Random walks and community detection in hypergraphs

Carletti, Timoteo; Fanelli, Duccio; Lambiotte, Renaud

Link to publication

Random walks and community detection in hypergraphs

Random walks and community detection in hypergraphs

Timoteo Carletti[1]

[1]*naXys, Namur Institute for Complex Systems, University of Namur, Belgium*

Duccio Fanelli[2]

[2]*Università degli Studi di Firenze, Dipartimento di Fisica e Astronomia,*
CSDC and INFN, via G. Sansone 1, 50019 Sesto Fiorentino, Italy

Renaud Lambiotte[3]

[3]*Mathematical Institute, University of Oxford, UK*

We propose a one-parameter family of random walk processes on hypergraphs, where a parameter biases the dynamics of the walker towards hyperedges of low or high cardinality. We show that for each value of the parameter, the resulting process defines its own hypergraph projection on a weighted network. We then explore the differences between them by considering the community structure associated to each random walk process. To do so, we adapt the Markov stability framework to hypergraphs and test it on artificial and real-world hypergraphs.

I. INTRODUCTION

In the last two decades, networks have emerged as a powerful framework to model and study complex systems [1, 2], providing a rich set of tools and methods that can be used independently of the nature of their constituting components [3–5]. At the core of network science, there is the interplay between structure and dynamics [6–8], as the topology imposes constraints of how dynamical processes propagate between nodes and, reversely, dynamical processes are at the core of several algorithms to extract information from the underlying structure [9]. An important context where this interplay is essential is the modular, or community, structure of networks. On the one hand, several works have shown that the presence of communities slows down diffusive processes on networks, in particular random walk processes [10]. Here, communities are understood as groups of densely connected nodes, leading to the presence of bottlenecks between communities. Important concepts include the Cheeger inequality, providing relations between the mixing time, e.g. time for a random walk process to relax to stationarity, and the notion of conductance, quantifying the bottlenecks in networks. On the other hand, random walk processes are at the heart of several algorithms to extract communities in networks. Community detection [11] is a central element in the toolkit of network science, allowing to produce coarse-grained representation of large-scale networks [12, 13] and thus to identify groups of nodes whose behaviour is relevant for the process under study. Methods based on random walks include the Map Equation [14] and Markov stability [15, 16], exploiting the fact that good communities tend to capture random walkers for long times before they can escape them.

Networks make assumptions about the structure of interacting systems, often implicitly, and recent research has questioned the relevance of these assumptions in real-world data, leading to the concept of higher-order networks [17, 18]. In networks, for instance, the building blocks for interactions are pairwise interactions, and the whole network is formed by combining such pairwise interactions. In contrast, many interacting systems are made of interactions that involve more than two nodes and can not be decomposed further, i.e. they act as a whole. A canonical example is collaboration networks, where groups of authors interact to produce research papers [19, 20]: the emerging results are a product of the group rather than reflecting pairwise interexchanges. More generally, systems characterised by multibody interactions abound in a variety of scientific domains, from functional brain networks [21, 22] to protein interaction networks [23] and ecology [24].

The inadequacy of standard networks to model multibody interactions motivates the development of appropriate higher-order models, often enriching and generalising the standard network paradigm. The two most popular approaches are simplicial complexes [25–27] and hypergraphs [28–30]. Simplicial complexes are a model from topological data analysis, whose aim is to characterise the shape of data, in terms of the presence of holes between the points, and have been proposed as generalisations of networks with applications in epidemic spreading [31, 32] and synchronisation [33, 34]. The focus of this paper is on hypergraphs, a domain that has a long tradition in graph theory [28] but whose impact on dynamics has only been considered more recently. Recent works include applications to social contagion model [35, 36], the modelling of random walks [20], the study of synchronisation [37–39], diffusion [36], non-linear consensus [40] and the emergence of Turing patterns [39]. Hypergraphs constitute a powerful and flexible paradigm, as they encode interactions by hyperedges, defined as groups of arbitrary size between nodes. In situations when all the hyperedges have size 2, hypergraphs reduce to standard networks. Hypergraphs have several advantages, as they allow to efficiently handle very large hyperedges and, even more importantly, a heterogenous distribution of

hyperedges' sizes. Moreover, the information on the high-order structure of the embedding support are stored in a matrix whose dimension depends only on the number of nodes [20, 36] thus avoiding the use of tensors.

The main purpose of this paper is to explore the interplay between dynamics and structure in hypergraphs. More specifically, we will generalise the notion of Markov stability. As we will discover, different types of random walk processes, and their associated Laplacian, can be defined on hypergraphs, each leading to different quality functions and different partitions into communities. This flexibility originates from the fact that hyperedges are characterised by a feature, their size, and that a choice has to be made to model how this feature affects the diffusion. Classical studies in graph theory often assume that the hypergraphs are uniform, that is the hypergedges all have the same size [41, 42]. The first random walk Laplacian defined for general hypergraphs can probably be traced back to the seminal paper of Zhou and collaborators [43], where each hyperedge is endowed with an arbitrary weight, acting as a bias to the walkers dynamics. While this weight is often considered to be a free parameter that can be chosen a priori, it may naturally emerge in certain models, such as in [20] where the transition rates of the process are linearly biased by the size of the hyperedges, i.e. a walker follows a hyperedge proportionally to its size. In general, different choices for this weight, and the resulting biases on the random walk trajectories, lead to different transition matrices. Our main result is to investigate and to characterise how these differences translate into different communies in hypergraphs.

This paper is organised as follows. In section II, we define random walks on hypergraphs and introduce a one-parameter family of processes, extending previous models. In section III, we generalise the concept of Markov stability to hypergraphs and, in section IV, we explore the impact of the parameter of the random walk process on the uncovered communities, looking at both artificial and real-world networks. Finally, in section V, we conclude and discuss the implications of our work.

II. HYPERGRAPHS AND RANDOM WALKS

Let us consider an hypergraph $\mathcal{H}(V, E)$, where $V = \{v_1, \ldots, v_n\}$ denotes the set of n nodes and $E = \{E_1, \ldots, E_m\}$ the set of m hyperedges, that is for all $\alpha = 1, \ldots, m$: $E_\alpha \subset V$, i.e. an unordered collection of vertices. Note that if $E_\alpha = \{u, v\}$, i.e. $|E_\alpha| = 2$, then the hyperedge is actually a "standard" edge denoting a binary interaction among u and v. If all hyperedges have size 2, the hypergraph is thus a network. The hypergraph can be encoded by its *incidence matrix* $e_{i\alpha}$, where we adopt the convention of using roman indexes for nodes and greek ones for hyperedges

$$e_{i\alpha} = \begin{cases} 1 & v_i \in E_\alpha \\ 0 & \text{otherwise}. \end{cases} \tag{1}$$

A standard procedure to construct the $n \times n$ adjacency matrix of the hypergraph is $\mathbf{A} = \mathbf{e}\mathbf{e}^T$, whose entry A_{ij} represents the number of hyperedges containing both nodes i and j. Note that it is often customary to set to zero the diagonal elements of the adjacency matrix. Let us also define the $m \times m$ hyperedges matrix $\mathbf{B} = \mathbf{e}^T\mathbf{e}$, whose entry $B_{\alpha\beta}$ counts the number of nodes in $E_\alpha \cap E_\beta$. $\mathbf{B}$ can be seen as the (weighted) adjacency matrix of the dual hypergraph, i.e. where hyperedges of the original hypergraph become nodes of the new structure and two nodes are connected by a weighted link counting how many nodes (in the original hypergraph) are shared by the two hyperedges, namely $B_{\alpha\beta}$. Note that a similar construction has been proposed in [44] to extract a n-clique graph from a network, namely a network whose nodes are the n-clique of the original one and whose nodes are connected if the n-clique share at least one node (in the original network). The main difference in the present case is that hyperedges can have an heterogeneous size distribution and thus provide a more flexible framework.

