SCIENTIFIC REP**O**RTS

Dynamical detection of network communities

Marcos G. Quiles[1], Elbert E. N. Macau[2] & Nicolás Rubido[3,4]

A prominent feature of complex networks is the appearance of communities, also known as modular structures. Specifically, communities are groups of nodes that are densely connected among each other but connect sparsely with others. However, detecting communities in networks is so far a major challenge, in particular, when networks evolve in time. Here, we propose a change in the community detection approach. It underlies in defining an intrinsic dynamic for the nodes of the network as interacting particles (based on diffusive equations of motion and on the topological properties of the network) that results in a fast convergence of the particle system into clustered patterns. The resulting patterns correspond to the communities of the network. Since our detection of communities is constructed from a dynamical process, it is able to analyse time-varying networks straightforwardly. Moreover, for static networks, our numerical experiments show that our approach achieves similar results as the methodologies currently recognized as the most efficient ones. Also, since our approach defines an N-body problem, it allows for efficient numerical implementations using parallel computations that increase its speed performance.

A remarkable feature observed in several complex networks is the presence of *communities*, namely, modular structures[1–4], as it is observed, for example, on the Internet[5], metabolic networks[6,7], financial time-series[8], or even in networks representing quantum systems[9]. Communities are groups of densely connected nodes within a network, while connections between nodes belonging to different communities are proportionally sparser. They characterize highly interactive local areas in a network, hence, their identification is important to understand the formation, growth mechanisms, and key structures of a network[10,11]. Moreover, the structure of communities shows similarities in regards to the characteristics of the nodes that compose them[4,12]. Thus, through the identification of communities we obtain fundamental information about the network characteristics.

Recently, various mechanism have been proposed for the emergence of communities[13–17], which also derive the heavy-tail degree-distribution and high clustering commonly observed in real-world networks. Nevertheless, detecting communities in any observed network is still an extensive task. Let us take the simplest case of community detection: dividing a network into two parts of equal size such that the number of links connecting these two parts is minimal. This is already a complex task since the computational time to resolve it is non-polynomial, i.e., it is a *NP*-Complete problem[2]. In general, real networks may consist of an arbitrary number of communities, with several sizes, and hierarchical structures within themselves (namely, a community composed by other sub-communities)[4], or even having soft[17] or fuzzy[18] communities (namely, nodes belonging partially to various communities), hence, the problem is even harder. Consequently, and given the importance and complexity of the community detection problem, several models have been proposed[18–40]. However, to the best of our knowledge, a method that efficiently detects these broad community scenarios and gives a physical interpretation for its process, is still missing.

Furthermore, real-world networks are usually time-varying, with sizes and structures that evolve continuously, complicating the community detection even further. For example, if we take a social network such as *Facebook*, new users (nodes) are added or removed daily and new friendships (links) are formed or eliminated. Similarly, an ecological network can change its trophic or symbiotic interactions, namely, the relationship between predators and preys or the intra-species interactions due to predations or competitions. Although several models for community detection in time-varying networks have been proposed, most are based on a static view-point of the

[1]Universidade Federal de São Paulo (Unifesp), Department of Science and Technology (DCT), 12247-014, São José dos Campos, SP, Brazil. [2]Laboratório Associado de Computação e Matemática Aplicada, Instituto Nacional de Pesquisas Espaciais, 12227-010, São José dos Campos, SP, Brazil. [3]Universidad de la República, Instituto de Física Facultad de Ciencias, Iguá 4225, 11400 Montevideo, Uruguay. [4]University of Aberdeen, King's College, Institute for Complex Systems and Mathematical Biology, SUPA, AB24 3UE Aberdeen, United Kingdom. Correspondence and requests for materials should be addressed to N.R. (email: nrubido@fisica.edu.uy)
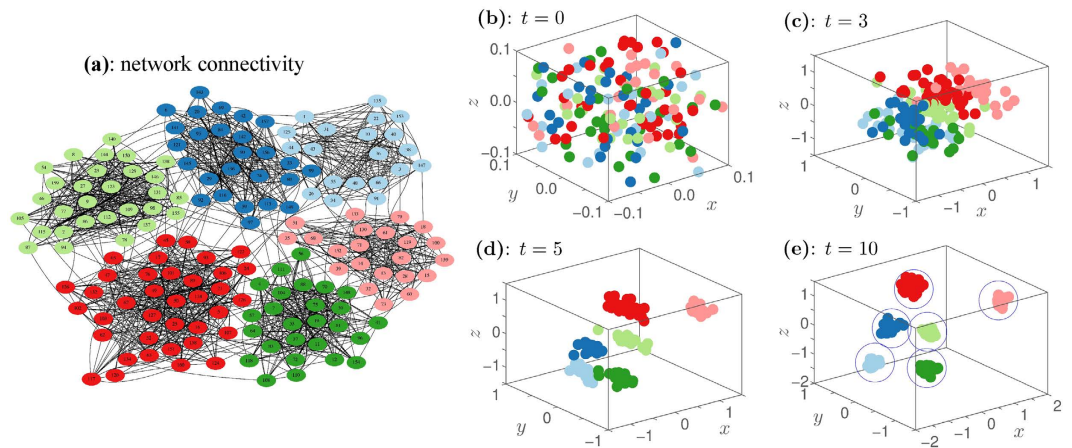
Figure 1. Evolution of our system of interacting particles for the detection of communities in a complex network. Panel (**a**) shows a network with 6 communities analyzed by our particle approach. We consider each node of the network as a particle. The particles interact with each other according to attractive (nodes that are connected in the network by a link) or repulsive (nodes in the network that are disconnected since a link is missing) forces. An arbitrary initial distribution ($t = 0$) of particles is shown in panel (**b**) that corresponds to the nodes in panel (**a**). The snapshots of the particles' evolution at $t = 3$ [panel (**c**)], $t = 5$ [panel (**d**)], and $t = 10$ [panel (**e**)] show the fast convergence of the system to an equilibrium state and the resultant community detection from the particle clusters [encircled on panel (**e**)].

network[4], neglecting its intrinsic evolution. Specifically, these models work as follows. A static snapshot of the network at time t is obtained and the communities of the snapshot are detected by some algorithm. After the network changes, another snapshot is taken at time $t + \delta$ and the algorithm is reapplied. Thus, the network structures previously glimpsed are disregarded, as well as the community evolution from time t to $t + \delta$.

