communications physics

ARTICLE

https://doi.org/10.1038/s42005-023-01270-5 OPEN

Compressing network populations with modal networks reveal structural diversity

Alec Kirkley [1,2,3]✉, Alexis Rojas [4], Martin Rosvall[5] & Jean-Gabriel Young [6,7]

Analyzing relational data consisting of multiple samples or layers involves critical challenges: How many networks are required to capture the variety of structures in the data? And what are the structures of these representative networks? We describe efficient nonparametric methods derived from the minimum description length principle to construct the network representations automatically. The methods input a population of networks or a multilayer network measured on a fixed set of nodes and output a small set of representative networks together with an assignment of each network sample or layer to one of the representative networks. We identify the representative networks and assign network samples to them with an efficient Monte Carlo scheme that minimizes our description length objective. For temporally ordered networks, we use a polynomial time dynamic programming approach that restricts the clusters of network layers to be temporally contiguous. These methods recover planted heterogeneity in synthetic network populations and identify essential structural heterogeneities in global trade and fossil record networks. Our methods are principled, scalable, parameter-free, and accommodate a wide range of data, providing a unified lens for exploratory analyses and preprocessing large sets of network samples.

[1] Institute of Data Science, University of Hong Kong, Hong Kong SAR, China. [2] Department of Urban Planning and Design, University of Hong Kong, Hong Kong SAR, China. [3] Urban Systems Institute, University of Hong Kong, Hong Kong SAR, China. [4] Department of Computer Science, University of Helsinki, Helsinki 00100, Finland. [5] Integrated Science Lab, Department of Physics, Umea University, SE-901 87 Umea, Sweden. [6] Department of Mathematics and Statistics, University of Vermont, Burlington, VT, USA. [7] Vermont Complex Systems Center, University of Vermont, Burlington, VT, USA. ✉email: alec.w.kirkley@gmail.com

A common way to measure a network is to gather multiple observations of the connectivity of the same nodes. Examples include the mobility patterns of a particular group of students encoded as a longitudinal set of co-location networks[1,2], measurements of connectivity among the same brain regions for different individuals[3], or the observation of protein-protein relationships through a variety of different interaction mechanisms[4]. These measurements can be viewed as a multilayer network[5] consisting of one layer for each measurement of all links between the nodes. For generality, we consider them as a *population of networks*—a set of independent network measurements on the same set of nodes, either over time or across systems with consistent, aligned node labels. There often are regularities among such collections of measurements, but each sample may differ substantially from the next. Summarizing these measurements with robust statistical analyses can separate regularities from noise and simplify downstream analyses such as network visualization or regression[6–15].

Most statistical methods for summarizing populations of networks share a similar approach. They model all the members of a population as realizations of a single representative network[6,9,13,16–18], which can be retrieved by fitting the model in question to the observed population. However, the strong assumption that a single "modal" network best explains the observed populations can lead to a poor representation of the data at hand[19,20]. For instance, accurately modeling a population of networks recording face-to-face interactions between elementary school pupils requires at least two representative networks if the data include networks observed during class and recess[21]. Modeling the measurements with a single network will most likely neglect essential variations in the pupil's face-to-face interactions, leading to similar oversights from summarizing a multimodal probability distribution with only its mean.

Recent research has examined related problems and led to, for example, methods for detecting abrupt regime changes in temporal series of networks[22,23], pooling information across subsets of layers of multiplex networks[24] and embedding nodes in common subspaces across network layers[11,25,26]. Several recent contributions have addressed the problem of summarizing populations of networks when multiple distinct underlying network representations are needed, using mixtures of parametric models[20,27–30], latent space models[31], or generative models based on ad hoc graph distance measures[19].

These methods cluster network populations with good performance but have some significant drawbacks. None of the methods discussed, except ref. [19], outputs a single sparse representative network for each cluster but requires handling ensembles of network structures, making downstream applications such as network visualization or regression cumbersome. Most of these methods also require potentially unrealistic modeling assumptions about the structure of the clusters. For example, that stochastic block models or random dot product graphs can model all network structures in the clusters. Specifying a generative model for the modal structures also has the downside of often requiring complex and time-consuming methods to perform the within-cluster estimation. Perhaps most critically, existing approaches require either specifying the number of modes ahead of time or resorting to regularization with ad hoc penalties[20,24,29,31] not motivated directly by the clustering objective or approximative information criteria[19,28,30] poorly adapted to network problems. Overall, current approaches for clustering network populations do not provide a principled solution for model selection and often demand extensive tuning and significant computational overhead from fitting the model to several choices of the number of clusters.

Here we introduce nonparametric inference methods which overcome these obstacles and provide a coherent framework through which to approach the problem of clustering network populations or multiplex network layers while extracting a representative modal network to summarize each cluster. Our solution employs the minimum description length principle, which allows us to derive an objective function that favors parsimonious representations in an information-theoretic sense and selects the number and composition of representative modal networks automatically from first principles. We first develop a fast Monte Carlo scheme for identifying the configuration of measurement clusters and modal networks that minimizes our description length objective. We then extend our framework to account for special cases of interest: bipartite/directed networks and contiguous clusters containing all ordered networks from the earliest to the latest. We show how to solve the latter problem in polynomial run time with a dynamic program[32]. We demonstrate our methods in applications involving synthetic and real network data, finding that they can effectively recover planted network modes and clusters even with considerable noise. Our methods also provide a concise and meaningful summary of real network populations from applications in global trade and macroevolutionary research.

