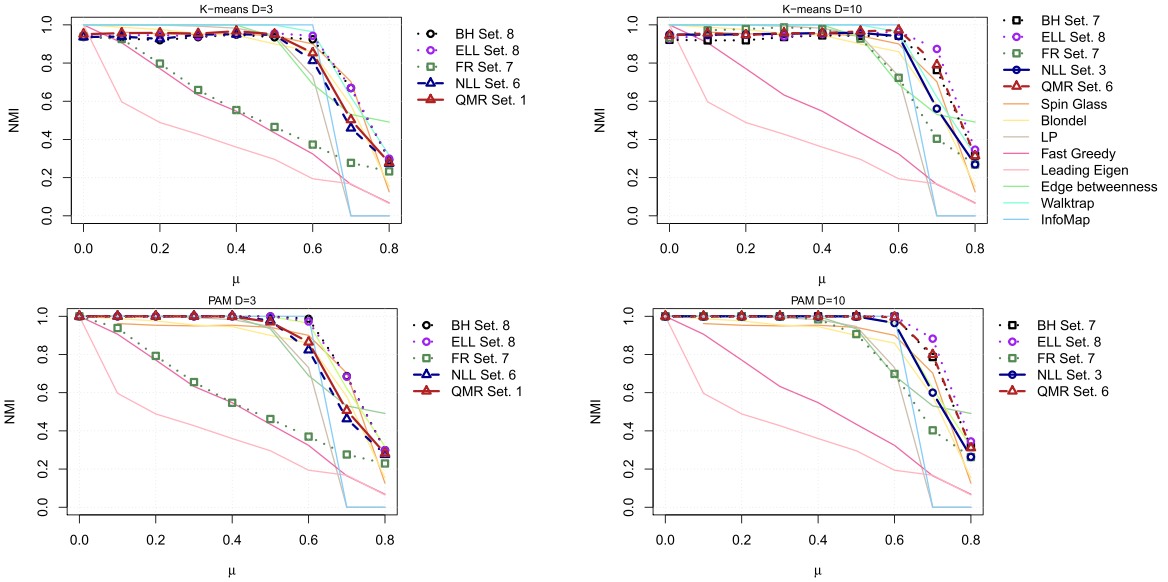

Fig. 9. NMI curves from 5 force models and 8 classical community detection methods (thin continuous lines) on Lancichinetti-Fortunato-Radichi networks [31] with small communities ($\min_C = 10$ and $\max_C = 50$ nodes, with $N = 1000$ nodes) as communities are mixed. Panels' layout and parameters are as in Fig. 7.

Next, we estimate the expected number of communities to use in PAM clustering for the LFR networks. We follow the same naive approach as with GN networks (Fig. 8); namely, we use the Hartigan index [26]. Fig. 11 shows the resultant NMI-curves for LFR networks with small (top panel) and big (bottom panel) communities, which correspond to the estimation of K for the bottom right panels in Figs. 9 and 10, respectively. This experiment notes that all force models lose accuracy – particularly around $\mu \simeq 0.5$ – but the ELL is only lightly affected, remaining the best FDA even when $\mu > 0.6$.

In order to finish the unweighted network analysis, we test whether we can improve FDA's accuracy in community detection by using DBSCAN [15], a density-based clustering that has been previously recommended for the BH force model [33]. We set the minimum number of points per cluster to $\min\{Pts\} = 2 \times D$ (D being the spatial dimension of the layout) and the threshold ϵ (defining the minimum distance at which two points can be classified as neighbours) by the K-distance graph heuristic proposed by Ester et al. [15]. From our tests, we can see that the accuracy from the force models decreases when using DBSCAN instead of K-means or PAM – even if we change $\min\{Pts\}$ and ϵ to improve results.

For example, Table 2 shows the NMI values we obtain for the ELL [45] and BH [33] force models in an LFR network realisation with small ($\min_C = 10$ and $\max_C = 50$ nodes) and unmixed ($\mu = 0$) communities. Because the communities are unmixed, any force model should be perfectly accurate, i.e., $NMI = 1$, or close to perfection. According to our results from Figs. 9 (bottom right panel) and 11 (top panel), ELL is the best algorithm for LFR networks with small communities when using PAM. However, from Table 2 we see that the resultant NMI values for both force models with DBSCAN are lower than those obtained from PAM. This happens even when having to estimate the number of partitions by the Hartigan index or increasing the layout dimensions to $D = 10$ (where $NMI = 0.99$ for DBSCAN in both models, but cannot surpass PAM with Hartigan index).

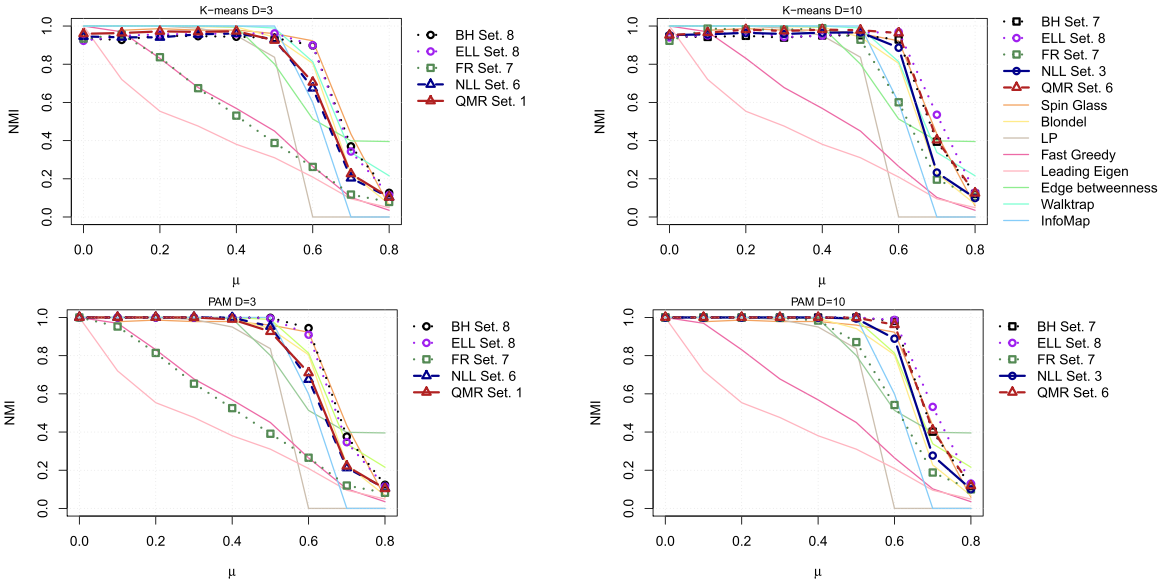

Fig. 10. NMI curves from the community detection methods from Fig. 9 on LFR networks with big communities ($\min_C = 20$ and $\max_C = 100$ nodes).