Here, we propose a change in the community detection approach. We consider the nodes of the network as particles obeying a particular dynamics that promptly converges to clustered patterns, namely, the network communities. As a result, our approach makes a fast and optimal community detection, in particular, for time-varying networks. Moreover, is numerically efficient, since N-body problems allow for parallel computations, and is adjustable, since the choice of dynamics for the particles is flexible. This allows to conceive different algorithms which can be tailored to suit different data-sets, increase computational speed (i.e., convergence to the clustered patterns) or improve cluster separation (i.e., communities distinguishability).

Specifically, our approach associates the nodes of a network, e.g., Fig. 1(a), to a spatially distributed system of interacting particles, e.g., Fig. 1(b), hence, it introduces a physical interpretation to the detection of communities in networks. We choose the interaction between the particles to be either attractive (for nodes in the network that are adjacent, i.e., a link exists that connects them) or repulsive (for nodes in the network that are non-adjacent). The functional form for the interactions is chosen such that the system quickly achieves a clustered state, namely, the equilibrium one [Fig. 1(e)], where different particle clusters correspond to different network communities. This functional form is set following the general idea behind diffusive dynamical systems, where a potential function defines the particle dynamics so that the system evolves towards an asymptotically stable equilibrium. Hence, our approach is mathematically tractable within the dynamical system's framework and solves elegantly the topological problems of community detection for any network, either static or time-varying. Also, by defining an N-body problem, it allows for efficient numerical implementations with parallel computations that increase its speed performance. In particular, we find that without parallel computations, our implementation has an $\mathcal{O}(T \times N^2)$ performance, N being the size of the network and T the number of iterations (see Supplementary Material for performance details).

Results

Model: complex networks as interacting particles. Let us consider a complex network $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ [$\mathcal{E}$] is the set of nodes [edges], for which we assign a set of particles in a D-dimensional space. We set $D = 3$ and start by placing the particles randomly, although neither the dimensionality of the space nor the initial distribution of particles seems critical. Our empirical findings show that results are nearly invariant if $D \geq 3$, hence, $D = 3$ is the numerically most efficient and graphically straightforward situation we can choose, and the community detection is based on the asymptotic state of the particle system, hence, close but randomly placed particles suffice. The i-th node in the network ($i = 1, \ldots, N$) is then associated to a particle's position, $\vec{x}_i(t) = (x_i, y_i, z_i)$, that evolves according to

$$\dot{\vec{x}}_i = \alpha \vec{F}_i^{(A)} + \beta \vec{F}_i^{(R)},\tag{1}$$

2

where $\vec{F}_i^{(A)}$ [$\vec{F}_i^{(R)}$] is the attractive [repulsive] interaction force that particle i is subject to due to the other particles (namely, the rest of the adjacent [non-adjacent] nodes) and $\alpha > 0$ [$\beta > 0$] is the relative strength for the attractive [repulsive] force magnitude. These strengths constitute control parameters of our approach.

Let us now set the interaction between particles i and j such that, whenever nodes i and j in the network are connected, namely, the adjacency matrix ij-th entry is $A_{ij} = 1$, the corresponding particles feel a mutual attraction, $\vec{I}_{ij}^{(A)}$. Contrary, if the adjacency matrix $A_{ij} = 0$, the corresponding particles feel a mutual repulsion, $\vec{I}_{ij}^{(R)}$. Consequently, the nodes that are [not] linked together correspond to particles that are [repelled] attracted to each other. These forces are designed so that the cumulative effect of all forces acting upon each particle (namely, $\vec{F}_i^{(A)} = \sum_j \vec{I}_{ij}^{(A)}$ and $\vec{F}_i^{(R)} = \sum_j \vec{I}_{ij}^{(R)}$), for optimally chosen values of α and β, drive the system of interacting particles to an asymptotic stable configuration in which the particles are attracted to different clusters. These clusters of particles are associated to the communities in the network, where particles that end in the same cluster identify a particular community in the network. Conceptually, we assume that if a community exists, the nodes within a community have a larger proportion of their links being shared within the community than the proportion of links connecting those nodes to other nodes outside their community. Hence, the corresponding particles within a community will have interactions that are more attractive than repulsive. Here, we consider the following interaction forces

$$\vec{F}_i^{(A)} = \sum_{j=1}^{N} \vec{I}_{ij}^{(A)} \equiv - \sum_{j=1}^{N} \frac{A_{ij}}{k_i} \frac{(\vec{x}_i - \vec{x}_j)}{\left\| \vec{x}_i - \vec{x}_j \right\|},$$

and

$$\vec{F}_i^{(R)} = \sum_{j=1}^{N} \vec{I}_{ij}^{(R)} \equiv \sum_{j=1}^{N} \frac{R_{ij}}{k_i} \frac{(\vec{x}_i - \vec{x}_j)}{\left\| \vec{x}_i - \vec{x}_j \right\|} e^{-\gamma \|\vec{x}_i - \vec{x}_j\|}, \tag{2}$$

where $\gamma > 0$ is the characteristic decay rate for the repulsive interaction as a function of the distance between particles, namely, $\left\| \vec{x}_i - \vec{x}_j \right\|$, $A_{ij} \geq 0$ is the adjacency matrix of the network, $k_i \equiv \sum_{j=1}^{N} A_{ij}$ is the node's degree, and $R_{ij} \equiv 1 - \delta_{ij} - A_{ij} \geq 0$ is the matrix of the absent links. We stress that other choices for the attractive and repulsive interactions are possible[31], leading to a faster convergence or other clustered patterns (Supplementary Material), although the choice of placing an exponential term in the repulsion is done to guarantee the particles' confinement. Without loss of generality, $\gamma = 1$ throughout our work.

The particular choice of interaction forces [Eq. (2)] we use makes our dynamical approach [Eq. (1)] a gradient system (Supplementary Material). Hence, it is a system that holds an attracting region, such that for any initial spatial distribution of particles close to the origin, the system converges to an equilibrium state. This final equilibrium-state corresponds to the network communities. In particular, to split these communities automatically, we use a clustering algorithm based on a centroids-seed approach, as explained in sec:methodsMethods. The clustering algorithm is similar to the K-means clustering algorithm, but with K varying dynamically. We note that force-directed algorithms[31,40] share similarities with our gradient system, where an energy model is defined and its global minimum is sought. These algorithms also use attractive (repulsive) force between adjacent (non-adjacent) nodes, which cluster the nodes achieving a graphical layout where communities are observed[31]. However, our particle approach also includes the weighing factors A_{ij}/k_i and R_{ij}/k_i that correspond to the unbiased random-walk probabilities of a diffusive processes on the network[29,30,33,38,39], namely, the transition probabilities for a random walker to diffuse from node i to j in a stochastic models. Since our numerical findings show that we overcome the problem of finding a local energy minima, we conjecture that the reason is due to the inclusion of these weighing factors into the particle dynamics.