We can define a random walk process on a hypergraph as follows. The agents are located on the nodes and hop between nodes at discrete times. In a general setting, the walkers may give more or less importance to hyperedges depending on their size, a choice that may then bias their moves [20]. This process is captured by the weighted adjacency matrix

$$K_{ij}^{(\sigma)} = \sum_\alpha (B_{\alpha\alpha} - 1)^\sigma e_{i\alpha} e_{j\alpha} \quad \forall i \neq j \quad \text{and} \quad K_{ii}^{(\sigma)} = 0, \tag{2}$$

where σ is a real parameter whose role will be discussed in the following; note that $B_{\alpha\alpha} - 1$ denotes the number of nodes in the hyperedge E_α available to the walker, i.e. discarding the node where she is sitting initially. It is worth emphasising that in principle we could have used a generic monotone function f of the hyperedge size, in defining the above quantities. For illustrative purposes we have here chosen to limit the analysis to the relevant, although particular, setting of a power law bias. The transition probabilities of the examined process are then obtained by

normalising the columns of the weighted adjacency matrix

$$T_{ij}^{(\sigma)} = \frac{K_{ij}^{(\sigma)}}{\sum_{\ell \neq i} K_{i\ell}^{(\sigma)}} \quad \forall i \neq j \quad \text{and } T_{ii}^{(\sigma)} = 0 \,. \tag{3}$$

Note that a "lazy" process could have been defined by assuming that walkers are able to stay put on a node in one step, for instance by taking $K_{ij}^{lazy,(\sigma)} = \sum_{\alpha} (B_{\alpha\alpha})^{\sigma} e_{i\alpha} e_{j\alpha}$.

This general definition covers several existing models of random walks on hypergraphs. For $\sigma = 1$, we get the random walk defined in [20] while for $\sigma = -1$ we obtain the one introduced by Zhou [43], as

$$T_{ij}^{(-1)} = \frac{K_{ij}^{(-1)}}{\sum_{\ell \neq i} K_{i\ell}^{(-1)}} = \frac{1}{\sum_{\ell \neq i} K_{i\ell}^{(-1)}} \sum_{\alpha} \frac{e_{i\alpha} e_{j\alpha}}{(B_{\alpha\alpha} - 1)} \,,$$

and

$$\sum_{\ell \neq i} K_{i\ell}^{(-1)} = \sum_{\ell \neq i} \sum_{\alpha} \frac{e_{i\alpha} e_{\ell\alpha}}{(B_{\alpha\alpha} - 1)} = \sum_{\alpha} e_{i\alpha} = k_i \,,$$

where we used the fact $\sum_{\ell \neq i} e_{\ell\alpha}$ equals the number of nodes in E_α without node i, that is $(B_{\alpha\alpha} - 1)$, which thus simplifies with the denominator. The last equality defines k_i to be number of hyperedges incident to node i, a quantity that is simply the node degree in the case of networks. In conclusion,

$$T_{ij}^{(-1)} = \sum_{\alpha} \frac{e_{i\alpha} e_{j\alpha}}{k_i (B_{\alpha\alpha} - 1)} \tag{4}$$

has the following interpretation: the walker sitting on node i choses uniformly at random one hyperedge among the incident ones, i.e. with probability $1/k_i$, and then it selects a node belonging to the latter, again uniformly at random, i.e. with probability $1/(B_{\alpha\alpha} - 1)$.

The case $\sigma = 0$ returns a random walk on the so called *clique reduced multigraph*. The latter is a multigraph where each pair of nodes is connected by a number of edges equal to the number of hyperedges containing that pair in the hypergraph (see top left panel of Fig. 1). In that case, the transition matrix simplifies into

$$T_{ij}^{(0)} = \frac{K_{ij}^{(0)}}{\sum_{\ell \neq i} K_{i\ell}^{(0)}} = \frac{A_{ij}}{\sum_{\ell \neq i} A_{i\ell}} \,,$$

where we used the definition of the hyperadjacency matrix $A_{ij} = \sum_{\alpha} e_{i\alpha} e_{j\alpha}$. Let us observe that the clique reduced multigraph is different from the *projected network* obtained by associating to each hyperedge a clique of the same size; the projected network can be interpreted as the unweighted (binarised) version of the *clique reduced* multigraph (see bottom left panel of Fig. 1).

From the definitions (2) and (3), it is clear that hyperedges with larger sizes will dominate and, thus, set the fate of the random process for large values of σ. When σ is very negative, in contrast, hyperedges with a small size will drive the random walk process. As such, σ can be pictured as a *size bias parameter* which dictates the importance of hyperedges depending on their size.

Given the transition probability (3), a continuous-time random walk can be defined by

$$\dot{p}_i(t) = \sum_j p_j(t) T_{ji}^{(\sigma)} - \sum_j p_i(t) T_{ij}^{(\sigma)} \,,$$

where $p_i(t)$ is the probability of finding the walker on node i at time t. Note that $\mathbf{p} = (p_1, \ldots, p_n)$ is a row vector, as often assumed once dealing with Markov processes. Using $\sum_j T_{ij}^{(\sigma)} = 1$, the latter can be rewritten

$$\dot{p}_i = \sum_j p_j (T_{ji}^{(\sigma)} - \delta_{ij}) = -\sum_j p_j L_{ji}^{(\sigma)} \,, \tag{5}$$

where

$$L_{ij}^{(\sigma)} = \delta_{ij} - T_{ij}^{(\sigma)} = \delta_{ij} - \frac{K_{ij}^{(\sigma)}}{\sum_{\ell \neq i} K_{i\ell}^{(\sigma)}} \,, \tag{6}$$

is a *random walk* Laplacian generalising that of standard networks. Note that the standard Laplacian is recovered in the case $|E_\alpha| = 2$ for all α.

As this Laplacian, and its associated random walk process, can be interpreted as a standard random walk on the weighted undirected network encoded by the symmetric adjacency matrix $K_{ij}^{(\sigma)}$, standard results naturally generalise and we find that the stationary state is

$$p_j^{(\infty)} = \frac{d_j^{(\sigma)}}{\sum_\ell d_\ell^{(\sigma)}} \,, \qquad (7)$$

where $d_j^{(\sigma)} = \sum_{\ell \neq j} K_{j\ell}^{(\sigma)}$ is the strength of node j in the weighted graph. This latter quantity is an immediate generalisation of the standard node degree, in a direction which enables to account for the existence of different types of hyperedges in the system. Note that this interpretation of (6) as a random walk on the weighted projected network $K_{ij}^{(\sigma)}$ only holds for that projection, and not for other projections such as the clique reduced multigraph (see Figure 1). This observation is critical, as it allows us to easily adapt standard results from network science to the hypergraph framework passing by the weighted projected network $K_{ij}^{(\sigma)}$.

Remark 1 (Connection with bipartite works). *In bipartite graphs, nodes can be split into two disjoint sets and each link can only connect nodes belonging to different sets. In practice, we can associate to each set a different meaning/ feature, e.g. in a coauthorship network we can have a set of nodes representing authors while the other one would correspond to papers that they co-wrote. Or in the case of multivariate data, each object is a node of the first set and each feature denotes a node of the second set. Then a link exists between an object and a feature whenever the former possesses the latter. We can consider nodes and hyperedges in an hypergraph as two different sets of objects and represent the relationship "node i belongs to hyperedge E_α" as a connection between the node and the hyperedge, seen as two different objects. In this way, the incidence matrix (1) can be thought of as the adjacency matrix of a bipartite graph.*

Projecting hypergraphs thus finds connections with the well-studied problem of projecting bipartite networks. Important choices include

$$w_{ij}^{Newman} = \sum_\alpha \frac{e_{i\alpha} e_{j\alpha}}{B_{\alpha\alpha} - 1},$$

proposed to build scientific collaboration networks [45] and clearly equivalent to $K_{ij}^{(-1)}$ in Eq. (2). Other choices include [46]

$$w_{ij}^{Zhou} = \sum_\alpha \frac{e_{i\alpha} e_{j\alpha}}{k_i B_{\alpha\alpha}},$$

where the full size of the hyperedge is now taken into account, which corresponds to $\sigma = -1$ with a lazy walker. The formulas (2) and (3) are thus a natural generalisation of projection methods for bipartite networks as well.

III.   MARKOV STABILITY AND COMMUNITY DETECTION IN HYPERGRAPHS

A community, roughly speaking, is a set of nodes that display more connections between them than they do to nodes in other communities. The same idea can be used in the hypergraph setting where now the measure of the connectivity should take into account both the number of hyperdedges and their sizes. Different algorithms for community detection are based on random walk processes and, more specifically, on the intuition that walkers should stay for long times inside good communities before escaping them, the rationale being that the large number of links pointing to nodes inside the same community will reduce the chance to leave the group. As we have shown in the previous section, the size bias parameter allows us to bias the trajectories of random walkers or, equivalently, to give more or less weight to certain edges in the projected graph. This observation motivates the use of random walkers with different values of σ in order to search for communities giving more, or less, importance to edges belonging to large hyperedges. One expects that the resulting communities would differ from those obtained in the standard clique reduced multigraph, except for the particular case $\sigma = 0$.