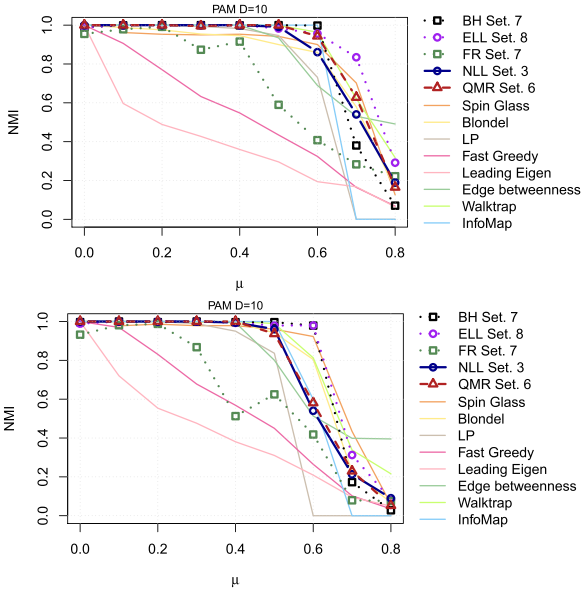

Fig. 11. NMI results from estimating the number of K partitions to use in PAM clustering of the LFR networks [31] and force models from the bottom right panels in Figs. 9 and 10. Estimation is carried as in Fig. 8.

Table 2
NMI values for the ELL [45] and BH [33] force models applied to an LFR network [31] with small and unmixed communities ($\mu = 0.0$) in a $D = 3$-dimensional layout.

	DBSCAN	PAM & Hartigan
ELL Set. 8	0.98	**1.0**
BH Set. 7	0.83	**0.99**

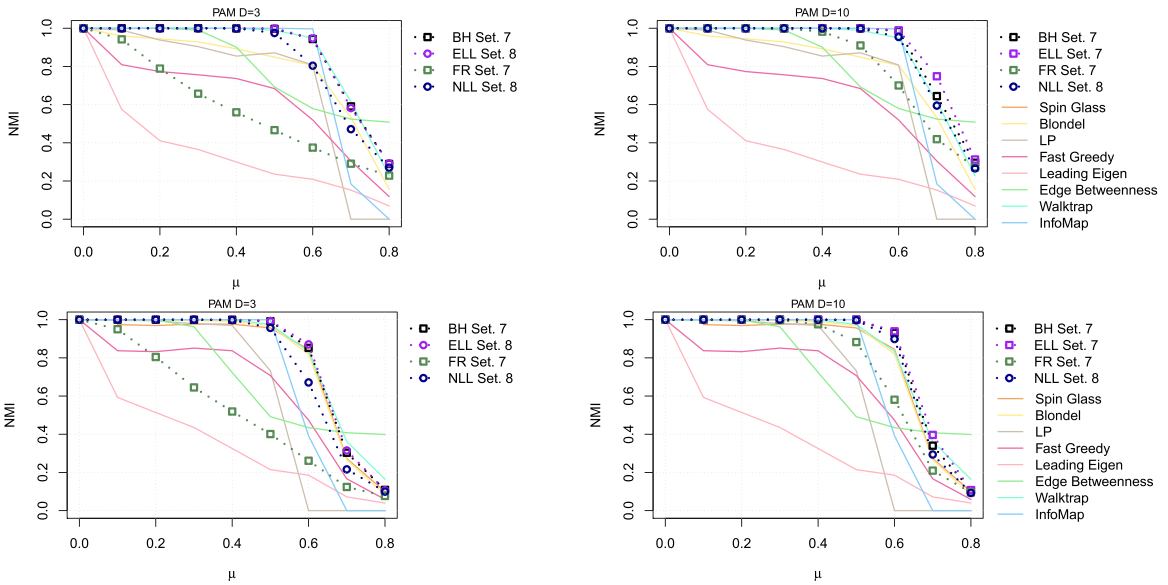

Fig. 12. NMI results from 4 force models (lines with symbols as in Table 1) with PAM clustering in $D = 3$-dimensional layouts [top] and $D = 10$-dimensional layouts [bottom] and 8 classical community detection methods (thin continuous lines) on LFR weighted-networks [30]. Top [Bottom] panels show results for networks with small [big] communities as they are increasingly mixed.

3.2.2. Weighted Networks

Here we present the accuracy tests for LFR weighted (LFRw) networks [30] using the 8 classical community detection methods and 4 of the force models – FR [20], NLL [44], ELL [45], and BH [33]. We exclude the QMR[53] because it lacks an extension to deal with weighted networks.

Our results on unweighted LFR networks show that PAM clustering achieves the highest community-detection accuracy for all force models. We corroborate that this is also the case for LFRw networks, obtaining better results with PAM than with K-means. Moreover, we find that force models with Euler or Fruchterman-Reingold Heuristic settings (detailed in Table 1) perform poorly in LFRw networks, where our results show that the Conjugate Gradient Approach is the only setting that provides accurate community detection results.

Fig. 12 shows the best NMI achieved in our experiments of LFRw networks with small ($\min_C = 10$ and $\max_C = 50$ nodes; top panels) and big ($\min_C = 20$ and $\max_C = 100$ nodes; bottom panels) communities, using a $D = 3$ (left panels) or 10 (right panels) dimensional layout. We note that the ELL and BH models have similar NMI values for all D and LFRw networks, but are higher than the NLL force model – particularly for $\mu > 0.5$. We also note that the effect of increasing D is significant for the NLL and FR methods, although, as expected, the FR is the least accurate force model. Similarly to our findings on LFR unweighted networks (Figs. 9 and 10), here, we show that WalkTrap [50], InfoMap [59], and Spin Glass [57] are the most accurate classical methods for small (top panels) and big (bottom panels) communities, and that the Leading Eigenvalue [40] is the most inaccurate method. Importantly, we find that when $D = 10$, the ELL and BH force models are the most accurate methods.

Now, we estimate the number of partitions to use for the PAM clustering analysis, following the same approach as in Figs. 8 and 11 for the GN and LFR unweighted networks, respectively. Fig. 13 shows the resultant NMI values of doing this estimation. We note that the accuracy of the force models rivals that of the best classical methods. Consequently, we can expect accurate results under real-world networks where the number of communities is unknown.