Results

We test our methods on a range of real and synthetic example network populations. First, we show that our algorithms can recover synthetically generated clusters and modes with high accuracy despite considerable noise levels. Applied to worldwide networks of food imports and exports, we find a strong compression that uses the difference between categories of products and the locations in which they are produced. We then apply our method for contiguous clustering of ordered network populations to a set of networks representing the fossil record from ordered geological stages in the last 500 million years[33]. We examine bipartite and unipartite representations of these systems and find close alignment between our inferred clusters and known global biotic transitions, including those triggered by mass extinction events.

Reconstruction of synthetic network populations. To demonstrate that our algorithms (presented in "Methods") can effectively identify modes and clusters in network populations, we test their ability to recover the underlying modes and clusters generated from the heterogeneous population model introduced in ref. [20]. We examine the robustness of these methods under varying noise levels that influence the similarity of the generated networks with the cluster's mode.

The generative model in ref. [20] supposes (using different notation) that we are given K modes $\mathcal{A}$ as well as the cluster assignments $\mathcal{C}$ of the networks $\mathcal{D}$. Each network $s \in C_k$ is generated by first taking each edge $(i,j) \in \boldsymbol{A}^{(k)}$ independently and adding it to $\boldsymbol{D}^{(s)}$ with probability α_k (the *true-positive rate*). Then, each of M_k^* possible edges absent from $\boldsymbol{A}^{(k)}$ is added to $\boldsymbol{D}^{(s)}$ with probability β_k (the *false-positive rate*). After performing this procedure for all clusters, the end result is a heterogeneous population of networks $\mathcal{D}$ with K underlying modes, with noise in the networks C_k surrounding each mode $\boldsymbol{A}^{(k)}$ determined by the rates α_k and β_k. The higher the true-positive rate α_k and the lower the false-positive rate β_k, the closer the networks in cluster C_k resemble their corresponding mode $\boldsymbol{A}^{(k)}$.

Employing Bayesian inference of the modes and cluster assignments as in ref. [20] involves adding prior probability distributions over the modes $\mathcal{A}$ and cluster assignments $\mathcal{C}$ to the heterogeneous network model[20]. With a specific choice of priors on the modes and cluster sizes, Eq. (15) is precisely the equation giving us the Maximum A Posteriori (MAP) estimators

of $\mathcal{A}$ and $\mathcal{C}$ in this model. We defer the details of this correspondence to Supplementary Note 1.

For our experiments, we use two modes, mode 1 and mode 3 from the diagram in "Methods", as the planted modes $\mathcal{A}_{true}$ we aim to recover. To provide a single intuitive parameter quantifying the noise level in the generative model, we choose the true- and false-positive rates to satisfy $p = \beta_1 = \beta_3 = 1 - \alpha_1 = 1 - \alpha_3$ for each run. Viewing the networks as binary adjacency matrices, the parameter p corresponds to the probability of flipping entries of the matrix from 0 to 1 and vice-versa when constructing a network from its assigned cluster. We denote the parameter p as the "flip probability" to emphasize this interpretation (same formulation as in ref. [20]). A flip probability $p = 0$ corresponds to clusters of networks identical to the cluster modes, and a flip probability of $p = 0.5$ corresponds to completely random networks with no clustering in the population. We thus expect it to be easy to recover the planted modes $\mathcal{A}_{true}$ and clusters $\mathcal{C}_{true}$ for $p = 0$, and the problem becomes more and more difficult as we approach $p = 0.5$.

We run three separate recovery experiments to test both the unconstrained and contiguous description length objectives in Eq. (15) and Eq. (19), respectively. For the unconstrained objective, in each run, we generate a population of S networks from the model described above, with each network generated from either mode 1 or mode 3 at random with equal probability. We then identify the modes $\mathcal{A}_{\mathrm{MDL}}$ and clusters $\mathcal{C}_{\mathrm{MDL}}$ that minimize the objective in Eq. (15) using the merge-split algorithm detailed in "Methods" and Supplementary Note 2. For the recovery of contiguous clusters, in one experiment we generate $S/2$ consecutive networks from each mode so that the population consists of $K = 2$ adjacent contiguous clusters. And in another experiment, we generate $S/4$ networks from mode 1, $S/4$ networks from mode 3, and repeat this so that there are $K = 4$ adjacent contiguous clusters of the S networks generated from the two distinct modes. For these two experiments, we run the dynamic programming algorithm detailed in "Methods" to identify the modes $\mathcal{A}_{\mathrm{MDL}}$ and clusters $\mathcal{C}_{\mathrm{MDL}}$ that minimize the objective in Eq. (19). In all three experiments, we generate a population of $S = 100$ networks, each constructed from its corresponding mode using the single flip probability p to introduce true- and false-positive edges.

To quantify the mode recovery error, we use the network distance quantified by the average Hamming distance between the inferred modes $\mathcal{A}$ and the planted modes $\mathcal{A}_{true}$. As both of our algorithms automatically select the optimal number of clusters K, the number of modes we infer can differ from the true number ($K = 2$ or $K = 4$, depending on the experiment). In each experiment, we therefore choose the K inferred modes in $\mathcal{A}$ with the largest corresponding clusters and compute the average Hamming distance between these and the true modes in $\mathcal{A}_{true}$. (Since there are $K!$ ways to choose the inferred mode labels, we choose the labeling that produces the smallest Hamming distance.) To measure the error between our inferred clusters $\mathcal{C}$ and the planted clusters $\mathcal{C}_{true}$ (the "partition distance"), we use one minus the normalized mutual information[34]. We also compute the inverse compression ratio (Eq. (17)) to measure how well the network population can be compressed. We pick a range of values of p to tune the noise level in the populations, and at each value of p we average these three quantities over 200 realizations of the model to smooth out noise due to randomness in the synthetic network populations. We choose $K_0 = 1$ for these experiments, but this choice has little to no effect on the results (see Supplementary Note 3).