Numerical experiment: static networks.

We use an explicit Euler scheme for the time discretization of the equations of motion [Eq. (1)] to have the fastest numerical evaluation, i.e., $d\vec{x}_i/dt \simeq [\vec{x}_i(t + \Delta t) - \vec{x}_i(t)]/\Delta t$ with $\Delta t = 1$, and we use the *SNAP* package[41] to implement our networks. The Euler scheme is always viable when the dynamics is a gradient system as convergence is then guaranteed. On the other hand, α and β in Eq. (1) are chosen from experiments with several networks. Our findings show that there is always a combination of values for these parameters that result in particle clusters. Namely, for each value of α one can find a value of β where communities are detected with a minimal error, as seen in Fig. 2(a), where the color code indicates the success rate that our model has (i.e., 0[1] corresponds to an unsuccessful[successful] detection) for a Girvan-Newman (GN) network[1] of $N = 128$ nodes. The relationship between α and β is formally deduced in the Supplementary Material.

We find that a successful community detection is possible for all static networks analysed when $\alpha = 1.0$ and $\beta \in (0.1, 0.4)$, as for example, is seen from Fig. 2(a). In general, if $\beta \gtrsim 0.4$, the repulsion is increased to a point where groups of particles are barely observed due to the influence of a strong repulsion. On the other hand, if $\beta \lesssim 0.1$, the attraction between particles overcomes repulsion causing all the particles or clusters to merge. These are the reasons why, as we vary β while holding α fixed, we detect a hierarchical structure of communities in the network from the resulting particle clusters. Figure 3 shows the particle's asymptotic states for different values of β and fixed $\alpha = 1.0$ on a network with 9 communities [Fig. 3(a)]. For small values of β ($\simeq 0.01$), the 9 communities are merged into a single indistinguishable cluster of particles [Fig. 3(b)], however, as β is increased the particles start to cluster differently and communities are gradually detected, first 3 [Fig. 3(c)] and later 9 [Fig. 3(d)]. This parameter tuning provides a useful hierarchical detection of communities, showing the versatility
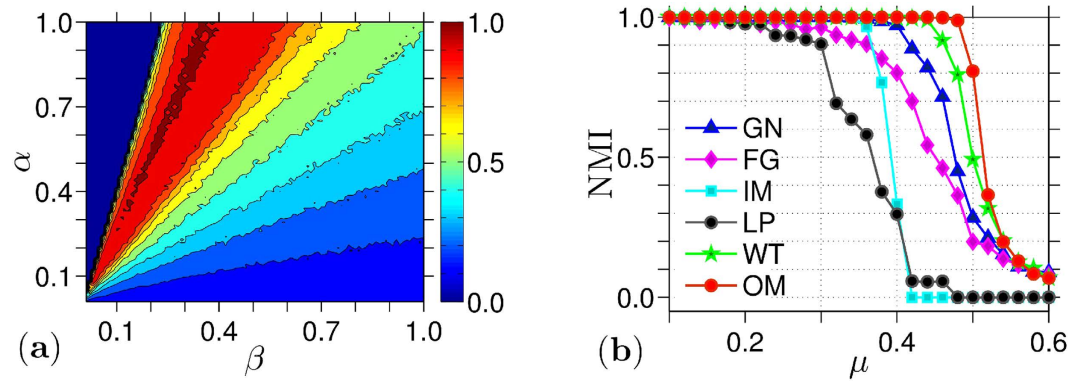
Figure 2. Community detection rates as a function of our parameters (left) and community distinguishability (right) in GN networks. The detection rates are measured by the normalized mutual information[2] (NMI), which gives 1 [0] for a correct [an incorrect] detection (see Methods for details). Community distinguishability in Girvan-Newman (GN) networks[1] is controlled by μ, a mixing parameter which blurs the community distinction as it is increased. Panel **(a)** shows the NMI (color code) that our algorithm gives as a function of the attractive and repulsive parameters, α and β, respectively, for a GN network with $N = 128$ nodes and $\mu = 0.45$. Panel **(b)** shows the NMI for different state-of-the-art methods, namely, GN[1], Fast Greedy[23] (FG), Info MAP[30] (IM), Label Propagation[25] (LP), Walk Trap[29] (WT), and our method (OM) [Eq. (1)] using $\alpha = 1$ and $\beta = 0.36$, which achieves the best detection rate.

of the particle approach, although, maintaining parameter robustness, namely, fine-tuning is generally unneeded (See the Supplementary Material for details on how to estimate β). In other words, we note that having a flexible choice for β allows us to have an algorithm which can detect soft[17] or fuzzy[18] communities as the parameter is tuned.

In order to evaluate the performance of our approach for community detection on general settings, we use the methodology considered in ref. 32. Particularly, we perform a set of experiments using the Girvan-Newman (GN)[1] and Lancichinetti-Fortunato-Radicchi (LFR)[42] benchmarks with the same parameters as in these references. On the other hand, in order to compare systematically our results on these benchmarks with other community detection methods [namely, Girvan-Newman[1] (GN), Fast-Greedy[23] (CNM), page-ranking[30] (InfoMAP), label-propagation[25] (PL), and Walk-trap[29] (RAK) methods] we use the normalized mutual information (NMI)[2], which measures the effectiveness that a community-detection method has to distinguish communities in any given network (see Methods for details).

The community structure of the networks in the GN and LFR benchmarks is controlled by a parameter known as the *mixing parameter*, μ. μ defines the proportion of links that a node in a community has connecting it to nodes from other communities. Meaning that, if $\mu = 0.0$, the communities are completely isolated, namely, inter-community links are absent. If $\mu = 0.5$, half of the node's links are connections with other nodes in its own community (i.e., intra-community links), and the other half of the links are inter-communities links. Hence, as μ increases the distinction between communities is gradually lost, which constitutes a test for the robustness and reliability of the community-detection method.