In the following, we will search communities by generalising the flexible framework of Markov stability [15, 16]. Let us consider a partition of the nodes of a hypergraph into c (non overlapping) communities, encoded by the $n \times c$ indicator matrix $\mathbf{C}$ with $C_{ij} \in \{0, 1\}$, where a 1 denotes that node i belongs to community j and 0 otherwise. Given a partition $\mathbf{C}$, one can define the Markov stability

$$r(t; \mathbf{C}) := \min_{0 \leq s \leq t} tr\left[\mathbf{C}^{\mathrm{T}} \left(\mathbf{\Pi} e^{-s\mathbf{L}^{(\sigma)}} - (\mathbf{p}^{(\infty)})^{\mathrm{T}} \mathbf{p}^{(\infty)}\right) \mathbf{C}\right] = tr\left[\mathbf{C}^{\mathrm{T}} \left(\mathbf{\Pi} e^{-t\mathbf{L}^{(\sigma)}} - (\mathbf{p}^{(\infty)})^{\mathrm{T}} \mathbf{p}^{(\infty)}\right) \mathbf{C}\right], \qquad (8)$$
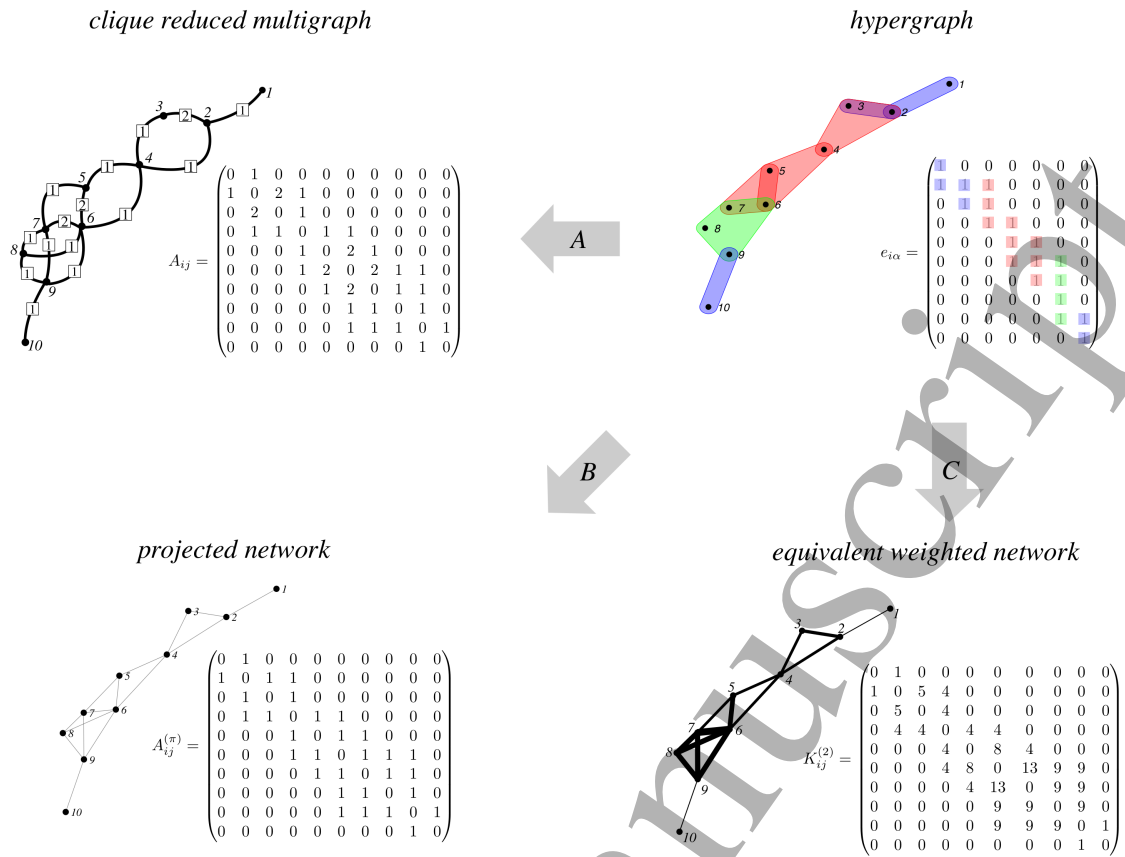
FIG. 1: **Hypergraph and their projection**. The top right panel shows a hypergraph, where hyperedges of different sizes are shown in different colours, the three 2-hyperedges are coloured in blue ($\{1,2\}$, $\{2,3\}$ and $\{9,10\}$), the three 3-hyperedges are coloured in red ($\{2,3,4\}$, $\{4,5,6\}$ and $\{5,6,7\}$) and the 4-hyperedge in green ($\{6,7,8,9\}$). Hyperedges do overlap and this "creates" new colours that are not associated to new hyperedges. Its structure is entirely encoded in the incidence matrix $e_{i\alpha}$ where, for ease of visualisation, we use the same colour code. Different projections can be constructed from the same hypergraph, each leading to a different weighted graph. In the clique reduced multigraph (top left panel), links between two nodes carry a weight proportional to the number of hyperedges including these two nodes. Alternatively, the so-called projected network is a binarised version of the clique reduction multigraph, so that a link exists between two nodes if they belong to at least one hyperedge. Finally, the family of projections $\mathbf{K}^{(\sigma)}$ defined in this paper, illustrated for the case $\sigma = 2$ at the bottom right, gives more or less weight to edges depending on the size of the hyperedges from which they are built.

where $\mathbf{\Pi}$ is the diagonal matrix containing $\mathbf{p}^{(\infty)}$ on the diagonal, $\mathbf{L}^{(\sigma)}$ is the above defined random walk Laplacian on the hypergraph and $(\mathbf{p}^{(\infty)})^{\mathrm{T}}\mathbf{p}^{(\infty)}$ is the matrix whose (i,j) entry is given by $p_i^{(\infty)}p_j^{(\infty)}$. Markov stability considers a random walk process at stationarity and it is made of two terms that are summed over all the communities of the partition. The first term measures the probability that a walker starts in a community and she is in the same community at time t. The second, negative term, that may be understood as a null model, measures the probability that two independent walkers belong to that community. An overview and thorough introduction to Markov stability can be found in [47].

Markov Stability $r(t; \mathbf{C})$ is a quality function quantifying the goodness of the partition $\mathbf{C}$ as a function of the time horizon of the random walk. One can thus determine, for any fixed t, the optimal partition $\mathbf{C}$, i.e. the one that maximises $r(t; \mathbf{C})$. The Markov stability can thus be used to rank partitions of a given graph at different time scales or, alternatively, as an objective function to be maximised for every time t in the space of all possible partitions of the hypergraph. We consider the latter and focus on a standard random walk on the weighted projected network defined by $K_{ij}^{(\sigma)}$, allowing us to use standard optimisation algorithms developed in [47, 48]. Let us recall that such optimisation problem is NP-hard. A large number of optimisation heuristics have been however developed to tackle the problem efficiently. The interested reader can consult for instance the works [9, 16, 47].

Remark 2 (Discrete time)**.** *Markov stability for random walks in discrete time is equivalent to the Newman-Girvan*

modularity [49] of the weighted adjacency matrix $K_{ij}^{(\sigma)}$, when $t = 1$ [47].

IV.   APPLICATIONS

In this section, we investigate the effect of σ on communities uncovered in artificial and real-life networks.