Figure 1 shows the results of our first reconstruction experiment. The reconstruction performance gradually worsens as p increases due to the increasing noise level in the sampled networks relative to their corresponding modes (Fig. 1a). In all experiments, the network distance reaches that expected for a completely random guess of the mode networks—a 50/50 coin flip to determine the existence of each edge, denoted by the dashed black line—when $p = 0.5$. The results in Fig. 1a indicate that in both the unconstrained and contiguous cases, our algorithms are capable of recovering the modes underlying these synthetic network populations with high accuracy, even for substantial levels of noise (up to $p \approx 0.3$, corresponding to an average of 30% of the edges/non-edges differing between each mode and networks in its cluster).

The partition distance shows similar gradual performance degradation, with substantial increases in the distance beginning at $p \approx 0.3$ for the contiguous experiments and $p \approx 0.15$ for the

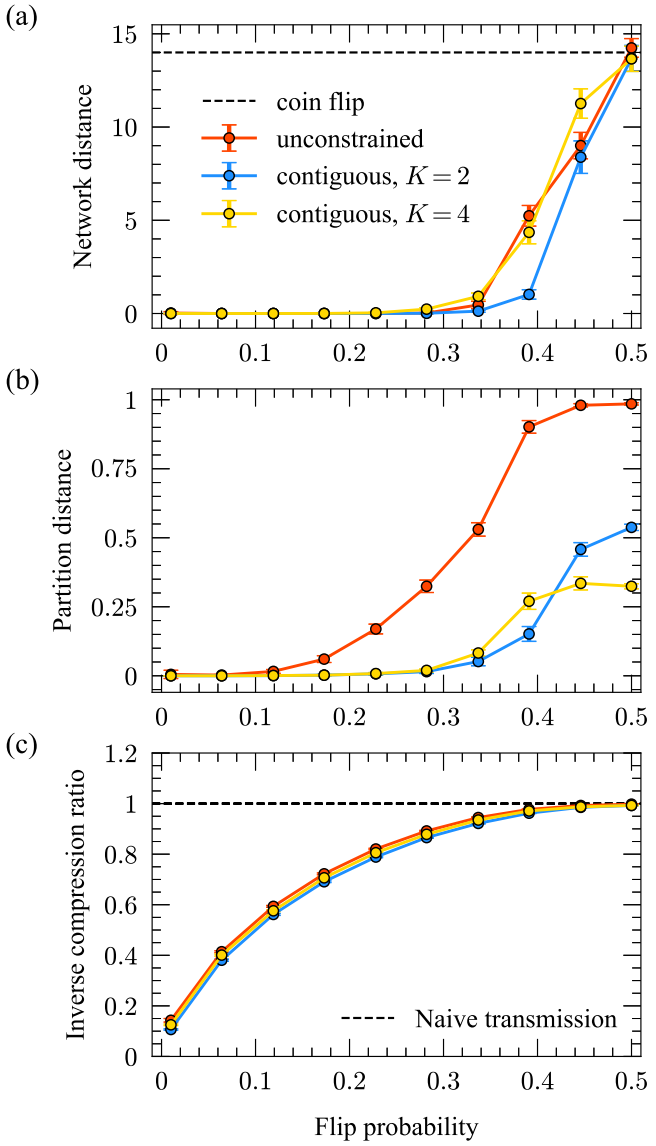

Fig. 1 Recovery of planted modes and their clusters in synthetic network populations. Various aspects of the recovery performance are plotted for the three experiments described in "Results". **a** Network distance, as quantified by the average Hamming distance between the true and inferred and modes (example modes 1 and 3 in "Methods"), for various flip probabilities p. **b** Partition distance, given by the one minus the normalized mutual information between the true and inferred clusterings of the network population. **c** Inverse compression ratio, given in Eq. (17). Each data point is an average over 200 realizations of the population for the corresponding value of the flip probability, and error bars correspond to three standard errors in the mean.

unconstrained experiment (Fig. 1b). The partition distance levels off at different values across the three experiments, with the unconstrained case exhibiting significantly worse performance than the contiguous cases. We expect this result since contiguity simplifies the reconstruction problem by reducing the space of possible clusterings. Because information-based measures account for the entire space of possible clusterings instead of the highly constrained set produced by contiguous partitions, they overestimate the similarity of partitions in this constrained set. This overestimation intensifies with more clusters[35].

The inverse compression ratio (Eq. (17)) for these experiments gradually approaches 1 (no compression relative to transmitting each network individually, denoted by the dashed black line) as the noise level p increases (Fig. 1c). This result is consistent with the intuition that noisier data will be harder to compress, while data with strong internal regularity will be much easier to compress, as the homogeneities can be exploited for shorter encodings. When p is small, we can achieve up to 10 times compression over the naive baseline by using the inferred underlying modes and clusters to transmit these network populations.

The results in Fig. 1 indicate that our algorithms can recover the underlying modes and their clusters in synthetic network populations. However, these results also depend on how distinguishable the underlying modes are. For identical modes, $\boldsymbol{A}^{(1)} = \boldsymbol{A}^{(2)}$, it is impossible to recover the cluster labels of the individual network samples $\boldsymbol{D}^{(s)}$. To investigate the dependence of the recovery performance on the modes themselves, we repeat the experiment in Fig. 1, except this time we systematically vary the mode networks $\mathcal{A}$ for each trial to achieve various levels of distinguishability. In each trial, we set $\boldsymbol{A}^{(1)}$ equal to mode 1 from "Methods" (as before), but then generate the edges in $\boldsymbol{A}^{(2)}$ from $\boldsymbol{A}^{(1)}$ using the flip probability γ, which we call the "mode separation". For mode separations $\gamma \approx 0$, it is challenging to recover the correct cluster labels of the individual sample networks because $\boldsymbol{A}^{(2)}$ will closely resemble $\boldsymbol{A}^{(1)}$. On the other hand, for mode separations $\gamma \approx 0.5$, the modes will typically be easily distinguished since $\boldsymbol{A}^{(2)}$ will have many edges/non-edges that have flipped relative to $\boldsymbol{A}^{(1)}$.