3.2.3. Hierarchical Organization

Real-world networks are far from being random graphs. In particular, they show a high degree of clustering and scale-free behaviour in node degrees and community sizes. The coexistence of these characteristics is left aside in most network models, typically being explained only when nodes can be organised hierarchically [56]. In general, detecting communities in hierarchical networks plays a central role in understanding complex systems [69]. However, community detection in hierarchical networks is virtually unexplored. Here, we use FDA to show that they can also be used to detect communities in hierarchical networks. In particular, we use a network example described by Noack [42], aiming to demonstrate the applicability of the FDA in revealing the network communities and hierarchies. Our results also highlight the weaknesses appearing in some classical methods when dealing with hierarchies.

We note from Fig. 14 that the hierarchical structure of the communities is lost when using the FR [20] model (left panel), but is directly appreciated from the ELL [45] model (right panel). Both force models place nodes at different locations, avoiding having nodes at the same locations, contrary to the BH [33] and QMR[53] models. We highlight that overlapping

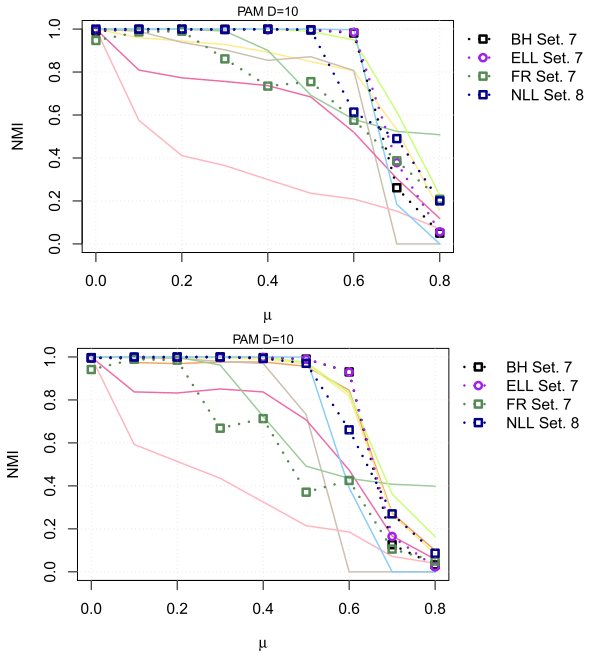

Fig. 13. NMI values obtained after estimating the number of partitions for PAM clustering by the Hartigan index applied to the force models in the right panels of Fig. 12.

Table 3
NMI values for the Hierarchical graph in Fig. 14.

	Macro	Micro
BH Set. 7	1	1
ELL Set. 7	1	1
NLL Set. 7	1	1
QMR Set. 5	1	1
FR Set. 7	1	0.54
Blondel	1	0.61
Spinglass	0.07	0.16

Fig. 14. Graph layout for a hierarchical network [42] when using Frucherman-Reingold (left layout) or the Edge Linear-Logarithmic (right layout) force model. The network contains 16 cliques of 50 nodes each, distributed equally among 4 communities.

nodes from the same community may improve community detection since it aids the detection of groups in the clustering phase.

Here, we implement a hierarchical clustering method to organise the layout data into a tree with varying levels of granularity. We perform this clustering by agglomeration (implementing the `hclust` function from R), merging groups according to their centroid. The tree is then pruned to reveal the communities hierarchically, which for Noack's network [42], it implies detecting 16 communities at the microscopic level (micro) and 4 communities at the macroscopic level (macro).

Our findings for the 5 force models in this hierarchical network are shown in Table 3, which also includes NMI values for 2 classical methods: Blondel [3] and Spin Glass [57], which have been previously used to reveal hierarchical community structures [69]. Except for the FR model, we note that all force models perfectly detect the community structure at both levels. On the other hand, these results show that classical community detection methods struggle to detect hierarchies accurately.

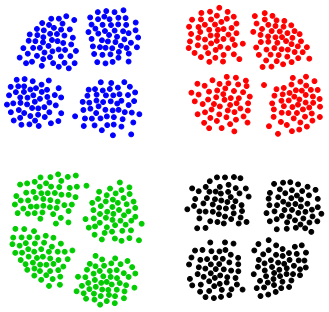

Fig. 15. Communities found by Blondel's method [3], in a graphical layout obtained by the Edge Linear-Logarithmic model [45], as in Fig. 14.

For example, Fig. 15 illustrates the NMI results in Table 3 for Blondel's method, where the ELL model is used to draw the graph layout as in Fig. 14. The different colours signal the communities that Blondel's method detects globally (macro), but cannot reveal the substructures (micro).

3.2.4. Zachary's Karate Club

Zachary's Karate Club has become a popular network for validating community detection algorithms; firstly used with this purpose by Girvan and Newman [21]. Such a network is known for catching the disagreement between the administrator and instructor of the club from the rupture of its members when the instructor left and started its own karate club. Here, we use the member declarations – described by Zachary [71] – about which club they attended after the instructor created a new club as ground truth to evaluate the communities detected by the FDAs.

The ELL (in Set. 7), NLL (in Set. 7), and QMR (in Set. 2) force models hold an NMI $\cong 0.8$ for the Karate club network, where only node 3 is classified incorrectly – this result has also been observed by Girvan and Newman [21]. We note that the NMI is unchanged when using K-Means or PAM with the number of communities set to $K = 2$, or when changing the dimensional space from $D = 3$ to $D = 10$. The FR (in Set. 7) force model also presents an NMI $\cong 0.8$ due to an incorrect classification of one node. However, in contrast to the former models, FR classifies wrongly node 20 when using K-Means but classifies wrongly node 3 with PAM. On the other hand, the BH force model (in Set. 7) has an NMI $\cong 0.7$, classifying incorrectly nodes 3 and 10 by all analysed clustering techniques (i.e., K-means, PAM, and DBSCAN). As noted in the previous network analyses, DBSCAN does not present advantages over K-Means or PAM.

When estimating the number of communities by the Hartigan index, we noted that the layout of force models indicates: $k = 3$ for models ELL, NLL, and QMR, $k = 5$ for the FR model, and $k = 15$ for the BH model. Fig. 16 shows the Zachary's Karate Club layout of ELL [FR] in the top [bottom]. The layout of the force model BH was omitted due to the fact that it is unlikely the existence of 15 communities in such a network.