A.   A toy example: the two features hypergraph

The first example is a toy model, a hypergraph where nodes are endowed with two features, letters A, B or C, and numbers 1 or 2. The hypergraph is composed of 6 nodes, A_1, A_2, B_1, B_2, C_1 and C_2, and 5 hyperedges, 3 of size 2 connecting nodes with the same letter, and 2 of size 3 connecting nodes with the same number (see panel (a) of Fig. 2). The matrix $\mathbf{K}^{(\sigma)}$ is easily computed as hyperedges of size 2 contribute a weight of 1^σ in the weighted network, and hyperedges of size 3 a weight of 2^σ. The transition probabilities are thus given by:

$$T_{X_1,X_2} = \frac{1}{2 \times 2^\sigma + 1}, T_{X_1,Y_1} = \frac{2^\sigma}{2 \times 2^\sigma + 1}, T_{X_1,Y_2} = 0, T_{X_1,X_1} = 0, T_{X_2,X_2} = 0 \quad \forall X, Y \in \{A, B, C\}; \qquad (9)$$

thus

$$\lim_{\sigma \to \infty} T_{X_1,X_2} = 0, \lim_{\sigma \to \infty} T_{X_1,Y_1} = 1/2 \quad \forall X, Y \in \{A, B, C\},$$

namely for large σ, hops among nodes with the same number are strongly favoured and hence the walker will remain for a long time in the same 3-hyperedge. On the other hand,

$$\lim_{\sigma \to -\infty} T_{X_1,X_2} = 1, \lim_{\sigma \to -\infty} T_{X_1,Y_1} = 0 \quad \forall X, Y \in \{A, B, C\}$$

and the walker will thus spend longer periods of times in the 2-hyperedges. In the first case, an optimisation of Markov stability is expected to return 2 communities for sufficiently large Markov times, while it will produce 3 communities in the second case, as can be seen in Fig. 2. In panel (b), we report the number of communities in the optimal partition as a function of the Markov time for several values of σ. The results, combined in panel (c), reveal a qualitative change of the optimal partition depending on σ, as expected.

The number of communities only provides incomplete knowledge about a partition. We complement it by the Derrida and Flyvbjerg number [50], also known as the Simpson diversity index, defined as

$$Y = \sum_{i=1}^{M} \frac{S_i^2}{N^2}, \qquad (10)$$

where S_i is the number of elements in the i-th group. The quantity is a standard measure of how uniform a probability is distributed and ranges from $Y = 1$ when all the nodes belong to one single group to $Y = 1/N$ when there are $M = N$ groups, each one containing a single node. If the nodes are uniformly shared among the M groups, i.e. $S_i \sim N/M$, then $Y \sim 1/M$.

For each couple σ and Markov time, we report (see panel (d) in Fig. 2) the value of $1/Y$. Here $N = 6$ and $M = 2$ or $M = 3$ (excluding the trivial case $M = 6$); in the former case, assuming the nodes to be equally shared among the two communities, we get $Y = 2 \times (3/6)^2 = 1/2$, while in the latter case we obtain $Y = 3 \times (2/6)^2 = 1/3$. One can observe that indeed $1/Y = 2$ (dark blue zone) for the same set of values for which the number of communities equals 2, and $1/Y = 3$ (light blue zone) for parameters corresponding to 3 communities.

To conclude this section, we investigate whether the size k of a hyperedge increases or decreases its likelihood to be inside a community, depending on σ. The results reported in Fig. 2 show a sharp transition at $\sigma = 0$ corresponding to a structural change in the communities detected; for negative σ the random walker will spend more time in the small size hyperedges and thus the nodes will be partitioned into communities of size 2 corresponding to the 2-hyperedges, i.e. $c_1 = \{A_1, A_2\}, \{B_1, B_2\}$ and $\{C_1, C_2\}$. On the other hand if $\sigma > 0$ the large size hyperedges will "capture" the walker for long times and thus the network will be partitioned into large size communities associated to the 3-hyperedges, i.e. $\{A_1, B_1, C_1\}$ and $\{A_2, B_2, C_2\}$.

By taking advantage of the simple structure of the hypergraph under scrutiny, we can analytically explain the existence of the abrupt transitions shown in Fig. 2. Indeed one can explicitly compute the time evolution of the random walk process by solving Eq. (5). This implies computing $e^{-t\mathbf{L}^{(\sigma)}}$ for all σ and the stationary solution $\mathbf{p}^{(\infty)}$ (see
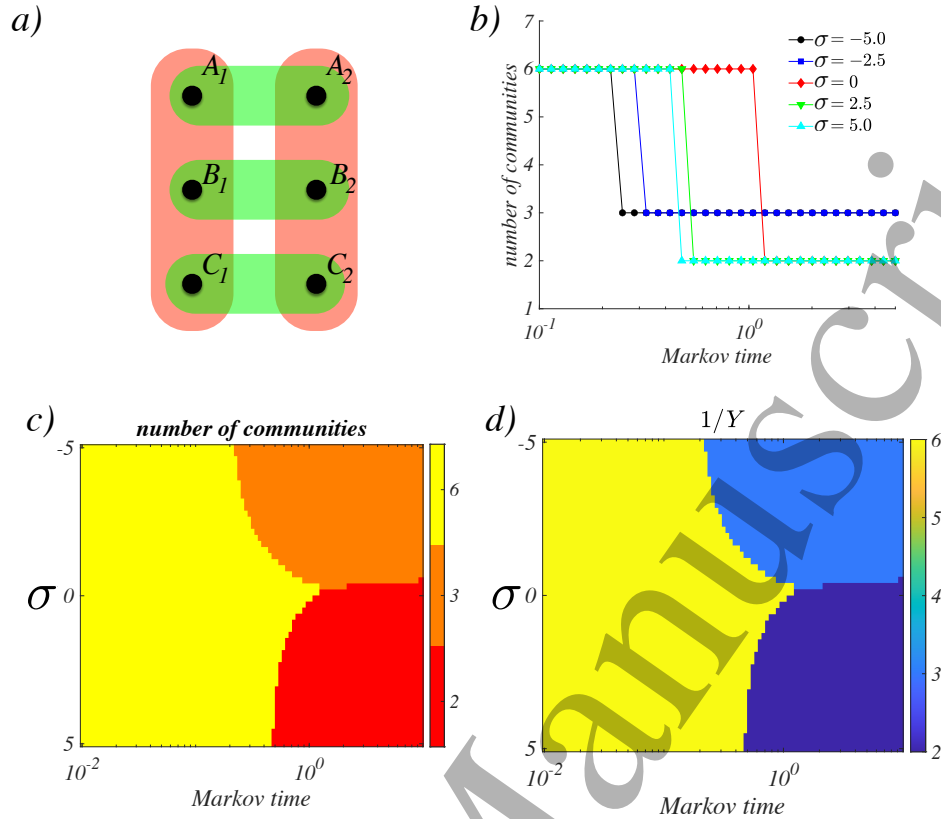
FIG. 2: **Two features hypergraph**. Panel (a): The hypergraph is made by 6 nodes and 5 hyperedges, two of size 3 (red ones) and three of size 2 (green ones). The former can be thought to represent the feature "nodes with the same number" while the former "nodes with the same letter". Panel (b): the number of communities versus the Markov time for several values of the size bias parameter σ; as expected from the theory, for large Markov time and large positive σ the SF detects 2 communities, i.e. the two 3-hyperedges, while for small negative σ the method reports 3 communities, i.e. the three 2-hyperedges. Panel (c): a global view of the transition between the number of detected communities as a function of σ and the Markov time. Panel (d): we report the inverse of the Derrida-Flyvbjerg number as a function of σ and the Markov time, namely a proxy of the composition of the communities.

Appendix A). One can consequently calculate the Markov stability for the three spontaneously emerging communities by looking at the hypergraph, that is by grouping together nodes with the same letter (resulting into 3 groups), nodes with the same number (resulting into 2 groups) and each one alone (6 groups). Each structure can be characterised by a suitable indicator matrix $\mathbf{C}$ (see Eq. (A2)), eventually allowing us to obtain $r(t; \mathbf{C})$ for all $t \geq 0$. Given this explicit function of time we can (numerically) compute the transition times, that is the time instants at which a structural change arises in the random walk process and one optimal partition is replaced by a better one.

B.　Hierarchical network

Our second example is directly inspired by the weighted hierarchical network presented in Fig. 1 of [16]. The hypergraph contains 16 node and 15 hyperedges (see panel (a) Fig. 3) and it has an hierarchical structure in terms of hyperedges. More precisely, 8 hyperedges (in blue in the Figure) contain 2 nodes, hence 4 hyperedges (in red in

the Figure) of size 4 are made, each one obtained by merging 2 hyperedges of size 2. Then the process is repeated, 2 hyperedges (in yellow in the Figure) of size 8 are created each one containing all the nodes of 2 hyperedges of size 4. And finally a large hyperedge (in grey in the Figure) of size 16 is made with all the nodes. Let us observe that the projected network is a complete network made by 16 nodes.