Figure 2 shows the results of this second experiment. The panels show the partition distance between the true and inferred cluster labels for a range of mode separations γ. In all experiments, the recovery becomes worse for lower values of the separation γ, but the algorithm still recovers a significant amount of cluster information even for relatively low γ. As in the previous set of experiments, the recovery performance is substantially worse for the discontiguous case compared with the contiguous cases, again due to the highly constrained ensemble of possible partitions considered by the partition distance in the contiguous cases.

In Supplementary Note 3, we show the recovery performance results for the network distance between the true and inferred modes as we vary the mode separation γ. For the mode recovery, the results are even more robust to the changes in mode separation. This result is consistent with the recovery performance in Fig. 1, where the recovery performance of the partitions starts to worsen at lower noise levels than the recovery of the modes. Thus, small perturbations in the inferred clusters may not affect the inferred modes much, since misclassified networks likely have little in common with the rest of their cluster.

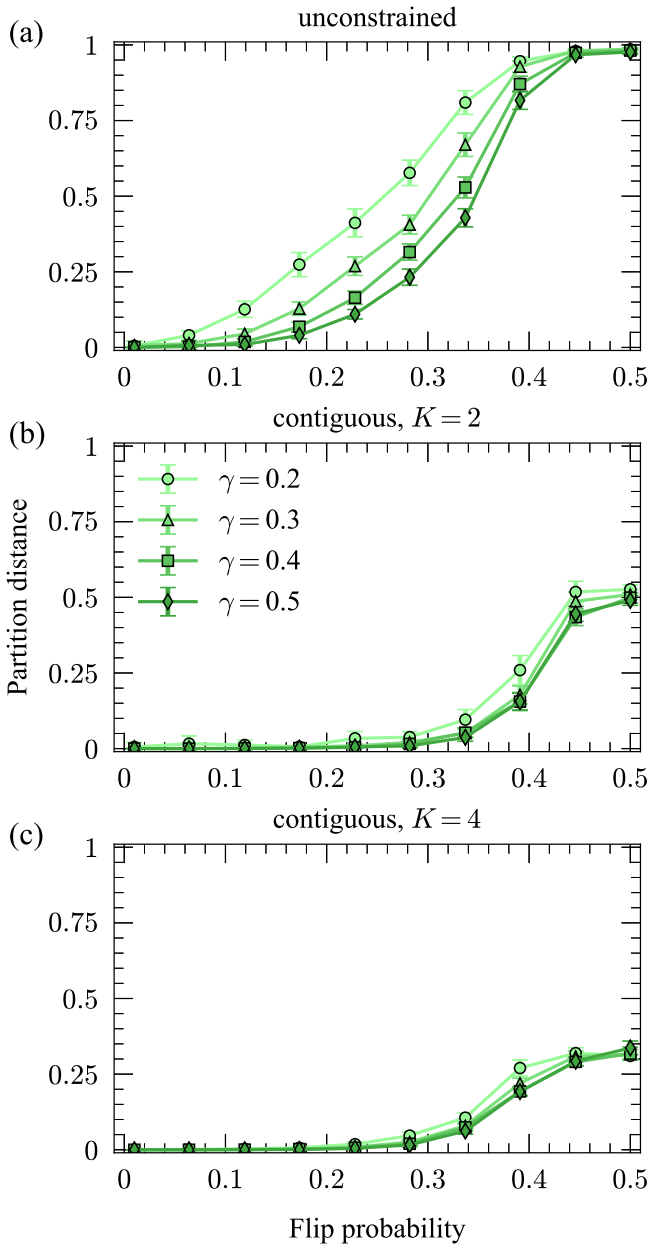

Fig. 2 Cluster recovery for different mode separations. Partition distance between true and inferred clusters for **a** unconstrained clustering, **b** contiguous clustering with $K = 2$, and **c** contiguous clustering with $K = 4$ for various values of the mode separation γ. Each data point is an average over 200 realizations of the population for the corresponding value of the flip probability, and error bars correspond to three standard errors in the mean.

Unordered network population representing global trade relationships. For our first example with empirical network data, we study a collection of worldwide import/export networks. The nodes represent countries and the edges encode trading relationships. The Food and Agriculture Organization of the United Nations (FAO) aggregates these data, and we use the trades made in 2010, as in ref. [24]. Each network in the collection corresponds to a category of products, for example, bread, meat, or cigars. We ignore information about the intensity of trades and merely record the presence or absence of a trading relationship for each category of products. The resulting collection comprises 364 networks (layers) on the same set of 214 nodes with 874.6 edges (average degree of 8.2) on average, with some sparse networks having as little as one edge and the densest containing 6529 edges. These networks are unordered, so we employ the discontiguous clustering method described in "Methods". We run the algorithm multiple times with a varying initial number of clusters K_0 to find the best optima, although as with the synthetic reconstruction examples the choice of K_0 has little impact on compression. The

best compression we find results in eight modes and achieves a compression ratio of $\eta(\mathcal{D}) = 0.562$, indicating that it is nearly twice as efficient to communicate the data when we use the modal networks and their clusters. In contrast, in ref. [24] a clustering analysis of the same network layers using structural reducibility—a measure of how many layers can be aggregated to reduce pairwise information redundancies among the layers—yielded 182 final aggregated layers, which would poorly compress the data under our scheme and not provide a significant benefit in downstream analyses due to the large final number of clusters. Key properties of the configuration of modes and clusters inferred by our algorithm are illustrated in Fig. 3.