For any method, as μ is increased from 0 to 1 and communities are gradually merged, the value of the NMI changes from 1 (i.e., all communities are properly detected) to 0 (i.e., no communities are detected). Ideally, the transition of the NMI values from 1 to 0 happens smoothly when $\mu \gtrsim 0.5$, which corresponds to the situation where communities start being indistinguishable. In this sense, we see from Figs 2(b) and 4 that our approach detects communities effectively for both benchmarks, GN and LFR respectively, and even outperforms (on average) the other state-of-the-art community detection methods[1,23,25,26,29,30] when μ is large. For example, we observe that our model can detect communities on the LFR benchmark to values up to $\mu \sim 0.8$ [Fig. 4(a)], which is a scenario where the community distinction is extremely subtle.

Numerical experiment: time-varying network. As an illustration of the efficiency of our approach in detecting the communities of time-varying networks, we show the results obtained for a particular scenario in Fig. 5. Using the methodology proposed in[43], we start with a network of $N = 128$ nodes with 4 communities containing an even number of nodes (namely, 32), as shown in Fig. 5(a) (where inter-community links have been discarded on the graphical representation for the sake of clarity). Then, the communities evolve dynamically by growing/shrinking and merging/splitting. On a particular instant, after this modification on the network structure, two communities are effectively merged into a single community of 64 nodes, which leaves the network with a total of three communities [Fig. 5(d)]. As Fig. 5(f–j) show for each instant snapshot on Fig. 5(a–e) [$t = 101$, $t = 103$, $t = 107$, $t = 110$, and $t = 120$, respectively], the spatial configuration of particles after a few iterations rapidly converges into a new steady-state, that again, corresponds to the correct detection of the communities present in the modified network. Since the new community structure detected by our approach is obtained by running the algorithm from the former clustered state, the convergence speed is increased in comparison to that of a random initial condition (see Methods for details and implementation details).
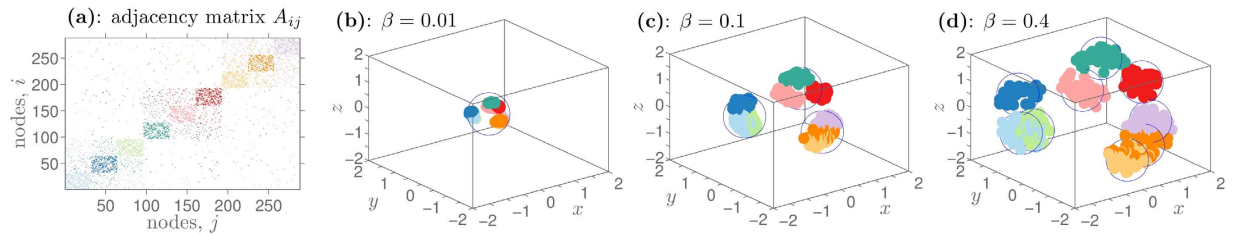
Figure 3. **Hierarchical community detection of a network using the interacting particles' approach.** Our approach [Eqs. (1)–(2)] is applied to the network in panel (**a**), where the dots in the matrix represent a link in the network connecting node i (row) to j (column) and the color code is introduced to highlight the distinction between the 9 communities in the network. The network has $N = 288$ nodes, $\langle k \rangle = k_{max} = 16$, $\mu_1 = 0.25$ (for the micro-communities, which have sizes of 32 nodes), and $\mu_2 = 0.08$ (for the macro-communities, which have sizes of 96 nodes). Panels (**b**–**d**) show the final state of the particle system for different repulsion strengths (β) and fixed attraction strength ($\alpha = 1.0$), starting from an arbitrary initial distribution of particles, as in Fig. 1(b). The community distinction emerges gradually and hierarchically as β is increased, which allows for soft community detection[17].
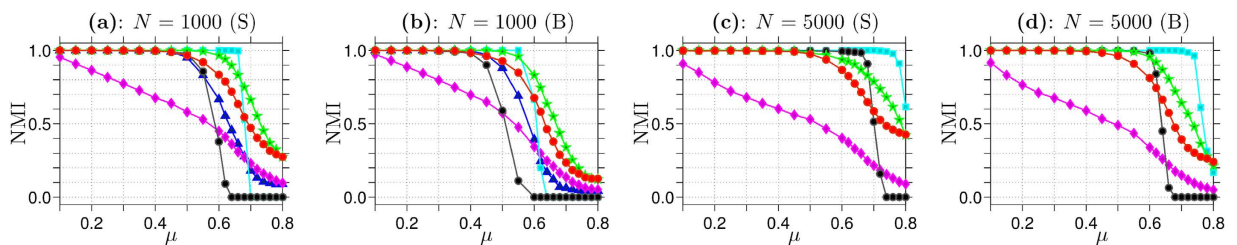

Figure 4. **Community detection rates as a function of community distinguishability in LFR networks.** The panels show the normalized mutual information[2] (NMI) values as a function of the mixing parameter, μ (namely, the degree of community distinguishability), for the algorithms considered in Fig. 3(b) on Lancichinetti-Fortunato-Radicchi networks[32]. Four network scenarios are shown: $N = 1000$ and $N = 5000$ nodes with "small" (S) [panels (**a**,**c**)] and "big" (B) [panels (**b**,**d**)] communities (see Methods for details on the community sizes). Each point on the curves correspond to the average of the NMI value over 200 network realizations, excluding the GN analysis for $N = 5000$ because of its high computational cost. The symbols follow the labels set in Fig. 3(b) for each algorithm and the parameters for our algorithm are: $\alpha = 1$ and $\beta = 0.15$ [panel (**a**)], 0.12 [panels (**b**,**c**)], and 0.09 [panel (**d**)].

Numerical experiment: real-world network. For the sake of completeness and to test the accuracy of our approach, we provide an experiments on a real-world social network. The experiment is conducted using the network of American football games introduced in ref. 22. This network contains 115 nodes, which represents the teams of the Division IA college games in the 2000 season. The links between nodes (teams) are the matches. The teams are split into twelve conferences of 8 to 12 teams. The matches between teams are more frequent between teams belonging to the same conferences, thus, we might expect the formation of communities. In our experiment, we perform a hierarchical detection of communities by varying the parameter β from 0.01 to 0.7. As it is shown in Fig. 6, the formation and the division of communities increases as the parameter β is increased. When $\beta \geq 0.55$, the communities revealed by our method are compatible to those observed in the real division of the conferences[22].