4. Discussion and Future Works

Gouvêa et al. [22] shows that up-to-date, there are at least 27 published works that tackle community detection problems employing a Force-Directed Algorithm (FDA). 10 (37%) of these works lack Clustering Analysis (CA), using instead visual inspection to decide which structures are communities. The remainder 17 works replace human beings with CA methods, achieving automation. Moreover, most are unconnected. There are 11 articles that cite one of Noack's works [42,45,47], but barely cite each other; plus, another 10 which have no citation to any of the 27 works. Overall, this shows significant fragmentation in FDA literature, leading to a lack in comparative analyses, which we summarise in Table A.4. Our present work is the first to quantitatively compare 5 different FDAs under different network benchmarks, clustering analyses, and algorithmic implementations, as well as compare these results with classical community detection methods.

Besides the 5 force models taken into account here, at least 6 other force models are proposed for community detection. Among them, the $(1, -1)$-energy [67], $(1, -2)$-energy [10], and $(2, -2)$-energy [6] models do not provide the topological characteristics recommended by Noack [47], but they showed good results in their authors applications. Hence, their use should be restricted to cases where specific visual aesthetics are sought [22]. The remaining 3 models cannot be described by an (a, r)-energy model, so they also lack a topological interpretation of the resultant distances between nodes [22].

4.1. Force-Directed Algorithm implementation

The main drawback in using FDAs to detect communities in networks is the significant number of options that compose the decision process to obtain the layout: selecting a force model and then finding the force model's stable steady-state. It may lead to different algorithmic implementations (and their settings), changing the resultant layouts (due to local minima). These limitations and implementation effects have been virtually unexplored up to now.

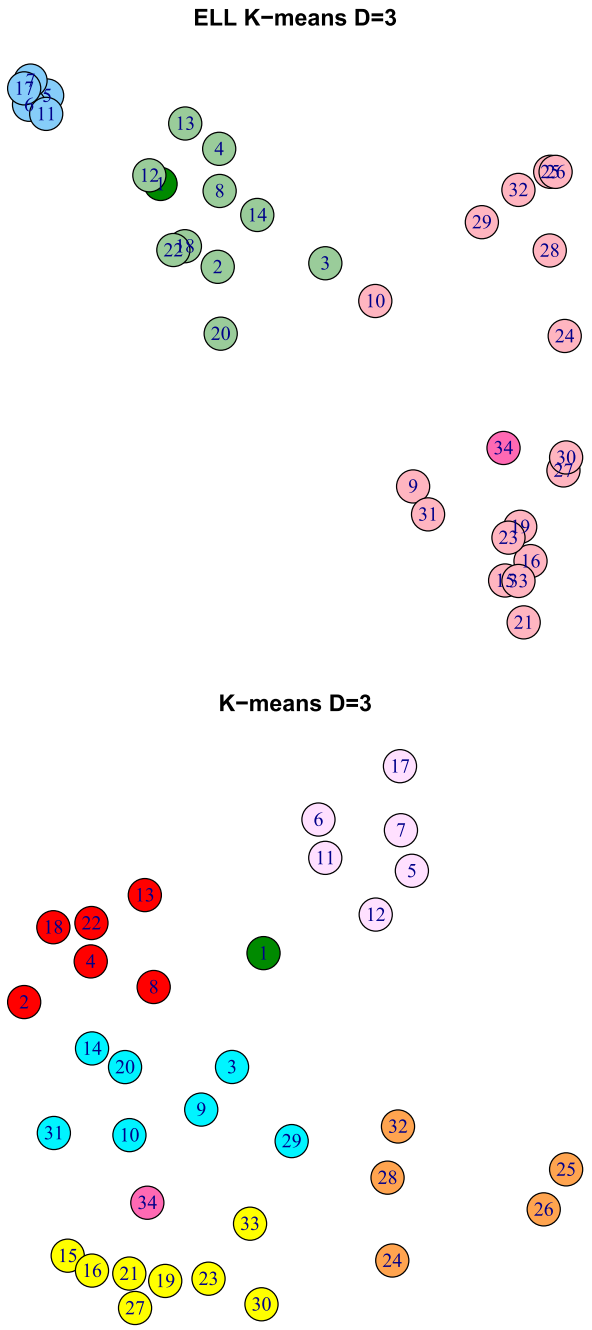

Fig. 16. Layout of Zachary's Karate Club network according to the ELL (top panel) and FR (bottom panel) models. Colors illustrate the communities detected from each model, where node 1 and 34 (highlighted) are club's instructor and administrator, respectively. The third community is formed by nodes 5, 7, 6, 11, and 17, and it was already observed by other community detection techniques. We are unaware of the detection of 5 communities, likely an untrustworthy structure.

We start bridging this gap by testing different settings (Table 1) on 5 force models (Sect. 2.2): Fruchterman-Reingold (FR) [20], Node [44] and Edge [45] Linear-Logarithmic (NLL and ELL), Black-Hole (BH) [33], and QMR [53]. In particular, we analyse the Euler method (numerical solution to the force model's evolution towards its steady-state), the Fruchterman-Reingold Heuristic (FRH), and a Conjugate Gradient Approach (GDA) (optimisation techniques to minimise the energy landscape of the force model).

Our findings show that different implementations affect the resultant accuracy of FDA community detection in network benchmarks (see Sect. 3). However, throughout our experiments, it was evident that by tuning the settings, we can improve

Table A1
Previous works dealing with force-directed algorithms that perform comparative studies with other force models.

	Compare a given FDA to		Benchmarks		
Study	Classical Methods	Other FDA	Real-World	Synthetic	Clustering
[33]	Blondel Infomap Label Propagation Spectral Clustering	Frucherman Reingold Model	DBLP Amazon IMDb Youtube Skitter	LFR	DBSCAN
[37]	GN Fast Greedy EO [14] MM [41]	—	Zachary's Karate Club Dolphins American Football Ring Jazz	—	Similar to PAM
[62]	GN Walktrap InfoMap	—	Zachary's Karate Club American Football Email URV	—	Similar to K-Means [53]
	GN Fast Greedy InfoMap Walktrap	—	American Football	GN LFR Hierarchical	Similar to PAM
[68]	GN [39] method	—	Zachary's Karate Club U.S. College Football	—	Proposed a distance-based approach

the FDA's accuracy independently of each implementation technique (i.e., Euler method, FRH, or GDA) and, in some cases, surpass the best classical community detection methods, such as the WalkTrap [50], Blondel [3], InfoMap [59], or Spin Glass [57] methods. At first, our experiments suggest that the best implementation (i.e., the choice of a numerical solution or the selection of an optimization technique) should be defined by the user knowledge domain.