In Fig. 3a we show the number of edges in each inferred mode, which indicates that these modes vary substantially in density to reflect the key underlying structures in networks within their corresponding clusters. The sizes of the clusters, shown in Fig. 3b, also vary substantially, with the most populated cluster (cluster 4) containing nearly 7 times as many networks as the least populated cluster (cluster 6). Some striking geographical commonalities and differences in the structure of the modes can be seen due to the varying composition of their corresponding clusters of networks. Figure 3c, d shows the differences and similarities respectively between the structure of the modes for clusters 5 and 7, which are chosen as example modes because of their modest densities and distinct distributions of product types (Fig. 3e). Edges that are in mode 7 but not in mode 5 are highlighted in blue, while edges in mode 5 but not in mode 7 are highlighted in red. Meanwhile, the shared edges common to both networks are shown in Fig. 3d in black. Mode 5, which contains a diversity of product types and a relatively large portion of grain and protein products, has a large number of edges connecting the Americas to Europe that are not present in mode 7. On the other hand, mode 7, which is primarily composed of networks representing the trade of fruits and vegetables, has many edges in the global south that are not present in mode 5. However, both modes share a common backbone of edges that are distributed globally.

We categorized the 364 products (the network layers being clustered) into 12 broader categories of product types, plotting their distributions within each cluster in Fig. 3e. There are a few interesting observations we can make about this figure. Nearly all of the dairy products are traded within networks in a single cluster (cluster 3), indicating a high degree of similarity in the trade patterns for dairy products across countries. A similar observation can be made for live animals, which are primarily traded in cluster 4. On the other hand, many of the other products (grains, proteins, sweets, fruits, vegetables, and drinks) are traded in reasonable proportion in all clusters, which may reflect the diversity of these products as well as their geographical sources, which can give rise to heterogeneous trading structures. The densities of the modes and the sizes of the clusters do not have a clear relationship, with cluster 6 containing the smallest number of networks but the densest mode, and clusters 4 and 5 having sparser modes and much larger clusters. This reflects a higher level of heterogeneity in the structure of the trading relationships captured in cluster 6, which requires a denser mode for optimal compression, while the converse is true for clusters 4 and 5.

We also identify substantial structural differences in the inferred modes. In Supplementary Note 4, we compute summary statistics (average degree, transitivity, and average betweenness) for the modes output in this experiment and the network layers in their corresponding clusters. The statistics vary much more across clusters than within the clusters, suggesting that the MDL optimal mode configuration exemplifies distinct network structures within the dataset. Because the within-cluster average value of each statistic and the corresponding value for the mode network are similar, our method provides an effective preprocessing step for network-level regression tasks.

Ordered network population representing the fossil record. We conclude our analysis with a study of a set of networks representing global marine fauna over the past 500 million years. We aggregate fossil occurrences of the shelled marine animals, including bryozoans, corals, brachiopods, mollusks, arthropods, and echinoderms, into a regular grid covering the Earth's surface[33]. From these data, we construct unweighted bipartite networks representing 90 ordered time intervals in Earth's history (geological stages): An edge between a genus and a grid cell indicates that the genus was observed in the grid cell during the network's corresponding geological stage. We also construct the unipartite projections of these networks: An edge from one genus to another indicates that these two genera were present in the same grid cell during the stage corresponding to the network. In total, there were 18,297 genus nodes, 664 grid cell nodes, 67,371.5 edges on average for the 90 unipartite graphs (average degree of 7.4), and 1462.2 edges on average for the bipartite graphs (average degree of 0.08, corresponding to an average of roughly 10 percent of genera being present at each layer).

In Fig. 4, we show the results of applying our clustering method for contiguous network populations (see "Methods") to both the unipartite and bipartite populations representing the post-Cambrian fossil record. We find clusters that capture the known large-scale organization of marine diversity. Major groups of marine animals archived in the fossil record are organized into global-scale assemblages that sequentially dominated oceans and shifted across major biotic transitions. Overall, the bipartite and unipartite fossil record network representations both result in transitions concurrent with the major known geological perturbations in Earth's history, including the so-called mass extinction events. However, differences in the clusters retrieved from the unipartite and bipartite representations of the underlying paleontological data highlight the impact of this choice on the observed macroevolutionary pattern[36].

We also use our methodology to assess the extent to which the standard division of the post-Cambrian rock record in the geological time scale and the well-known mass extinction events compress the assembled networks. Specifically, we evaluate the inverse compression ratio in Eq. (17) on three different partitions of the fossil record networks that are defined by clustering the assembled networks into geological eras (Paleozoic, Mesozoic, and Cenozoic), geological periods (Ordovician to Quaternary), and six time intervals between the five mass extinctions in Fig. 4, with planted modes constructed by placing the networks into each cluster and applying the greedy algorithm described in "Methods" and Supplementary Note 2.

Table 1 shows the results of these experiments. All three partitions compress the fossil record networks almost as much as the optimal partition, which represents a natural division based on major regularities. Accordingly, the planted partition based on mass extinctions is almost as good as this optimal partition because mass extinctions are concurrent with the major geological events shaping the history of marine life. In contrast, partitions based either on standard geological eras or periods are less optimal, likely because they represent, to some extent, arbitrary divisions that are maintained for historical reasons. Our results here provide a complementary perspective to the work in ref. [33], where a multilayer network clustering algorithm was employed that clusters nodes within and across layers to reveal three major biotic transitions from the fossil data. In Supplementary Note 5 we review this and other existing multiplex and network population-clustering techniques, discussing the similarities and differences with our proposed methods.