Discussion

Our findings show that, treating a network as a set of interacting particles, where the force between particles is attractive [repulsive] when nodes are adjacent [non-adjacent] and is weighed by the random walk probability of transitioning between the nodes, allows to detect communities with high accuracy and low parameter sensitivity, outperforming several state-of-the-art community-detection algorithms. In summary, the main contributions from our approach are various. First, its dynamical nature. This means that, if a change in the network topology occurs, such as the inclusion or removal of a node or link, it is naturally interpreted as a perturbation in the particle system, thus reaching a new equilibrium state after a short transient. In this way, we avoid reapplying our approach when structural changes happen. On the contrary, most of the community detection methods are unable to deal with time-varying networks straightforwardly, since the algorithms must be reapplied every time a structural change is observed. Second, the adjustment of the interaction parameters allows us to detect communities hierarchically, which could also allow to identify networks with soft communities[17]. Third, the flexibility of our particle approach allows the choice of other functional forms for the interactions between particles, hence, designing different community-detection algorithms.
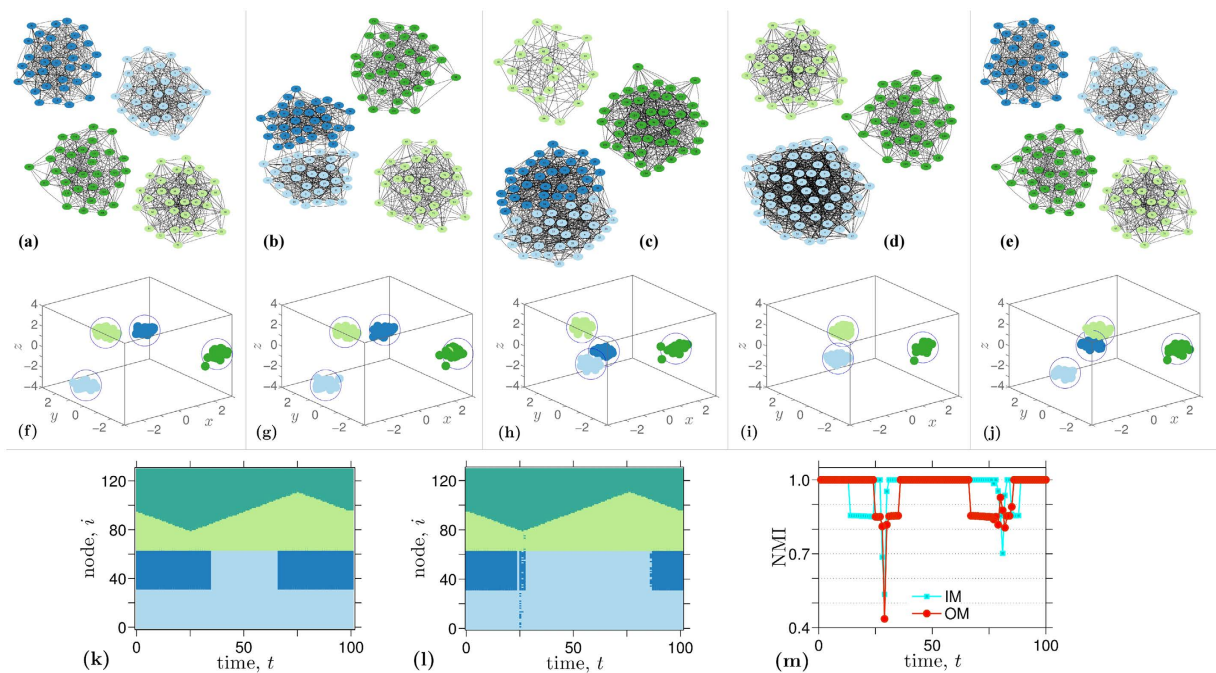
Figure 5. Evolution of our dynamical model of interacting particles for community detection on a time-varying complex network. Snapshots of the network evolution are shown in panels (**a**–**e**). The evolution starts with a network composed of 4 communities with 32 nodes each. From (**a**–**e**), using the methodology proposed in Granell, *et al.*[43], the communities evolve dynamically by growing/shrinking and merging/splitting. Panels (**f**–**j**) depict the clustered particles' steady-state of the networks in panels (**a**–**e**). Panel (**k**) illustrates in color code the evolution of the communities evolving in time and panel (**l**) illustrates the outcome of our approach, namely, the identification of the nodes belonging to one of the three or four communities present in panels (**a**–**e**). Panel (**m**) shows the NMI achieved by our model and by the Infomap[30].
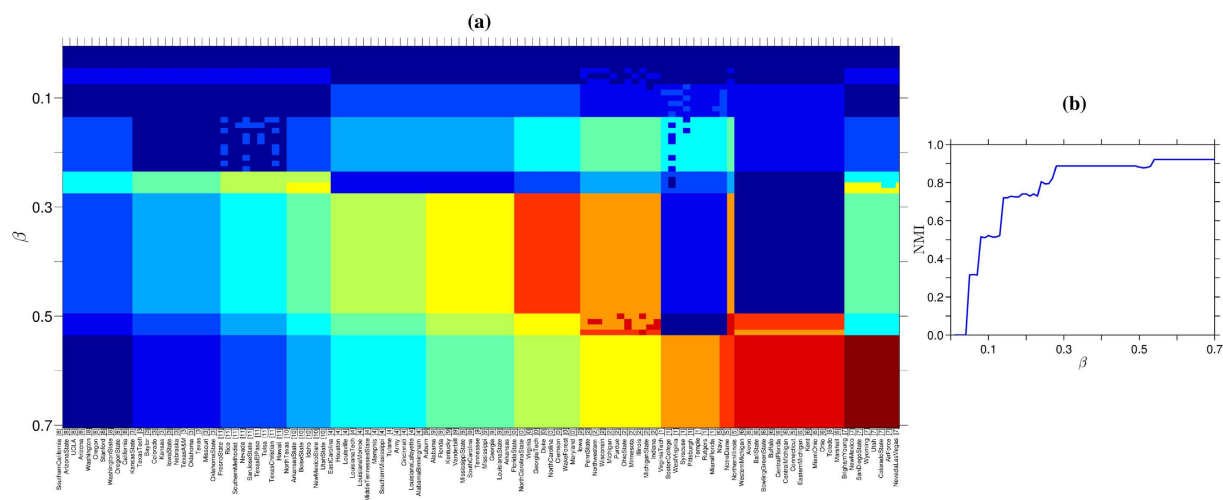
Figure 6. Hierarchical detection of communities on the American Football network. The color code in panel (**a**) corresponds to our hierarchical detection of the different communities that are observed in this network of $N = 115$ teams (nodes) where the football matches are the links. Each football team in this panel (horizontal axes) belongs to a community[22], which is signalled by a number between 0 to 11. From top to bottom, as β is increased (i.e., the repulsive force strength), the communities that our approach detects start to split hierarchically, hence, more colors emerge. In particular, the NMI that we achieve on this network for different β values is shown in panel (**b**).