4.2. Force-Directed Algorithm layout

Force models are responsible for highlighting topological characteristics of interest through the graph layout aesthetics [65]. A layout is the spatial distribution of nodes (particles) into the D-dimensional space set by the model; that is, the stable steady-state. In the context of community detection, force models should result in layouts where all nodes from a given community are close to each other and far from other nodes and communities. We note that seeking such a layout is different from maximising the distance between non-adjacent nodes since non-adjacent nodes can belong to the same community – even adjacent nodes can belong to different communities.

To the best of our knowledge, Noack's works on community detection [42,46] were the first to set formal requirements on the layout's spatial distribution of nodes. These works qualify good layouts as those that place densely intra-connected subsets of nodes at a distance proportional to the number of edges inter-connecting the subsets. This is achieved by Noack's NLL and ELL models [43–45], which are related to modularity optimisation methods [47]. However, these models – as the FR [66] model – also place nodes in different spatial locations even within a community, as depicted in the 3 top panels in Fig. 17.

Having a layout where nodes must be placed at different locations is an unnecessary restriction for community detection since communities can be detected by Clustering Analysis even when nodes are placed in the same position – as long as different subsets of nodes are clearly separated – as it can be seen from the bottom panels in Fig. 17 for the BH [33] and QMR[53] models.

Nevertheless, we find that Noack's ELL [45] model achieves the best accuracy from all force models and classical methods when applied to Lancichinetti-Fortunato-Radichi (LFR) [30] unweighted networks with small and big communities (Figs. 9 and 10), closely followed by the QMR[53] and BH [33] models. Moreover, the ELL model remains the most accurate in weighted LFR networks (Fig. 12). Furthermore, we have also tested (not shown) whether we can improve the ELL accuracy even further by making the repulsive forces act only over non-adjacent nodes, but our results remain unchanged.

4.3. Computational Complexity

Regardless of the chosen technique (i.e., whether a numerical integration or an optimization scheme) to implement the force models, the process to reach a stable steady-state is of order $\mathcal{O}(c \times N^2)$, c being the number of iterations (cycles) needed to move the particles until a stable steady-state is reached. The order $\mathcal{O}(c \times N^2)$ stems from the computation of the repulsion term. Force models like FR, ELL, NLL, and BH require computing the repulsion between all node pairs (N^2). The QMR model performs this calculation only in non-adjacent nodes, but as the networks are usually sparse, the computation of the repulsion term tends to be computed for almost all nodes.

Fortunately, the computation of the repulsion term can be approximated because of the presence of round-off errors, truncation, and discretization effects. Hence, the process to reach a stable steady-state is reduced to $\mathcal{O}(c \times (M + N\log(N))$ [46,51,66]. The idea of approximating the repulsion terms was proposed by Barnes and Hut [2] to solve N-body problems, and it is widely used in Graph Visualization Literature Lim [33].

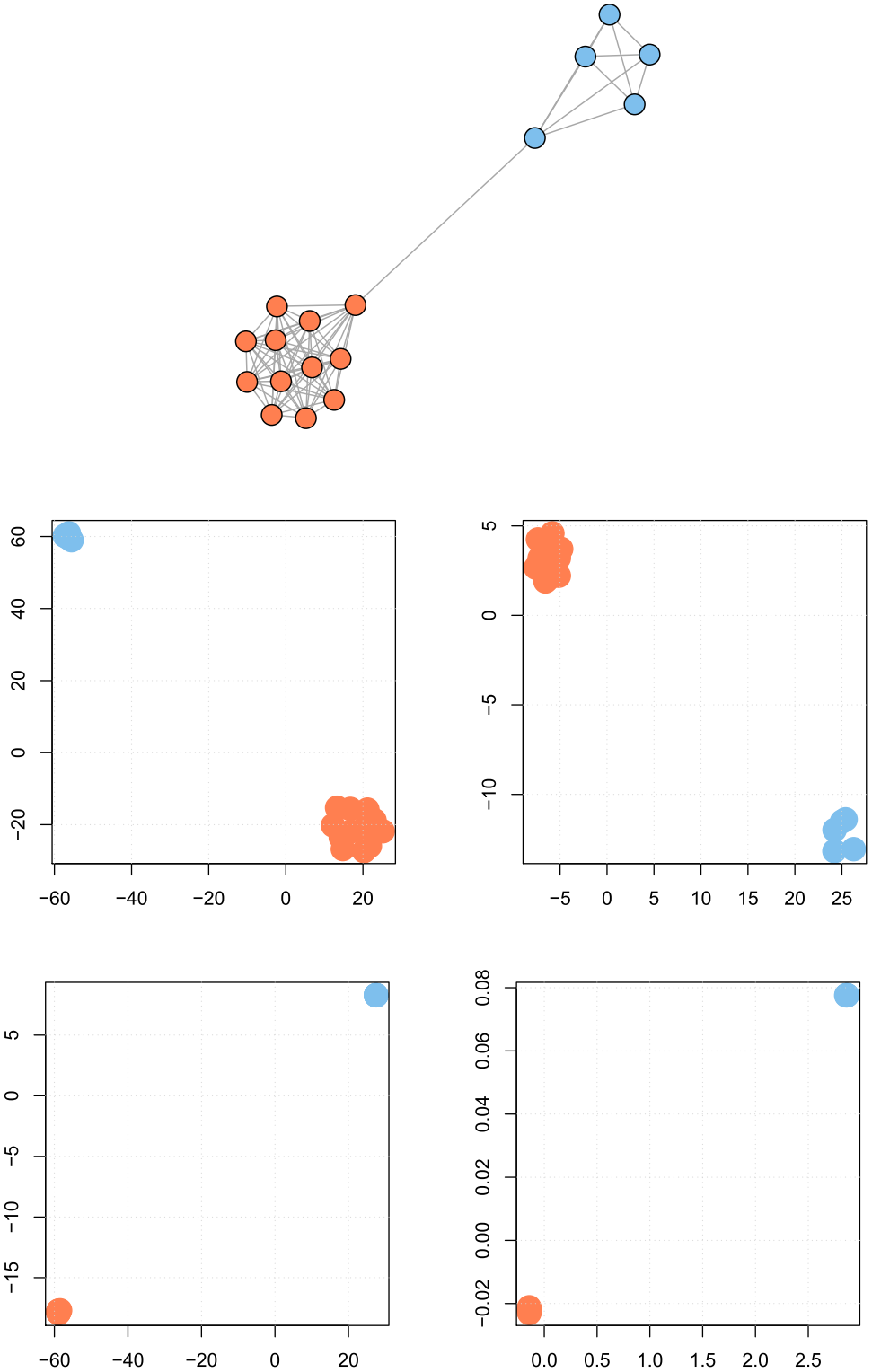

Fig. 17. Force models applied to a network with $N = 17$ nodes and 2 communities (with 5 and 12 nodes inter-connected by a single edge). Panels show two-dimensional layouts resultant from applying Frucherman-Reingold [20] (top), Edge LinLog (middle left) [45], Node LinLog (middle right) [44], Black-Hole (bottom left) [33], or QMR [53] (bottom right) models.