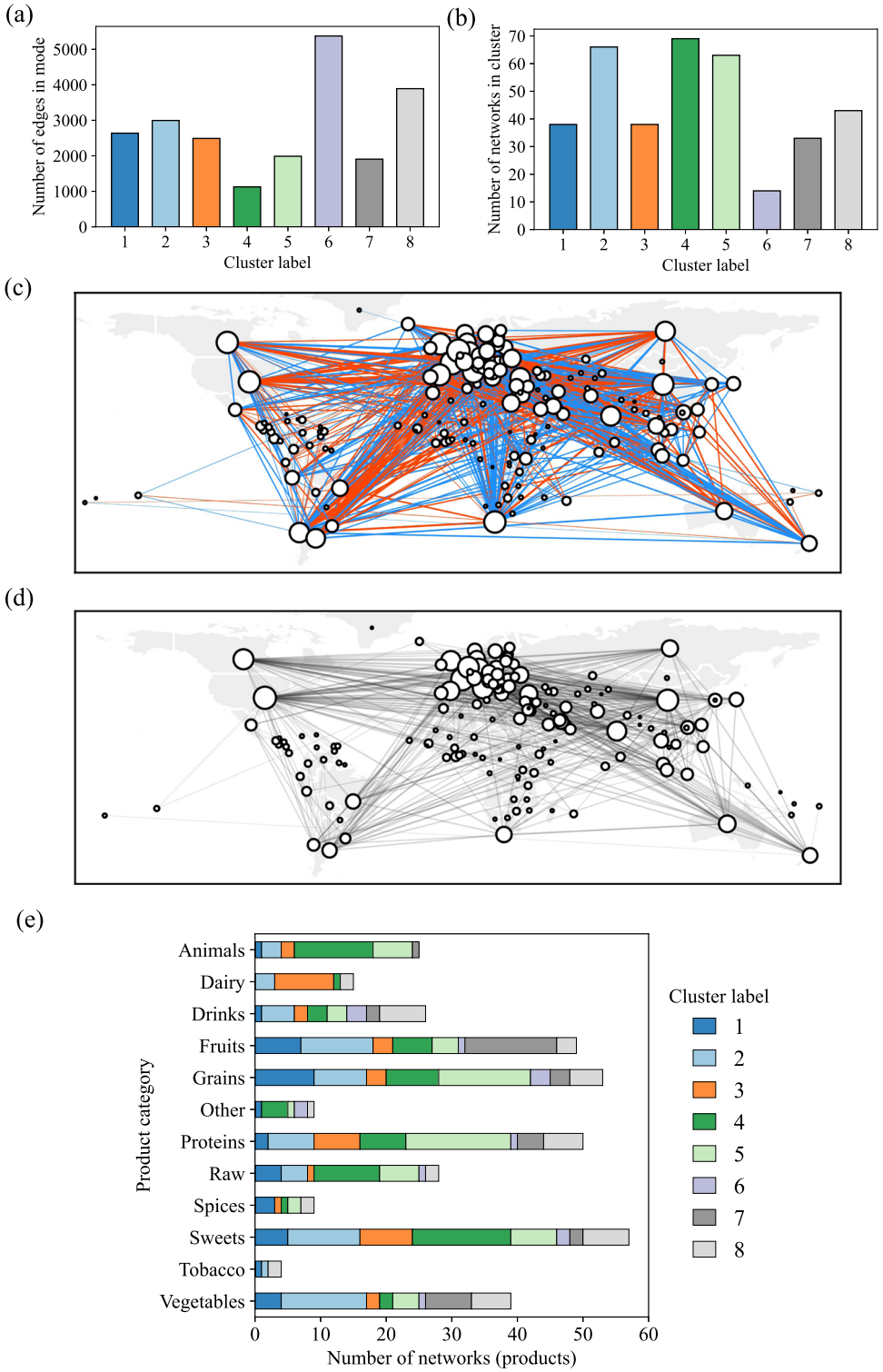

Fig. 3 Discontiguous networks of imports and exports. We apply our algorithm for clustering discontiguous populations (see "Methods") to a collection of trade networks[24] described in "Results" to identify similar networks of products. **a** Number of edges in each cluster's mode. **b** Number of networks in each cluster. **c** Edges in mode 7 but not mode 5 are colored in blue, while edges in mode 5 but not in mode 7 are colored in red, highlighting the differences between these two modes. **d** The shared backbone of edges common to both modes 5 and 7. **e** Distribution of product types across the networks in each cluster.

Conclusion

We have used the minimum description length principle to develop efficient parameter-free methods for summarizing populations of networks using a small set of representative modal networks that succinctly describe the variation across the population. For clustering network populations with no ordering, we have developed a fast merge-split Monte Carlo procedure that performs a series of moves to refine a partition of the networks. For clustering ordered networks into contiguous clusters, we employ a time and memory-efficient dynamic programming

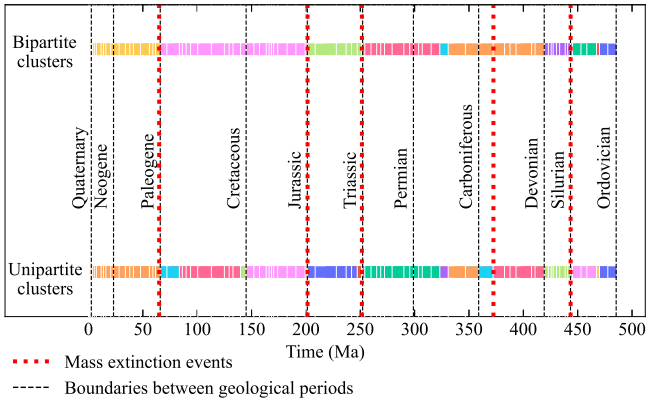

Fig. 4 Contiguous clusters of network representing the post-Cambrian fossil record. We apply the dynamic programming algorithm of "Methods" to the unipartite genus-genus network population (lower bar) and the bipartite genus-location network population (upper bar) described in "Results" to identify key time intervals with distinct fossil assemblages. The clusters inferred by the algorithm are represented with distinct colors, and the networks, one per each post-Cambrian geological stage, are separated by white lines. Boundaries between geological periods, i.e., larger scale rock units[55], are indicated by dashed vertical black lines. The five major mass extinction events[56] are shown in dotted vertical red lines.