We note that, although the use of particle systems to solve a wide range of problems is well-known[19] dating back to the use of molecular dynamics simulation techniques for hydrodynamic problems and celestial

mechanics N-body problems, this particle approach is novel when it comes to community detection in networks. Furthermore, since N-body problems allow for parallel computations and we choose a particle dynamics that derives from a potential function, our approach allows for the design of numerically efficient and stable algorithms, namely, algorithms that require minimum floating point operations per iteration and allow for larger iteration time-steps.

Methods

Network benchmarks and the Normalized Mutual Information.

We evaluate our methodology systematically following ref. 32 and using the normalized mutual information[2] (NMI). Specifically, we perform a set of experiments taking networks that are considered benchmarks for testing community detection algorithms and evaluate the efficiency of our approach to detect communities on these networks by means of the resultant NMI value.

The *benchmarks* we choose are the Girvan-Newman (GN)[1] networks and the Lancichinetti-Fortunato-Radicchi (LFR)[42] networks, which are implemented using the same parameters as in these references. In particular, for the LFR networks the average degree was set to 20, the maximum degree to 50, the exponent of the degree distribution to -2.0, and the exponent of the community size distribution to -1.0. With these parameters, the following scenarios were considered: networks with 1000 nodes and community sizes varying from 10 to 50 nodes, which we name as small (S); networks with equal size but with communities varying from 20 to 100 nodes, which we name as big (B); and two scenarios more that follow the same range for the community size as the former two cases but with networks with 5000 nodes.

The effectiveness of the algorithms in detecting communities is quantified by the *normalized mutual information* (NMI) measure[2]. This measure is calculated from a confusion matrix $\mathbf{N}$, where rows correspond to the expected community structure and the columns correspond to the obtained community structure. The NMI is then defined by

$$I = \frac{-2\sum_{i=1}^{M_R}\sum_{j=1}^{M_F} N_{ij} \, \log(N_{ij}N/N_i N_j)}{\sum_{i=1}^{M_R} N_i \log(N_i/N) + \sum_{j=1}^{M_F} N_j \log(N_j/N)}, \tag{3}$$

where M_R [M_F] corresponds to the number of expected [found] communities, N_{ij} represents the number of nodes belonging to the real community i but clustered within community j according to the algorithm's outcome, N_i [N_j] defines the row [column] sum over i [j] of matrix $\mathbf{N}$, and N represents the total number of nodes in the network.

Community Detection Algorithm: centroids-seed approach.

The particles in our approach [Eqs. (1)–(2)] self-organize into clusters after a short transient. This transient period is evaluated by analysing the instantaneous variations in the average of the repulsive interactions between particles, $\Delta R(t)$, where

$$\Delta R(t) = \frac{1}{N} \left| \sum_i \left\| \frac{\vec{F}_i^{(R)}(t)}{k_i} \right\| - \sum_i \left\| \frac{\vec{F}_i^{(R)}(t-1)}{k_i} \right\| \right|. \tag{4}$$

If $\Delta R(t)$ at time t is below a certain threshold θ_r, an equilibrium state has been reached and the algorithm iterations can be stopped. This equilibrium state provides the community structure of the network for a given set of parameters (α and β).

In order to differentiate the particles belonging to different clusters automatically, after the transient, a centroids-seed approach is taken[34]. Namely, seeds are added randomly into the particle's space with spatial position given by $\mathbf{s}_k(t) = (rand(), rand(), rand())$, with $k = 1, \ldots, S$, where S is the total number of seeds. These seeds $\vec{s}_k$ are used to identify the communities according to their membership. In order to identify the community that each node belongs to, a variable y_i is defined as the community label, i.e. if $y_i(t) = 1$, it means that at time t node i, associated with particle $\vec{x}_i$, belongs to the community number 1, or in other words, is associated with the seed $\vec{s}_1$.

Hence, the community assignment of each particle is done by evaluating the distance from the particle, e.g., $\vec{x}_i$, to all existing seeds, $\vec{s}_k$, at time t by

$$y_i(t) = \arg\{\min_k \|\vec{x}_i(t) - \vec{s}_k(t)\|\}, \tag{5}$$

which means particle $\vec{x}_i$ is always linked with its closest seed. We also calculate the quadratic error of each seed from

$$E_{s_k}(n) = \sum_{\forall i \in \Delta_k} (\|\vec{x}_i(t) - \vec{s}_k(t)\|)^2 \tag{6}$$

where Δ_k represents the set of particles associated with the seed s_k and $|\Delta_k|$ is the number of particles in the set Δ_k. Consequently, the error $E_{s_k}(n)$ is somewhat the average quadratic distance of all particles within a cluster at iteration n. Then,

1. if any seed has error zero, $E_{s_k}(n) = 0$, it means that this seed is isolated in the particle space, and it is associated with none or only one particle; consequently, the seed is removed;
2. if any seed has error greater than a threshold θ_s, $E_{s_k}(n) \geq \theta_s$, it means that this seed is associated with a

Data: Network - adjacency-list

1 Load network;
2 Start particles;
3 Set parameters;
4 **while** $\Delta R(t) > \theta_r$ **do** % method's core
5   Run Model - Eqs (1)-(2)
6 **end**
7 **while** $\Delta E(n) > \theta_c$ **do** % clustering algorithm
8   Merge close seeds;
9   Assign each particle to its closest seed;
10   Check removal condition (1);
11   Check insertion condition (2);
12   Recompute position of the seeds - Eq. (8);
13 **end**

Figure 7. Community detection algorithm with the centroids-seed pseudo-code.

highly heterogeneous cluster of particles, which indicates that a new cluster must be created. Thus, a new seed is inserted.