We then optimise the Markov stability to determine, as a function of the Markov time, $t \in [10^{-2}, 10]$, and the size bias parameter σ, the optimal partition of nodes into communities. The results presented in the panel (b) of Fig. 3 clearly show that the method is able to detect the hierarchical structure of the communities, that is starting with 16 isolated nodes for very short Markov time, the method is able to capture the intermediate coarse grained structures made by 8, 4 and 2 communities as the Markov time increases. A straightforward application of the definition (2) allows to obtain the following values for the matrix $\mathbf{K}^{(\sigma)}$:

$$K_{12}^{(\sigma)} = 1 + 3^{\sigma} + 7^{\sigma} + 15^{\sigma}, K_{13}^{(\sigma)} = K_{14}^{(\sigma)} = 3^{\sigma} + 7^{\sigma} + 15^{\sigma}$$
$$K_{15}^{(\sigma)} = K_{16}^{(\sigma)} = K_{17}^{(\sigma)} = K_{18}^{(\sigma)} = 7^{\sigma} + 15^{\sigma} \text{ and } K_{1j}^{(\sigma)} = 15^{\sigma} \quad \forall j = 9, \dots, 16,$$

the idea being that the larger is the second index, j, the smaller is the number of hyperedges containing both i and j, but with increasing sizes. For instance $i = 1$ and $j = 2$ belong to 4 hyperedges whose sizes are 2, 4, 8 and 16, while $i = 1$ and $j = 5$ belong to 2 hyperedges whose sizes are 8 and 16. By symmetry on can easily compute all the remaining entries $K_{ij}^{(\sigma)}$.

Then recalling the definition of the transition probability (3) one can prove that in the limit of extremely large σ we get:

$$\lim_{\sigma \to \infty} T_{ij}^{(\sigma)} = \frac{1}{15} \quad \forall i, j \in \{1, \dots, 16\},$$

that is the random walker executes jumps among nodes with uniform probability, regardless of the hyperedge size. On the other hand we can prove that

$$\lim_{\sigma \to -\infty} T_{12}^{(\sigma)} = 1 \text{ and } \lim_{\sigma \to -\infty} T_{1j}^{(\sigma)} = 0 \quad \forall j \in \{1, \dots, 16\} \setminus \{2\},$$

and similarly for nodes belonging to the remaining 2-hyperedges. Hence the process forces the walker to remain in the smallest hyperedges, i.e. those of size 2.

Starting from these observations, we can explain the results reported in Fig. 3. In Panel (c) we present the number of communities detected for each couple σ and Markov time; one can observe that for large positive σ, the algorithm returns the finest partition, i.e. made of 16 communities (green zone), up to long Markov time and then suddenly the coarsest one made by 2 large communities (red zone). This result is intuitive as, for large σ, the transition probability are uniform. Only negative σ allow to explore the intermediate partitions (yellow and orange zones) and eventually end up with the partition into 8 groups of size 2, as predicted by the behaviour of the transition probability in the limit $\sigma \to -\infty$.

Let us observe that the information reported in panels (b) and (c) of Fig. 3 is limited to the number of communities. We could have used the optimal indicator matrices obtained from the Markov stability framework to access the groups composition and thus highlight the hierarchical structure. We decided however to resort again to the diversity index Y and exploit the hypergraph symmetry to explicitly compute the latter and thus infer the size of the communities. For this reason the information about the number of communities is complemented with the diversity index in panel (d) of Fig. 3. The value $1/Y = 2$ (dark blue zone) corresponds to $M = 2$ communities each made by 8 nodes, indeed $Y = 2 \times (8/16)^2 = 1/2$, the value $1/Y = 4$ (light blue zone) is associated to $M = 4$ communities each made by 4 nodes, $Y = 4 \times (4/16)^2 = 1/4$. Finally the region associated to $1/Y = 8$ (cyan zone) corresponds to $M = 8$ and 2 nodes per group, $Y = 8 \times (2/16)^2 = 1/8$.

C. Animals hypernetwork

The third example is based on the dataset containing an ensemble of animals from a zoologically heterogeneous set, taken from the UCI Machine Learning Depository [51]. The dataset is made of 101 animals, each one endowed with 20 features, such as tail, hair, legs and so on [20]. For each animal we know the ground truth, that is its corresponding class, e.g. Mammal (41 elements), Bird (20 elements), Reptile (5 elements), Fish (13 elements), Amphibian (4 elements), Bug (8 elements) and Invertebrate (10 elements). Here nodes are animals and hyperedges features; the goal is to use the random walk to cluster "similar animals" into communities. In Fig. 4 we report the community structure obtained by optimising Markov stability as a function of the size bias parameter σ and the
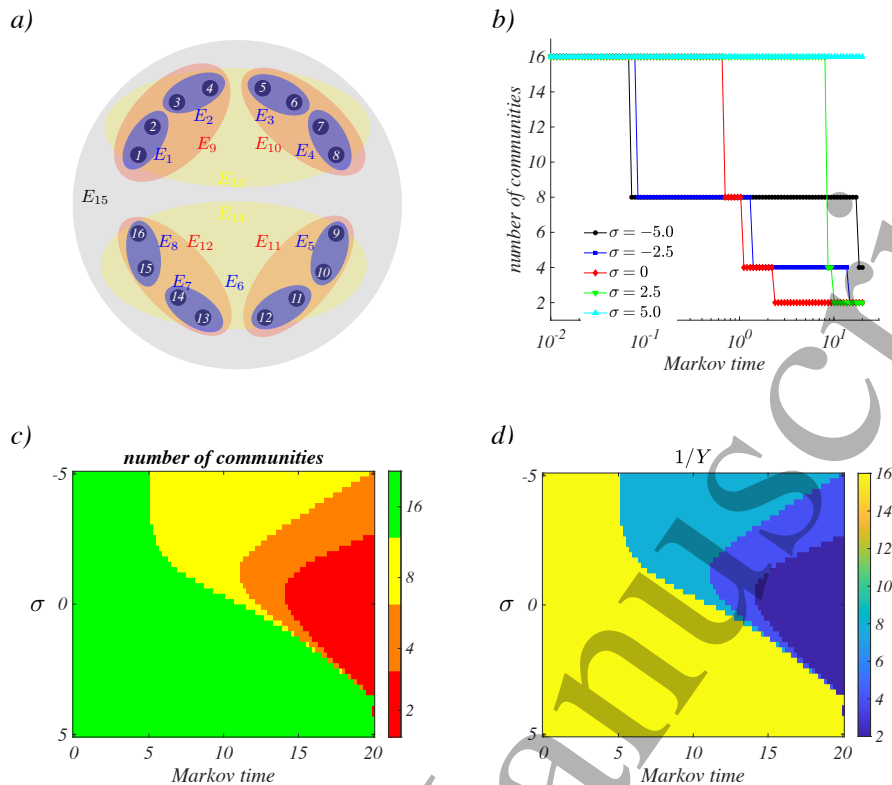
FIG. 3: **Hierarchical hypergraph**. Panel (a): The hypergraph is made by 16 nodes and 15 (non simple) hyperedges. There are 8 hyperedges of size 2 (blue), 4 hyperedges of size 4 (red), 2 hyperedges of size 4 (yellow) and 1 hyperedge of size 16 (grey). Panel (b): the number of communities versus the Markov time for several values of the size bias parameter σ. Panel (c): a global view of the transition between the number of detected communities as a function of σ and the Markov time. One can observe that for large positive σ the SF exhibits a sharp transition between 16 communities, each made by a single node, to 2 communities corresponding to the two 8-hyperedges. On the other hand negative σ allow to explore the intermediate structures passing from 2-hyperedges, 4-hyperedges and then 8-hyperedges. Panel (d): the proxy for the composition of the communities, $1/Y$.

Markov times. On the panel (a) we report the number of communities as a function of the Markov time for several values of $\sigma \in \{-5, -2, -1, 0, 1, 2, 5\}$, while in the panel (b) we present a more global view. From both panels one can appreciate the nonlinear interplay between σ and the number of communities detected at a given Markov time; indeed for small enough (negative) σ, the number of detected communities is small, for all the considered Markov times. On the other hand, for a fixed Markov time, increasing σ allows to sharply pass from few to many communities, whose sizes are presented in the panel (d).