Although our work tested force models on networks of size $\mathcal{O}(N) \sim 10^3$, [33] describe experiments with the BH force model on real-world networks with millions of nodes and edges. Furthermore, other works, such as those by Yang and Liu [68] or Coleman and Parker [9], claim that a layout can be achieved from quick algorithms whose computational complexity is $\mathcal{O}(N)$. Here, we highlight that more work is needed in order to verify if approximations that provide layouts with complexity $\mathcal{O}(c \times N)$ are adequate to detect communities. Furthermore, to date, no studies estimate the value of c in the context of community detection. In Graph Visualization, it is expected that c grows linearly with N [45,66]. Future work could be focused on analysing the growth rate of c as a function of N in the community detection framework.

It is worth noting that FDAs can also be used as an embedding [33]. Graph embedding algorithms usually require multiple free parameter settings, which are known for their high computational cost [64]. Here, we show that ELL, NLL, and BH require the parameters involved with implementing the force model, which are summarized by the number of cycles (i.e., c) when the gradient conjugate approach is used. So, by verifying if a layout may be achieved in $\mathcal{O}(c \times N)$ and concluding that c may be seen as a constant of the growth rate N, it means introducing an efficient graph embedding technique. However, it should be highlighted that in detecting communities without visual human inspection, the final computational cost will always depend on the clustering technique chosen by the user. This potential drawback has been recently discussed by Tandon et al. [64].

4.4. Clustering Analysis

We evaluate the accuracy in detecting communities by force models using 2 distance-based methods – K-means and Partition Around Medoids (PAM) – and 1 density-based method – Density-Based Spatial Clustering of Applications with Noise (DBSCAN). Our work is the first comparative study on clustering analysis effects on community detection to the best of our knowledge. Our results show that different clustering methods can improve the detection accuracy depending on the benchmark.

K-means and PAM are partitioning methods for clustering analysis, which are known to work well when points are located in spherically-shaped clusters. However, these methods are inefficient when clusters have complex shapes, such as "S" shapes or oval clusters [25] (Fig. 5). We find that K-means is the best clustering method to partition the resultant layouts from applying force models to unweighted Girvan-Newman [21] (GN) networks (Fig. 7), and PAM is the best when it comes to partitioning layouts obtained from unweighted LFR [30] networks (Figs. 9, 10, and 12).

In contrast to partitioning methods, density-based methods can find clusters of arbitrary shapes, such as DBSCAN; but they are sensitive to input parameters. Our findings show that DBSCAN gets less accurate results than PAM (even when estimating the number of partitions by means of the Hartigan index) in LFR unweighted networks (Table 2)[1]. This result is somehow unexpected for the BH model [33] since the model was proposed using DBSCAN.

Overall, our results indicate that GN [21] and LFR [30] networks have community structures that appear to organise themselves into layouts with spherically-shaped clusters. Possibly, these spherical clusters have different densities and sizes, which would explain why DBSCAN produces less accurate results than the partitioning methods. However, research on the possible shapes that network communities can take in a force-model layout is missing – such research could clarify our results and help to improve the choice of clustering analysis.

5. Conclusion

Community detection has been intensively pursued in the 21st century [17], where a community is broadly defined as a set of nodes that share a higher link density (inter-connections) than the network's link density and that have a lower link density (intra-connections) with nodes outside their set. Thus, finding a community is a complex (NP) combinatorial problem with many possible solutions. The task becomes even more challenging if the search must take into account that real-world communities can have weights, exhibit a hierarchical structure, overlap (with nodes participating in different communities), and change in time.

Force models – under certain parameters and conditions – produce graphical layouts that are consistent with modularity [47]. Hence, force models are expected to be as efficient in detecting network communities as methods based on modularity. Also, force-based models may be seen as complementary to the classical community-detection approach since force models aid in network visualisation, which can be used to visually evaluate the quality of the detected communities by any method.

In the context of methods based on modularity optimisation, the force-directed approach has the advantage of being insensitive to the resolution limit. The resolution limit is an inherent characteristic of all methods based on modularity optimisation, stemming from its mathematical formulation. Namely, methods based on modularity optimisation cannot detect communities smaller than a given scale – estimated from the total size of the network and the degree of inter-connectedness of the modules – even when the community structure is unambiguous [18].

In summary, we show that Force-Directed Algorithms (FDA) can accurately detect communities and compete with the best classical community detection methods. Our findings are supported by testing 5 force models, changing their algorithmic implementations – we follow the 9 settings from Table 1 and use $D = 3$ to $D = 10$ for the layout dimensions – and

[1] We omitted it to avoid redundancy, but the same results were observed in LFR weighted networks.

clustering analysis of the resultant layout – where we use *K*-Means, Partition around Medoids, and Density-Based Spatial Clustering of Applications with Noise. Our numerical experiments consist of quantifying the similarity between the built-in structure of a benchmark network and the structure recovered by the FDA and comparing it with that of 8 classical community detection methods. We report a total of 54 tests per FDA and benchmark network, helping us to conclude that: 1) FDAs can be more accurate than classical community detection methods, 2) FDAs depend on the algorithmic implementation, where typically optimisation techniques achieve better results than numerical integration, 3) higher-dimensional layouts improve accuracy, and 4) distance-based clustering improves results.

CRediT authorship contribution statement

Alessandra M.M.M. Gouvêa: Formal analysis, Visualization, Writing – original draft. **Nicolás Rubido:** Writing – review & editing, Validation. **Elbert E.N. Macau:** Funding acquisition. **Marcos G. Quiles:** Conceptualization, Funding acquisition, Methodology, Supervision, Validation.

Acknowledgment

A.M.M.M.G. and M.G.Q acknowledge the support of the São Paulo Research Foundation (FAPESP), Brazil, Proc. 2015/50122-0, 2016/23642-6, 2016/23698-1, 2016/16291-2, 2017/05831-9, 2019/26283-5, and 2019/00157-3; and the National Council for Scientific and Technological Development (CNPq), Brazil, Proc. 434886/2018-1 and 313426/2018-0. N.R. acknowledges the Comisión Sectorial de Investigación Científica (CSIC), Uruguay, group grant “CSIC2018 - FID 13 - Grupo ID 722”.