These conditions need the definition of a threshold θ_s, which sets the maximum heterogeneity level allowed in each particle cluster. In particular, if any seed is removed or added, the particles are reassigned to the seeds [Eq. (5)]. Finally, the position of the seeds themselves are reassigned by

$$\vec{s}_j(t) = \frac{1}{|\Delta_j|} \sum_{\forall i \in \Delta_j} \vec{x}_i(t).$$

(7)

This process is repeated until the convergence of the seeds, which is observed when the variation of the errors of the seeds, $E_{s_k}(n)$, are stabilized. Namely, when $\Delta E(n) \leq \theta_c$, where

$$\Delta E(n) = \sum E_{s_k}(n) - \sum E_{s_k}(n-1)$$

(8)

and θ_c we set constant at 10^{-2}. From our numerical experiments, we observe that the number of steps (n) required to reach the stopping condition has a linear relationship to the number of detected communities, which we note is identical to the linear relationship found for the run-time reported for the Potts model[18].

The overall algorithm is summarized in Fig. 7 and can be found in[44]. Throughout this work, we set the threshold parameters, namely, θ_r and θ_s, to 10^{-2} and 0.5, respectively.

Time-varying networks: algorithm implementation. In general, a time-varying network starts from an initial topology, namely, an initial network connectivity, and then it evolves its links according to some known or unknown function. Our method takes into account this initial network connectivity to calculate the equilibrium state of the associated particle system (where particles start from a random initial placement but close to the origin). Then, it evolves the particle systems from this initial equilibrium state at the same time as the links in the network are modified due to the network's evolution. Also, if the network grows or shrinks as time evolves (i.e., N increases or decreases), particles are added close to the origin or removed. However, the network's topology evolution is carried at a slower time-scale than the clustering dynamics between particles. Hence, we can think that at any time, we are pausing the network's evolution and computing its communities, thus, retrieving a snapshot-like analysis. Nevertheless, we highlight the fact that our approach is applied continuously, contrary to the static-network snapshot method, hence, producing a real-time community-detection method that is unbiased.

The modified equations of motion for the particles are

$$\vec{F}_i^{(A)}(t) \equiv - \sum_{j=1}^{N(\tau)} \frac{A_{ij}(\tau)}{k_i(\tau)} \frac{(\vec{x}_i(t) - \vec{x}_j(t))}{\left\| \vec{x}_i(t) - \vec{x}_j(t) \right\|},$$

$$\vec{F}_i^{(R)}(t) \equiv \sum_{j=1}^{N(\tau)} \frac{R_{ij}(\tau)}{k_i(\tau)} \frac{\left(\vec{x}_i(t) - \vec{x}_j(t)\right)}{\left\| \vec{x}_i(t) - \vec{x}_j(t) \right\|} e^{-\gamma \| \vec{x}_i(t) - \vec{x}_j(t) \|},$$

(9)

where $\vec{x}_i(t)$ is the i-th particle position at time t and $A_{ij}(\tau)$ [$R_{ij}(\tau)$] is the adjacency [complementary adjacency] matrix, that, at time t, has evolved τ. Similarly, the number of nodes in the network, $N(\tau)$, and their degree, $k_i(\tau) = \sum_j A_{ij}(\tau)$, which at time t has a value according to the evolution of the connectivity matrix $A_{ij}(\tau)$. The

```
   Data: Network - adjacency-list
 1 Load the initial network;
 2 Start particles;
 3 Set parameters;
 4 repeat
 5     while ΔR(t) > θ_r do % method's core
 6       │ Run Model - Eqs (1)-(2)
 7     end
 8     while ΔE(t) > θ_c do % clustering algorithm
 9       │ Merge close seeds;
10       │ Assign each particle to its closest seed;
11       │ Check removal condition (1);
12       │ Check insertion condition (2);
13       │ Recompute position of the seeds - Eq. (8);
14     end
15     if network changes then
16       │ load changes;
17     else
18       │ sleep until a change has been detected;
19     end
20 until no more network changes are expected;
```

Figure 8. Community detection algorithm for time-varying networks.

time τ is the instantaneous time for the evolving topology, which in terms of t, is much slower. Namely, $\dot{t} = 1$ while $\dot{\tau} = \varepsilon \ll 1$, which is treating the evolution of the particle forces and the evolution of the connectivity as the decoupling of a dynamical system between its fast and slow dynamics.

The algorithm in Fig. 8 summarizes how the method is applied to time-varying networks. It is worth noting that, in contrast to Fig. 7, both the model's core and the clustering routine do not start from a random initial condition, but from the condition built on the previous iteration. Thus, the number of steps necessary to reach a new equilibrium (position of the seeds) is lowered.