Table 1 Compression results for different partitions of the fossil record.

	Eras	Extinction events	Periods	MDL optimal
$K(\mathcal{D}_{uni})$	3	6	11	17
$K(\mathcal{D}_{bi})$	3	6	11	10
$\eta(\mathcal{D}_{uni})$	0.814	0.805	0.810	0.796
$\eta(\mathcal{D}_{bi})$	0.842	0.838	0.850	0.833

$\mathcal{D}_{uni}$ denotes the set of fossil record networks in the unipartite genus-genus representation, and $\mathcal{D}_{bi}$ denotes the same networks in the bipartite genus-location representation.

approach. These algorithms can accurately reconstruct modes and associated clusters in synthetic datasets and identify significant heterogeneities in real network datasets derived from trading relationships and fossil records. Our methods are principled, nonparametric, and efficient in summarizing complex sets of independent network measurements, providing an essential tool for exploratory and visual analyses of network data and preprocessing large sets of network measurements for downstream applications.

This information-theoretic framework for representing network populations with modal networks can be extended in several ways. For example, a multi-step encoding that allows for hierarchical partitions of network populations would capture multiple levels of heterogeneity in the data. More complex encodings that exploit structural regularities within the networks would allow for simultaneous inference of mesoscale structures—such as communities, core-periphery divisions, or specific informative subgraphs[37]—along with the modes and clusters. The encodings can also be adapted to capture weighted networks with multi-edges by altering the combinatorial expressions for the number of allowable edge configurations.

Methods

Minimum description length objective. For our clustering method, we rely on the minimum description length (MDL) principle: the best model among a set of candidate models is the one that permits the greatest compression—or shortest description—of a dataset[38]. The MDL principle provides a principled criterion for statistical model selection and has consequently been employed in various applications ranging from regression to time series analysis to clustering[39]. A large body of research uses the MDL principle for clustering data, including studies on MDL-based methods for mixture models that accommodate continuous[40,41] and categorical data[42], as well as methods that are based on more general probabilistic generative models[43]. The MDL approach has also been applied to complex network data, most notably for community detection algorithms to cluster nodes within a network[35,44,45] and for decomposing graphs into subgraphs[46–50], but also for clustering entire partitions of networks[51]. Our methods are similar in spirit to the one presented in ref. [51] for identifying representative community divisions among a set of plausible network partitions. Both approaches involve transmitting first a set of representatives and then the dataset itself by describing how each partition or network differs from its corresponding representative. However, the methods differ substantially in their details since they address fundamentally different questions.

We consider an experiment in which the initial data are a population of networks consisting of S undirected, unweighted networks $\mathcal{D} = \{\boldsymbol{D}^{(1)}, \ldots, \boldsymbol{D}^{(S)}\}$ on a set of N labeled nodes. The networks record, for instance, the co-location patterns among kids in a class of N students over S class periods.

We aim to summarize these data with K modal networks $\mathcal{A} = \{\boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(K)}\}$ (also undirected and unweighted) on the same set of nodes, with associated clusters of networks $\mathcal{C} = \{C_1, \ldots, C_K\}$, where C_k comprises networks that are similar to the mode $\boldsymbol{A}^{(k)}$. This summary would allow researchers to, for instance, perform all downstream network analyses on a small set of representative networks—the modes—instead of a large set of networks likely to include measurement errors and from which it is difficult to draw valid conclusions.

We assume for simplicity of presentation that all networks $\mathcal{D}$ and $\mathcal{A}$ have no self- or multi-edges, although we can account for them straightforwardly. While K can be fixed if desired, we assume that it is unknown and must be determined from regularities in the data.

To select among all the possible modes and assignments of networks to clusters, we employ information theory and construct an objective function that quantifies how much information is needed to communicate the structure of the network population $\mathcal{D}$ to a receiver. Clustering networks in groups of mostly similar instances allows us to communicate the population $\mathcal{D}$ efficiently in three steps: first the modes, then the clusters, and finally the networks $\mathcal{D}$ themselves as a series of small modifications to the modes $\mathcal{A}$. The MDL principle tells us that any compression achieved in this way reveals modes and clusters that are genuine regularities of the population rather than noise[38].

We first establish a baseline for the code length: the number of bits needed to communicate $\mathcal{D}$ without using any regularities. One way to do this is to first communicate the parameters of the population at a negligible information cost (size S, number of nodes N, and the total number of edges E in all networks of $\mathcal{D}$) and then transmit the population $\mathcal{D}$ directly. There are $\binom{N}{2}$ possible edge positions in each of the S undirected networks in $\mathcal{D}$, or $S\binom{N}{2}$ possible edge positions for the whole population. So these networks can be configured in $\binom{S\binom{N}{2}}{E}$ ways. It thus takes approximately

$$\mathcal{L}_0(\mathcal{D}) = \log\binom{S\binom{N}{2}}{E} \tag{1}$$

bits to transmit these networks to a receiver. (We use the convention $\log \equiv \log_2$ for brevity.) Applying Stirling's approximation $\log x! \approx x\log x - x/\ln(2)$, we obtain