Appendix A. Comparative Studies on Force Models

Table A.4 shows previous works dealing with force-directed algorithms that perform comparative studies with other force mode.

References

[1] C.C. Aggarwal, C.K. Reddy, Data clustering, Algorithms and applications. Chapman&Hall/CRC Data mining and Knowledge Discovery series, Londra (2014).
[2] J. Barnes, P. Hut, A hierarchical o (n log n) force-calculation algorithm, Nature 324 (1986) 446–449, doi:10.1038/324446a0.
[3] V.D. Blondel, J.L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, Journal of Statistical Mechanics: Theory and Experiment 2008 (2008) P10008.
[4] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, D.U. Hwang, Complex networks: Structure and dynamics, Physics Reports 424 (2006) 175–308, doi:10.1016/j.physrep.2005.10.009.
[5] U. Brandes, Drawing on physical analogies, in: Drawing graphs, Springer, 2001, pp. 71–86.
[6] Y. Cai, J.A. Morales, S. Wang, P. Pimentel, W. Casey, A. Volkmann, Pheromone model based visualization of malware distribution networks, in: International Conference on Computational Science, Springer, 2018, pp. 55–68.
[7] S.H. Cheong, Y.W. Si, Force-directed algorithms for schematic drawings and placement: A survey, Information Visualization 19 (2020) 65–91, doi:10.1177/1473871618821740.
[8] A. Clauset, M.E. Newman, C. Moore, Finding community structure in very large networks, Physical review E 70 (2004) 066111, doi:10.1103/PhysRevE.70.066111.
[9] M.K. Coleman, D.S. Parker, Aesthetics-based graph layout for human consumption, Software: Practice and Experience 26 (1996) 1415–1438, doi:10.1002/(SICI)1097-024X(199612)26:12<1415::AID-SPE69>3.0.CO;2-P.
[10] A. Crippa, L. Cerliani, L. Nanetti, J.B. Roerdink, Heuristics for connectivity-based brain parcellation of sma/pre-sma through force-directed graph layout, NeuroImage 54 (2011) 2176–2184, doi:10.1016/j.neuroimage.2010.09.075.
[11] L. Danon, A. Diaz-Guilera, J. Duch, A. Arenas, Comparing community structure identification, Journal of Statistical Mechanics: Theory and Experiment 2005 (2005) P09008, doi:10.1088/1742-5468/2005/09/p09008.
[12] V.L. Dao, C. Bothorel, P. Lenca, Community structure: A comparative evaluation of community detection methods, Network Science 8 (2020) 1–41, doi:10.1017/nws.2019.59.
[13] R. Davidson, D. Harel, Drawing graphs nicely using simulated annealing, ACM Transactions on Graphics 15 (1996) 301–331, doi:10.1145/234535.234538.
[14] J. Duch, A. Arenas, Community detection in complex networks using extremal optimization, Physical review E 72 (2005) 027104, doi:10.1103/PhysRevE.72.027104.
[15] M. Ester, H.P. Kriegel, J. Sander, X. Xu, et al., A density-based algorithm for discovering clusters in large spatial databases with noise, in: KDD-96 Proceedings, 1996, pp. 226–231.
[16] B.S. Everitt, S. Landau, M. Leese, D. Stahl, Cluster analysis 5th ed, 2011, (????).
[17] S. Fortunato, Community detection in graphs, Physics Reports 486 (2010) 75–174, doi:10.1016/j.physrep.2009.11.002.
[18] S. Fortunato, M. Barthelemy, Resolution limit in community detection, Proceedings of the National Academy of Sciences 104 (2007) 36–41, doi:10.1073/pnas.060596510.
[19] S. Fortunato, D. Hric, Community detection in networks: A user guide, Physics Reports 659 (2016) 1–44, doi:10.1016/j.physrep.2016.09.002.
[20] T.M. Fruchterman, E.M. Reingold, Graph drawing by force-directed placement, Software: Practice and Experience 21 (1991) 1129–1164, doi:10.1002/spe.4380211102.
[21] M. Girvan, M.E. Newman, Community structure in social and biological networks, Proceedings of the National Academy of Sciences 99 (2002) 7821–7826, doi:10.1073/pnas.122653799.
[22] M.M.M.A. Gouvêa, S.T. Silva, E.E.N. Macau, M. Quiles, Force-directed algorithms as a tool to supportcommunity detection: A review, The European Physical Journal Special Topics 19 (2020) 65–91, doi:10.1140/epjs/s11734-021-00167-0.
[23] C. Granell, R.K. Darst, A. Arenas, S. Fortunato, S. Gómez, Benchmark model to assess community structure in evolving networks, Physical Review E 92 (2015) 012805, doi:10.1103/PhysRevE.92.012805.
[24] M. Gupta, C.C. Aggarwal, J. Han, Y. Sun, Evolutionary clustering and analysis of bibliographic networks, in: 2011 International Conference on Advances in Social Networks Analysis and Mining, IEEE,63–70, 2011.
[25] J. Han, M. Kamber, J. Pei, Data mining concepts and techniques third edition, The Morgan Kaufmann Series in Data Management Systems 5 (2011) 83–124.