References

1. Girvan, M. & Newman, M. E. J. Community structure in social and biological networks. *Proc. Natl. Acad. Sci.* **99,** 7821–7826 (2002).
2. Danon, L., Díaz-Guilera, A., Duch, J. & Arenas, A. Comparing community structure identification. *J. Stat. Mech* **P09008,** 1–10 (2005).
3. Danon, L., Duch, J., Arenas, A. & Díaz-Guilera, A. *Large Scale Structure and Dynamics of Complex Networks: From Information Technology to Finance and Natural Science.* Ch. Community structure identification, 93–113 (World Scientific, 2007).
4. Fortunato, S. Community detection in graphs. *Phys. Rep.* **486,** 75–174 (2010).
5. Flake, G. W., Lawrence, S., Giles, C. L. & Coetzee, F. M. Self-organization and identification of web communities. *Computer* **35,** 66–70 (2002).
6. Jeong, H., Tombor, B., Albert, R., Oltvai, Z. N. & Barabási, A.-L. The large scale organization of metabolic networks. *Nature* **407,** 651–654 (2000).
7. Guimera, R. & Nunes Amaral, L. A. Functional cartography of complex metabolic networks. *Nature* **433,** 895–900 (2005).
8. MacMahon, M. & Garlaschelli, D. Community detection for Correlation Matrices. *Phys. Rev. X* **5,** 021006 (2015).
9. Faccin, M., Migdal, P., Johnson, T. H., Bergholm, V. & Biamonte, J. D. Community detection in Quantum Complex Networks. *Phys. Rev. X* **4,** 041012 (2014).
10. Barabási, A.-L. *Linked: How Everything Is Connected to Everything Else and What It Means* (Plume Editors, 2002).
11. Newman, M. E. J. *Networks: An Introduction* (Oxford University Press, 2010).
12. Lancichinetti, A., Radicchi, F., Ramasco, J. J. & Fortunato, S. Finding Statistically Significant Communities in Networks. *PLOS One* **6,** 1–18 (2011).
13. Boguñá, M., Pastor-Satorras, R., Díaz-Guilera, A. & Arenas, A. Models of social networks based on social distance attachment. *Phys. Rev. E* **70,** 056122 (2004).
14. Papadopoulos, F., Kitsak, M., Serrano, M. A., Boguñá, M. & Krioukov, D. Popularity versus similarity in growing networks. *Nature* **489,** 537–540 (2012).
15. Bianconi, G., Darst, R. K., Iacovacci, J. & Fortunato, S. Triadic closure as a basic generating mechanism of communities in complex networks. *Phys. Rev. E* **90,** 042806 (2014).
16. Wu, Z., Menichetti, G., Rahmede, C. & Bianconi, G. Emergent Complex Network Geometry. *Sci. Rep.* **5,** 10073 (2015).
17. Zuev, K., Boguñá, M., Bianconi, G. & Krioukov, D. Emergence of Soft Communities from Geometric Preferential Attachment. *Sci. Rep.* **5,** 9421 (2015).
18. Reichardt, J. & Bornholdt, S. Detecting fuzzy community structures in complex networks with a Potts model. *Phys. Rev. Lett.* **93,** 218701 (2004).
19. Li, S. & Liu, W. K. Meshfree and particle methods and their applications. *Appl. Mech. Rev.* **55,** 1–34 (2002).
20. Zhou, H. Distance, dissimilarity index, and network community structure. *Phys. Rev. E* **67,** 061901 (2003).
21. Newman, M. E. J. & Girvan, M. Finding and evaluating community structure in networks. *Phys. Rev. E* **69,** 026113 (2004).
22. Newman, M. E. J. Fast algorithm for detecting community structure in networks. *Phys. Rev. E* **69,** 066133 (2004).

23. Clauset, A., Newman, M. E. J. & Moore, C. Finding community structure in very large networks. *Phys. Rev. E* **70,** 066111 (2004).
24. Clauset, A. Finding local community structure in networks. *Phys. Rev. E* **72,** 026132 (2005).
25. Pons, P. & Latapy, M. Computing communities in large networks using random walks. *Computer and Information Sciences-ISCIS 2005* Springer Berlin Heidelberg, 284–293 (2005).
26. Newman, M. E. J. Finding community structure using the eigenvectors of matrices. *Phys. Rev. E* **74,** 036104 (2006).
27. Boccaletti, S., Ivanchenko, M., Latora, V., Pluchino, A. & Rapisarda, A. Detecting complex network modularity by dynamical clustering. *Phys. Rev. E* **75,** 045102 (2007).
28. Costa, L. F., Rodrigues, F. A., Travieso, G. & Boas, P. R. V. Characterization of complex networks: a survey of measurements. *Adv. Phys.* **56,** 167–242 (2007).
29. Raghavan, U. N., Albert, R. & Kumara, S. Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E* **76,** 036106 (2007).
30. Rosvall, M. & Bergstrom, C. T. Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci.* **105,** 1118–1123 (2008).
31. Noack, A. Modularity clustering is force-directed layout. *Phys. Rev. E* **79,** 026102 (2009).
32. Lancichinetti, A. & Fortunato, S. Community detection algorithms: A comparative analysis. *Phys. Rev. E* **80,** 056117 (2009).
33. Morarescu, I. C. & Girard, A. Opinion dynamics with decaying confidence: application to community detection in graphs. *IEEE T. Automat. Contr.* **56(8),** 1862–1873 (2011).
34. Breve, F. A., Zhao, L., Quiles, M. G., Pedrycz, W. & Liu, J. Particle Competition and Cooperation in Networks for Semi-Supervised Learning. *IEEE T. Knowl. Data En.* **24,** 1686–1698 (2012).
35. Zhao, Z. *et al.* Topic oriented community detection through social objects and link analysis in social networks. *Knowl.-Based Syst.* **26,** 164–173 (2012).
36. Chen, J. & Saad, Y. Dense Subgraph Extraction with Application to Community Detection. *IEEE T. Knowl. Data En.* **24,** 1216–1230 (2012).
37. Massaro, E., Bagnoli, F., Guazzini, A. & Lio, P. Information dynamics algorithm for detecting communities in networks. *Commun. Nonlinear Sci. Numer. Simulat.* **17,** 4294–4203 (2012).
38. Delvenne, J.-C., Schaub, M. T., Yaliraki, S. N. & Barahona, M. The stability of a graph partition: A dynamics-based framework for community detection. *Dynamics On and Of Complex Networks: Applications to Time-Varying Dynamical Systems* Springer New York **2,** 221–242 (2013).
39. Lambiotte, R., Delvenne, J.-C. & Barahona, M. Random Walks, Markov Processes and the Multiscale Modular Organization of Complex Networks. *IEEE T. Network Sci. En.* **1(2)** 76–90 (2014).
40. Tamassia, R. *Handbook of graph drawing and visualization* (CRC press, 2013).
41. Leskovec, J. & Sosič, R. SNAP: Stanford Network and Analysis Platform. SNAP, (2014). Date of access: 11/05/2015. URL http://snap.stanford.edu/snap.
42. Lancichinetti, A., Fortunato, S. & Radicchi, F. Benchmark graphs for testing community detection algorithms. *Phys. Rev. E* **78,** 046110 (2008).
43. Granell, C., Darst, R. K., Arenas, A., Fortunato, S. & Gómez, S. Benchmark model to assess community structure in evolving networks. *Phys. Rev. E* **92(1),** 012805 (2015).
44. Quiles, M. G. Particle Community: A dynamical model for detecting communities in complex networks. GitHub, (2016). Date of access 28/01/2016. URL https://github.com/quiles/ParticleCommunity.

Acknowledgements

Author Contributions

M.G.Q. performed the numerical experiments and analysed the results. N.R. performed the mathematical derivations. All authors contributed with the conception of the method and to the writing of the manuscript.

Additional Information

Supplementary information accompanies this paper at http://www.nature.com/srep

Competing financial interests: The authors declare no competing financial interests.

How to cite this article: Quiles, M. G. *et al.* Dynamical detection of network communities. *Sci. Rep.* **6**, 25570; doi: 10.1038/srep25570 (2016).