Finally in panel (c), we compare the community structure obtained for given σ and Markov time, with the ground truth, i.e. the known classes to which any animal belongs to. To do this we used the *Adjusted Rand Index* (ARI) [52], a method allowing to compare two partitions of the elements of a given set. The larger is the index the more similar are the two partitions. The ARI is an improvement of the Rand index adjusted for the chance grouping of elements. Our numerical results reveal that optimal values of ARI are obtained for complex combinations of the size bias parameter σ and Markov time, hence motivating the possibility to tune these parameters in real-world settings.

With the results presented in Fig. 4 we propose a sort of aggregated information about the database. Indeed the number of communities or the Derrida and Flyvbjer number are not capable to unravel the hidden structure of the actual communities. A fine description of the zoo database is beyond the scope of this work, however to make one step further in this direction, we present in the Appendix B a preliminary analysis of the community composition as

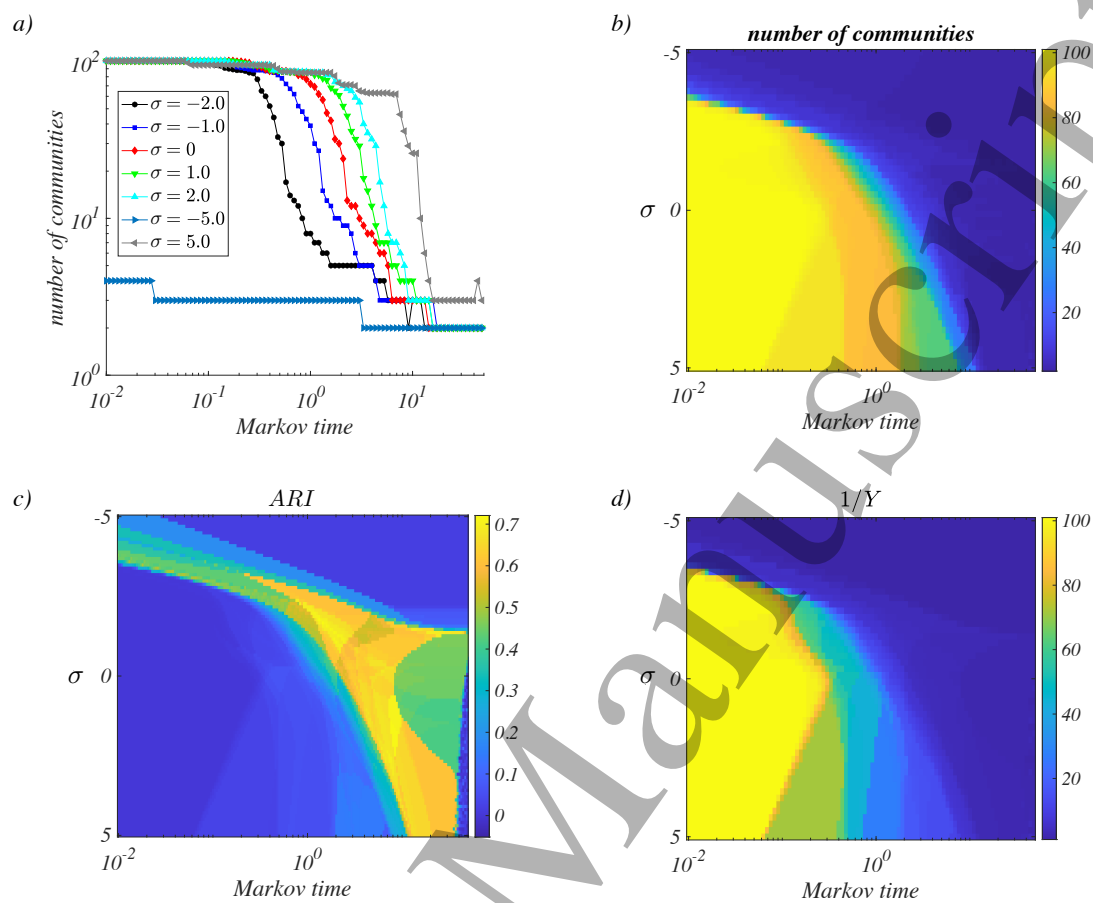a function of the Markov time for a fixed value of the size bias parameter σ.

FIG. 4: **Zoo database**. Panel (a): number of communities versus Markov time for several values of σ. Panel (b): number of communities as a function of σ and Markov time. Panel (c) Computed communities versus the ground truth measured using the ARI, larger values (corresponding to yellowish color) are associated to a good matching among the two partitions. Panel (d): composition of the communities measured with the inverse of the Derrida and Flyvbjer number, as a function of σ and Markov time.

V. CONCLUSIONS

In this article, we have considered a family of random walks on hypergraphs with a parameter controlling the bias of the dynamics towards hyperedges of low or high size. As we have shown, the process naturally provides different ways to project hypergraphs on networks and includes standard approaches like the clique-reduced multigraph as special cases. The resulting projections may radically differ depending on the size bias parameter and we have explored this dependency through its effect on community structure. Building on Markov stability, we have developed a general framework to uncover communities in hypergraphs, which we have tested on artificial and real-world networks. For future research, interesting questions include the determination of appropriate values of the size bias parameter to unravel interesting patterns in empirical data, as well as a more thorough comparison of the weighted projections, for instance with graph distance measures [53] or their impact for other flow-based network metrics such as the Map Equation or PageRank.

Appendix A: Computation of the Markov stability for the "two features hypergraph"

The aim of this section is to exploit the simple structure of the two features hypergraph presented in Section IV A (see also panel (a) of Fig. 2) to explicitly compute the Markov stability $r(t; \mathbf{C})$ as a function of the Markov time t as well as the size bias parameter σ. In this way we will be able to explain the abrupt switches in the number of detected communities, as shown in the panel (b) of Fig. 2.

Recalling the definition of the generalised random walk Laplace operator (6) and the transition probabilities computed in Eq. (9), we can obtain

$$\mathbf{L}^{(\sigma)} = \begin{pmatrix} 1 & b & b & a & 0 & 0 \\ b & 1 & b & 0 & a & 0 \\ b & b & 1 & 0 & 0 & a \\ a & 0 & 0 & 1 & b & b \\ 0 & a & 0 & b & 1 & b \\ 0 & 0 & a & b & b & 1 \end{pmatrix}, \tag{A1}$$

where

$$a = -\frac{1}{1 + 2^{\sigma+1}} \quad \text{and} \quad b = -\frac{2^{\sigma}}{1 + 2^{\sigma+1}}.$$

Let us observe that by construction $1 + a + 2b = 0$.

By using again the explicit form of $K_{j\ell}^{(\sigma)}$ and the nodes symmetries (each node is connected to the node with the same letter but different number and to the two nodes with different letter but same number), one can compute the stationary state Eq. (7) that is found to be

$$p_j^{(\infty)} = \frac{1}{6}.$$

We are hence in a position to compute the matrices $\mathbf{\Pi}$ and $(\mathbf{p}^{(\infty)})^{\mathrm{T}} \mathbf{p}^{(\infty)}$ needed in Eq. (8)

$$\mathbf{\Pi} = \frac{1}{6} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad (\mathbf{p}^{(\infty)})^{\mathrm{T}} \mathbf{p}^{(\infty)} = \frac{1}{36} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

The last step to compute the Markov stability is to evaluate $e^{-s\mathbf{L}^{(\sigma)}}$. This can be straightforwardly obtained by computing the eigenvalues and eigenvectors of the Laplace matrix, namely $\mathbf{L}^{(\sigma)} \mathbf{\Phi} = \mathbf{\Phi} \mathbf{\Lambda}$ where

$$\mathbf{\Phi} = \begin{pmatrix} 1 & -1 & -1 & 1 & 1 & -1 \\ 1 & 1 & 0 & -1 & 0 & -1 \\ 1 & 0 & 1 & 0 & -1 & -1 \\ 1 & -1 & -1 & -1 & -1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{\Lambda} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & a-b+1 & 0 & 0 & 0 & 0 \\ 0 & 0 & a-b+1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1-a-b & 0 & 0 \\ 0 & 0 & 0 & 0 & 1-a-b & 0 \\ 0 & 0 & 0 & 0 & 0 & 1-a+2b \end{pmatrix}.$$

For the sake of definitiveness we will compute the Markov stability for the three natural communities listed in the following:

- 3 communities, each one composed of nodes with the same letter;

- 2 communities, each one composed of nodes with the same number;

- 6 communities, each one composed of a single node.