[26] J.A. Hartigan, Clustering algorithms, John Wiley & Sons, Inc., 1975.
[27] M.A. Javed, M.S. Younis, S. Latif, J. Qadir, A. Baig, Community detection in networks: A multidisciplinary review, Journal of Network and Computer Applications 108 (2018) 87–111, doi:10.1016/j.jnca.2018.02.011.
[28] S. Jiang, X. Li, X. Chen, Z. Wang, M. Perc, C. Gao, Multi-objective optimization for community detection in multilayer networks, Europhysics Letters 135 (2021) 18001.
[29] M. Kaufmann, D. Wagner, Drawing graphs: methods and models, Springer, 2003.
[30] A. Lancichinetti, S. Fortunato, Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities, Physical Review E 80 (2009) 016118, doi:10.1103/PhysRevE.80.016118.
[31] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, Physical review E 78 (2008) 046110, doi:10.1103/PhysRevE.78.046110.
[32] H.J. Li, L. Wang, Y. Zhang, M. Perc, Optimization of identifiability for efficient community detection, New Journal of Physics 22 (2020) 063035.
[33] S. Lim, J. Kim, J.G. Lee, Blackhole: Robust community detection inspired by graph drawing, in: IEEE 32nd International Conference on Data Engineering (ICDE), IEEE, 2016, pp. 25–36.
[34] T. Liu, D.B. Ahmed, F. Bouali, G. Venturini, Visual and interactive exploration of a large collection of open datasets, in: 17th International Conference on Information Visualisation, IEEE,285–290, 2013.
[35] R.D. Luce, A.D. Perry, A method of matrix analysis of group structure, Psychometrika 14 (1949) 95–116, doi:10.1007/BF02289146.
[36] D.M. Maia, J.E. de Oliveira, M.G. Quiles, E.E. Macau, Community detection in complex networks via adapted kuramoto dynamics, Communications in Nonlinear Science and Numerical Simulation 53 (2017) 130–141, doi:10.1016/j.cnsns.2017.05.002.
[37] P.J. McSweeney, K. Mehrotra, J.C. Oh, A force-directed layout for community detection with automatic clusterization, Simulating Interacting Agents and Social Phenomena, Springer,49–63, 2010.
[38] E.M. Mohamed, T. Agouti, A. Tikniouine, M. El Adnani, A comprehensive literature review on community detection: Approaches and applications, Procedia Computer Science 151 (2019) 295–302, doi:10.1016/j.procs.2019.04.042.
[39] M.E. Newman, Fast algorithm for detecting community structure in networks, Physical review E 69 (2004) 066133, doi:10.1103/PhysRevE.69.066133.
[40] M.E. Newman, Finding community structure in networks using the eigenvectors of matrices, Physical review E 74 (2006) 036104, doi:10.1103/PhysRevE.74.036104.
[41] M.E. Newman, Modularity and community structure in networks, Proceedings of the National Academy of Sciences 103 (2006) 8577–8582, doi:10.1073/pnas.0601602103.
[42] A. Noack, An energy model for visual graph clustering, International Symposium on Graph Drawing, Springer,425–436, 2003.
[43] A. Noack, Energy models for drawing clustered small-world graphs, Technical Report. Fachgebiet Praktische Informatik., 2004.
[44] A. Noack, Energy-based clustering of graphs with nonuniform degrees, in: International Symposium on Graph Drawing, Springer, 2005, pp. 309–320.
[45] A. Noack, Energy models for graph clustering, J. Graph Algorithms Appl. 11 (2007) 453–480.
[46] A. Noack, Unified quality measures for clusterings, layouts, and orderings of graphs, and their application as software design criteria, Ph.D. thesis. Naturwissenschaften und Informatik der Brandenburgischen Technischen Universitt Cottbus, 2007.
[47] A. Noack, Modularity clustering is force-directed layout, Physical Review E 79 (2009) 026102, doi:10.1103/PhysRevE.79.026102.
[48] A. Palmer, O. Sinnen, Scheduling algorithm based on force directed clustering, in: 9th International Conference on Parallel and Distributed Computing, Applications and Technologies, IEEE,311–318, 2008.
[49] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.
[50] P. Pons, M. Latapy, Computing communities in large networks using random walks, International Symposium on Computer and Information Sciences, Springer,284–293, 2005.
[51] A.J. Quigley, Large scale relational information visualization, clustering, and abstraction Ph.D. thesis. University of Newcastle (2001).
[52] M. Quiles, Particle community: A dynamical model for detecting communities in complex networks, 2016, (????).
[53] M.G. Quiles, E.E. Macau, N. Rubido, Dynamical detection of network communities, Scientific Reports 6 (2016) 25570, doi:10.1038/srep25570.
[54] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, D. Parisi, Defining and identifying communities in networks, Proceedings of the National Academy of Sciences 101 (2004) 2658–2663, doi:10.1073/pnas.04000541.
[55] U.N. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in large-scale networks, Physical Review E 76 (2007) 036106, doi:10.1103/PhysRevE.76.036106.
[56] E. Ravasz, A.L. Barabási, Hierarchical organization in complex networks, Physical review E 67 (2003) 026112, doi:10.1103/PhysRevE.67.026112.
[57] J. Reichardt, S. Bornholdt, Statistical mechanics of community detection, Physical review E 74 (2006) 016110, doi:10.1103/PhysRevE.74.016110.
[58] G. Rossetti, R. Cazabet, Community discovery in dynamic networks: a survey, ACM Computing Surveys 51 (2018) 1–37, doi:10.1145/3172867.
[59] M. Rosvall, C.T. Bergstrom, Maps of random walks on complex networks reveal community structure, Proceedings of the National Academy of Sciences 105 (2008) 1118–1123, doi:10.1073/pnas.070685110.
[60] M. Rosvall, J.C. Delvenne, M.T. Schaub, R. Lambiotte, Different approaches to community detection, Advances in network clustering and blockmodeling (2019) 105–119.
[61] S.B. Seidman, B.L. Foster, A graph-theoretic generalization of the clique concept, Journal of Mathematical Sociology 6 (1978) 139–154, doi:10.1080/0022250X.1978.9989883.
[62] Y. Song, S. Bressan, Force-directed layout community detection, in: International Conference on Database and Expert Systems Applications, Springer,419–427, 2013.
[63] P.N. Tan, M. Steinbach, V. Kumar, Introduction to data mining, Pearson Education India, 2016.
[64] A. Tandon, A. Albeshri, V. Thayananthan, W. Alhalabi, F. Radicchi, S. Fortunato, Community detection in networks using graph embeddings, Physical Review E 103 (2021) 022316, doi:10.1103/PhysRevE.103.022316.
[65] D. Tunkelang, Jiggle: Java interactive graph layout environment, International Symposium on Graph Drawing, Springer,413–422, 1998.
[66] D. Tunkelang, D. Sleator, P. Heckbert, B. Maggs, A numerical optimization approach to general graph drawing, Ph.D. thesis. Carnegie Mellon's School of Computer Science (1999).
[67] M. Udrescu, L. Udrescu, A drug repurposing method based on drug-drug interaction networks and using energy model layouts, Computational Methods for Drug Repurposing, Springer,185–201, 2019.
[68] B. Yang, D.Y. Liu, Force-based incremental algorithm for mining community structure in dynamic network, Journal of Computer Science and Technology 21 (2006) 393–400, doi:10.1007/s11390-006-0393-1.
[69] Z. Yang, J.I. Perotti, C.J. Tessone, Hierarchical benchmark graphs for testing community detection algorithms, Physical review E 96 (2017) 052311, doi:10.1103/PhysRevE.96.052311.
[70] V. Zabiniako, Using force-based graph layout for clustering of relational data, in: East European Conference on Advances in Databases and Information Systems, Springer,193–201, 2009.
[71] W.W. Zachary, An information flow model for conflict and fission in small groups, Journal of Anthropological Research 33 (1977) 452–473, doi:10.1086/jar.33.4.3629752.