Each case will be determined by defining an appropriate indicator matrix $\mathbf{C}$:

$$\mathbf{C}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \mathbf{C}_2 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{C}_3 = \mathbf{1}, \tag{A2}$$

assuming to order nodes as A_1, B_1, C_1, A_2, B_2, C_2.

Wrapping together these intermediate steps we are now able to explicitly compute the Markov stability $r(t; \mathbf{C}_i)$ for $i = 1, 2, 3$:

$$r_1 \equiv r(t; \mathbf{C}_1) = \frac{2}{3} e^{-t \frac{3\,2^\sigma}{1+2^{\sigma+1}}}, r_2 \equiv r(t; \mathbf{C}_2) = \frac{1}{2} e^{-t \frac{2}{1+2^{\sigma+1}}} \text{ and } r_3 \equiv r(t; \mathbf{C}_3) = \frac{r_1}{2} + \frac{r_2}{3} + r_1 r_2. \quad (A3)$$

In Fig. 5 we report the above three functions for three values of σ (-5 left panel, 0 middle panel and 5 right panel). For each fixed Markov time, the optimal partition is the one which displays the larger value of r_i; hence the intersection points correspond to a new arrangement of nodes into groups with a possible abrupt variation into the number of communities detected (see Fig. 6), to be compared with Fig. 2. We can observe (see left panel of Fig. 5) that in the case $\sigma = -5$ the process identifies 6 communities for small enough Markov times ($r_3(t) > \max\{r_1(t), r_2(t)\}$) to then pass at $t \sim 0.2188$ to 3 communities ($r_2(t) > \max\{r_1(t), r_3(t)\}$). This transition corresponds to the abrupt jump of the curve corresponding to $\sigma = -5$ in Figs. 6 and 2. A similar behaviour holds true for $\sigma = 5$ (see right panel of Fig. 5): in this case the 6 community state leaves its place to the 2 communities one at $t \sim 0.4743$.
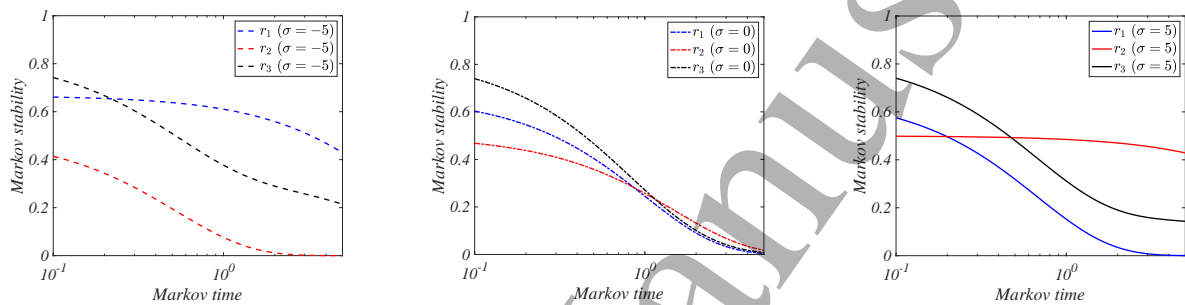

FIG. 5: **Markov stability**. We report the exact Markov stability for the three community structures identified by the indicator matrices (A2) ($\mathbf{C}_1$ in blue, $\mathbf{C}_2$ in red and $\mathbf{C}_3$ in black). The left panel corresponds to $\sigma = -5$, the middle panel to $\sigma = 0$ and the right one to $\sigma = 5$.

Using the explicit formulas for the functions $r(t; \mathbf{C}_i)$ given by Eq. (A3) one can compute the Markov times at which two of such functions assume the same value. For instance, taking $\sigma = -5$ one can compute the value of t such that $r_1(t) = r_3(t)$ (see left panel of Fig. 5):

$$r_1(t) = r_3(t) = \frac{r_1(t)}{2} + \frac{r_2(t)}{3} + r_1(t) r_2(t) \Rightarrow \frac{r_1(t)}{2} = \frac{r_2(t)}{3} + r_1(t) r_2(t),$$

Being $r_i > 0$ we can divide by $r_1 r_2$ and thus obtain the transcendental equation

$$\frac{x^{(3\,2^{\sigma-1})}}{2} + 1 = x,$$

where $x = e^{t \frac{2}{1+2^{\sigma+1}}}$, whose solution can be numerically computed to give $x_1 \sim 1.5097$ corresponding to $t_1 \sim 0.2188$. Similarly for $\sigma = 5$ one can obtain an intersection between $r_2(t)$ and $r_3(t)$ at $t_2 = 0.4743$.

Appendix B: Community structure of the animal hypernetwork

The aim of this section is to explore the structure of the communities as computed by using the Markov stability in the zoo database and thus to complement the information provided in Fig. 4. We thus set $\sigma = 1$ and select seven representative Markov times, i.e. epochs at which the number of community stays (locally) constant, denoted with the letters A to G, and we show how animals, namely nodes, merge to create fewer and larger communities (see Fig. 7). In the left panel of the latter figure, we show the number of communities detected for the choice $\sigma = 1$, the blue symbols represent the chosen Markov times. This is the same curve as reported in green in panel (a) of Fig. 4.
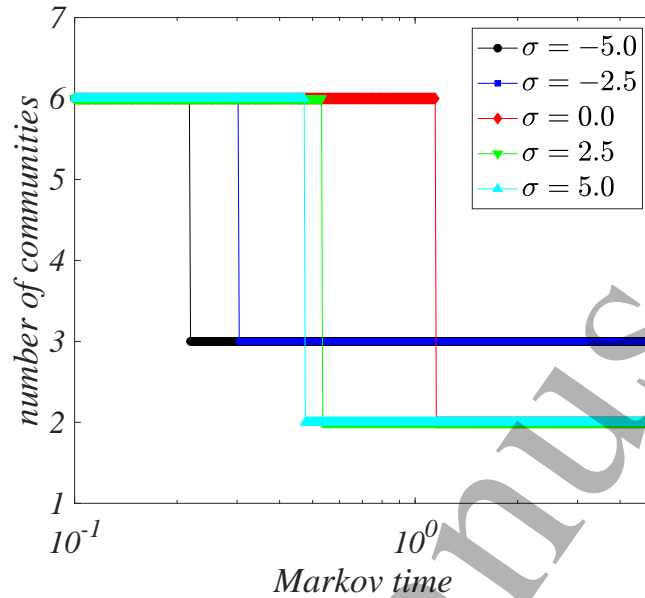
FIG. 6: **Number of communities obtained with Markov stability**. We report the number of communities determined by using the exact Markov stability. The used values of σ as well as the symbols are the same as those employed in Fig. 2.

Let us comment the obtained communities and their structure. The isolated community (on the top of the right panel) lasting up to time F, when it merges with the larger community, contains a single animal, the platypus; because of the peculiarities of this animal we can easily understand that it can be hardly classified.

At time C all Mammals, but the platypus, the dolphin and the porpoise, are correctly set in the same community, the latter two being set in the group of Fish. The method clusters animals in 3 large groups, the one (38 elements) corresponding to Mammal, the one (32 elements) containing all the Birds (and few more other animals, e.g. 8 Bugs some of which do fly such as, flea, gnat, honeybee, housefly, ladybird, moth, wasp) and the one (26 elements) containing all the Fish (and few more other animals, e.g. 7 Invertebrate living in the sea such as, clam, crab, crayfish, lobster, octopus, seawasp and starfish).

From time C to time D the smaller classes are merged into the larger ones except for the platypus that keeps being alone. Passing from D to E the two non-Mammal classes do merge and we are thus facing with a classification Mammals versus non-Mammals (and still the platypus left alone). The platypus is set with the non-Mammals at time F and eventually all the animals are grouped together at time G.

[1] R. Albert and A.-L. Barabási, Reviews of modern physics **74**, 47 (2002).
[2] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, Physics Reports **424**, 175 (2006).
[3] M. E. Newman, *Networks: An Introduction* (Oxford University Press, Oxford, 2010).
[4] A.-L. Barabási et al., *Network science* (Cambridge university press, 2016).
[5] V. Latora, V. Nicosia, and G. Russo, *Complex networks: principles, methods and applications* (Cambridge University Press, 2017).
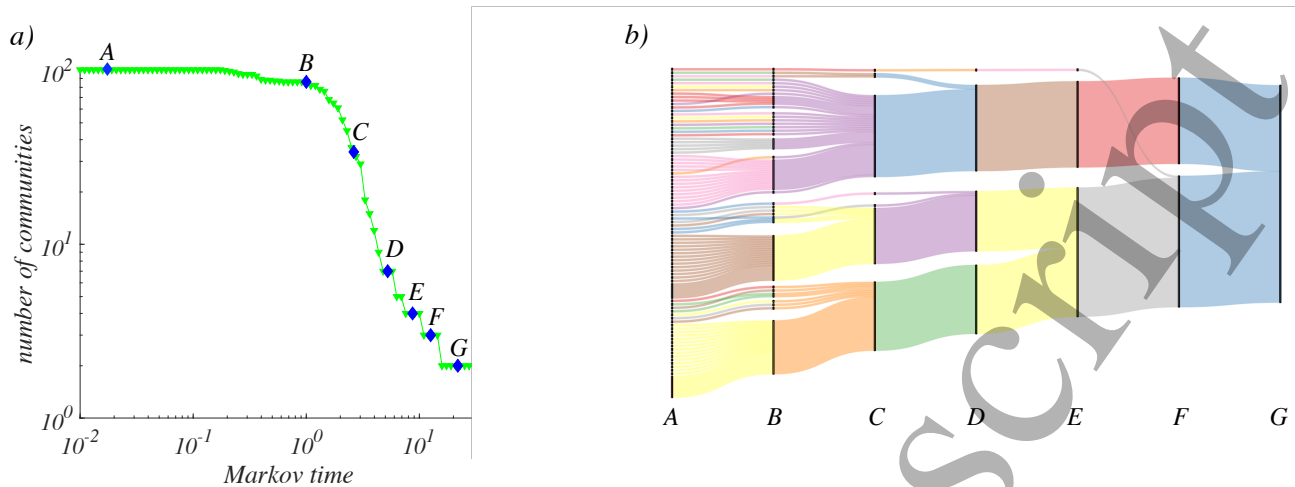
FIG. 7: **Zoo database**. Left panel: the number of communities versus the Markov time. Right panel: how nodes share among the communities (vertical black bars) versus the Markov time.

[6] C. Castellano, S. Fortunato, and V. Loreto, Reviews of modern physics **81**, 591 (2009).
[7] A. Arenas, A. Díaz-Guilera, J. Kurths, Y. Moreno, and C. Zhou, Physics reports **469**, 93 (2008).
[8] A. Barrat, M. Barthlemy, and A. Vespignani, *Dynamical Processes on Complex Networks* (Cambridge University Press, New York, 2001).
[9] M. T. Schaub, J.-C. Delvenne, R. Lambiotte, and M. Barahona, Advances in Network Clustering and Blockmodeling pp. 333–361 (2019).
[10] S. Redner, *A guide to first-passage processes* (Cambridge University Press, 2001).
[11] S. Fortunato and C. Castellano, *Community Structure in Graphs* (Springer New York, 2012).
[12] M. T. Schaub, J.-C. Delvenne, R. Lambiotte, and M. Barahona, Phys. Rev. E **99**, 062308 (2019).
[13] M. T. Schaub, N. O'Clery, Y. N. Billeh, J.-C. Delvenne, R. Lambiotte, and M. Barahona, Chaos **26**, 094821 (2016).
[14] M. Rosvall and C. T. Bergstrom, Proceedings of the National Academy of Sciences **105**, 1118 (2008).
[15] J.-C. Delvenne, S. N. Yaliraki, and M. Barahona, Proceedings of the National Academy of Sciences **107**, 12755 (2010).
[16] R. Lambiotte, J.-C. Delvenne, and M. Barahona, IEEE Transactions on Network Science and Engineering **1**, 76 (2014).
[17] A. R. Benson, D. F. Gleich, and J. Leskovec, Science **353**, 163 (2016).
[18] R. Lambiotte, M. Rosvall, and I. Scholtes, Nat. Phys. **15**, 313 (2019).
[19] A. Patania, G. Petri, and F. Vaccarino, EPJ Data Sci. **6**, 18 (2017).
[20] T. Carletti, F. Battiston, G. Cencetti, and D. Fanelli, Physical Review E **101**, 022308 (2020).
[21] G. Petri, P. Expert, F. Turkheimer, R. Carhart-Harris, D. Nutt, P. J. Hellyer, and F. Vaccarino, Journal of The Royal Society Interface **11**, 20140873 (2014).
[22] L.-D. Lord, P. Expert, H. Fernandes, G. Petri, T. Van Hartevelt, F. Vaccarino, G. Deco, F. Turkheimer, and M. Kringelbach, Front. Syst. Neurosci. **10**, 85 (2016).
[23] E. Estrada and G. Ross, J. Their. Biol. **438**, 46 (2018).
[24] J. Grilli, G. Barabás, M. J. Michalska-Smith, and S. Allesina, Nature **548**, 210 (2017).
[25] K. Devriendt and P. Van Mieghem, Journal of Complex Networks **7**, 469 (2019).
[26] O. T. Courtney and G. Bianconi, Physical Review E **93**, 062311 (2016).
[27] G. Petri and A. Barrat, Physical Review Letters **121**, 228301 (2018).
[28] C. Berge, *Graphs and hypergraphs*, North-Holland Pub. Co. (American Elsevier Pub. Co, 1973).
[29] E. Estrada and J. A. Rodríguez-Velázquez, arXiv preprint physics/0505137 (2005).
[30] G. Ghoshal, V. Zlatić, G. Caldarelli, and M. E. Newman, Physical Review E **79**, 066118 (2009).
[31] A. Bodó, G. Katona, and P. Simon, Bull. Math. Biol. **78**, 713 (2016).
[32] I. Iacopini, G. Petri, A. Barrat, and V. Latora, Nature communications **10**, 2485 (2019).
[33] M. Lucas, G. Cencetti, and F. Battiston, arXiv preprint arXiv: 2003.09734v1 (2020).
[34] L. V. Gambuzza, F. Di Patti, G. L., S. Lepri, M. Romance, R. Criado, M. Frasca, V. Latora, and S. Boccaletti, arXiv preprint arXiv:2004.03913v1 (2020).

[35] G. F. de Arruda, G. Petri, and Y. Moreno, Phys. Rev. Research **2**, 023032 (2020).
[36] G. Ferraz de Arruda, M. Tizzani, and Y. Moreno, arXiv preprint arXiv:2005.10891 (2020).
[37] A. Krawiecki, Chaos, Solitons and Fractals **65**, 44 (2014).
[38] R. Mulas, C. Kuehn, and J. Jost, arXiv preprint arXiv: 2003.13775v1 (2020).
[39] T. Carletti, D. Fanelli, and S. Nicoletti, J.Phys.Complex. **1**, 035006 (2020).
[40] L. Neuhäuser, A. Mellor, and R. Lambiotte, Physical Review E **101**, 032310 (2020).
[41] L. Lu and X. Peng, in *International Workshop on Algorithms and Models for the Web-Graph* (Springer, 2011), pp. 14–25.
[42] A. Helali and M. Löwe, Statistics and Probability Letters **154**, 1 (2019).
[43] D. Zhou, J. Huang, and B. Schölkopf, in *Advances in neural information processing systems* (2007), pp. 1601–1608.
[44] T. S. Evans, Journal of Statistical Mechanics: Theory and Experiment **2010**, P12037 (2010), URL `https://doi.org/10.1088%2F1742-5468%2F2010%2F12%2Fp12037`.
[45] M. E. Newman, Physical Review E **64**, 016132 (2001).
[46] T. Zhou, J. Ren, M. Medo, and Y.-C. Zhang, arXiv preprint arXiv:0707.0540v2 (2020).
[47] J.-C. Delvenne, M. Schaub, S. N. Yaliraki, and M. Barahona, in *Dynamics On and Of Complex Networks (II)* (2013), p. 221.
[48] M. Schaud, J.-C. Delvenne, S. N. Yaliraki, and M. Barahona, PLoS ONE **7**, e32210 (2012).
[49] M. E. Newman, Proceedings of the national academy of sciences **103**, 8577 (2006).
[50] B. Derrida and H. Flyvbjerg, J. Phys. A **19**, 1003 (1986).
[51] D. Dua and C. Graff, *UCI machine learning repository* (2017), URL `https://archive.ics.uci.edu/ml/datasets/Zoo`.
[52] L. Hubert and P. Arabie, Journal of Classification **2**, 193 (1985).
[53] C. Donnat, S. Holmes, et al., The Annals of Applied Statistics **12**, 971 (2018).