$$\mathcal{L}_0(\mathcal{D}) \approx S\binom{N}{2} H_b\left(\frac{E}{S\binom{N}{2}}\right) \tag{2}$$

written in terms of the binary Shannon entropy

$$H_b(p) = -p\log p - (1-p)\log(1-p). \tag{3}$$

In practice, we expect to need many fewer bits than $\mathcal{L}_0$ to communicate $\mathcal{D}$, because the population of networks will often have regularities. We propose a multi-part encoding that identifies such regularities by grouping similar networks in clusters $\mathcal{C}$ with modes $\mathcal{A}$, which proceeds as follows. First, we send a small number of modes $\mathcal{A}$ in their entirety, which ideally captures most of the heterogeneity in the population $\mathcal{D}$. This step is costly but will save us information later. We then send the network clusters $\mathcal{C}$ by transmitting the cluster label of each network $s \in \mathcal{D}$. Finally, we transmit the edges of networks in each cluster, using the already transmitted modes as a starting point to compress this part of the encoding significantly. The expected code length can be quantified using simple combinatorial expressions, and the configuration of modes $\mathcal{A}$ and clusters $\mathcal{C}$ that minimizes the total expected code length—the MDL configuration—provides a succinct summary of the data $\mathcal{D}$. Figure 5 summarizes the transmission process and the individual description length contributions.

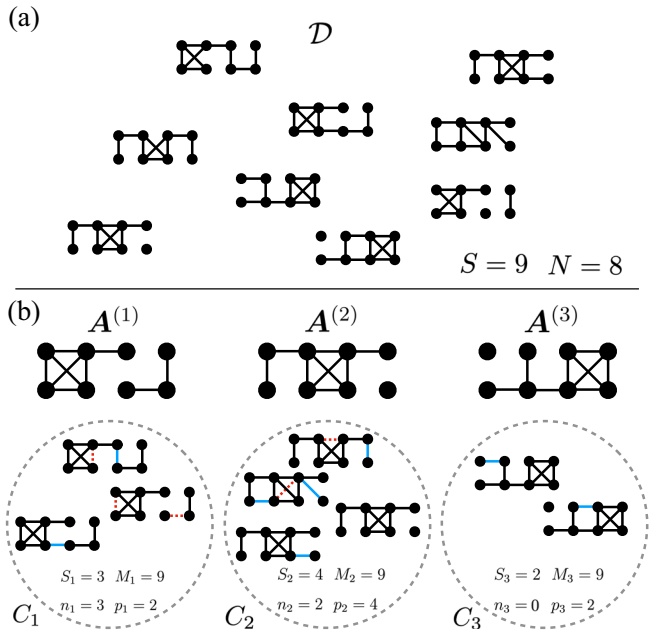

Fig. 5 Information transmission scheme. a Example population of networks $\mathcal{D}$, with $S = 9$ networks of $N = 8$ nodes each. **b** Representative modes $\{\boldsymbol{A}^{(k)}\}$ with their corresponding clusters of networks $\{C_k\}$. First, each mode network is transmitted individually in its entirety, with information content $\mathcal{L}(\boldsymbol{A}^{(k)})$ given by Eq. (4). Then, networks in the population are assigned to disjoint clusters surrounding each mode, requiring information content given by Eq. (6). Finally, all the networks $\boldsymbol{D}^{(s)}$ in each cluster C_k are transmitted, given the number of false-negative and false-positive edges n_k and p_k in the cluster (represented with dotted red and solid blue lines, respectively). The information content of this step is given by ℓ_k in Eq. (10). Different choices of clusters and modes lead to different total information content, and the aim is to identify the clusters and modes that minimize this information content.

The expected length of this multi-part encoding is the sum of the description length of each part of the code that has significant communication costs. The modes are the first objects that incur such costs. Following the same reasoning as before, we denote the number of edges in mode k as M_k and conclude that we can transmit the positions of the occupied edges in mode $\boldsymbol{A}^{(k)}$ using approximately

$$\mathcal{L}(\boldsymbol{A}^{(k)}) = \log\binom{\binom{N}{2}}{M_k} \approx \binom{N}{2} H_b\left(\frac{M_k}{\binom{N}{2}}\right) \tag{4}$$

bits, where the second expression results from a Stirling approximation as in Eq. (2). We can therefore transmit all the modes with a total code length of

$$\mathcal{L}(\mathcal{A}) = \sum_{k=1}^{K} \mathcal{L}(\boldsymbol{A}^{(k)}) \tag{5}$$

bits.

The next step is to transmit the cluster label k of each network in $\mathcal{D}$. For this part of the code, we first send the number of networks S_k in each cluster $k = 1, \ldots, K$ at a negligible cost and then specify a particular clustering compatible with these constraints. The multinomial coefficient $\binom{S}{S_1\, S_2\, \cdots\, S_k}$ gives the total number of possible combinations of these cluster labels. The information content of this step is thus

$$\mathcal{L}(\mathcal{C}) = \log\binom{S}{S_1\, S_2\, \cdots\, S_k} \approx S H\big(\{S_k/S\}\big), \tag{6}$$

where we again use the Stirling approximation and where

$$H(\{q_k\}) = -\sum_{k=1}^{K} q_k \log q_k \tag{7}$$

is the Shannon entropy of a distribution $\{q_k\}$.

Finally, we transmit the network population $\mathcal{D}$ by sending the *differences* between the networks in each cluster and their associated mode. To calculate the length of this part of the code, we focus on a particular cluster C_k and count the number of times we will have to remove an edge from the mode $\boldsymbol{A}^{(k)}$ when specifying the structure of networks in its cluster using $\boldsymbol{A}^{(k)}$ as a reference. We call these edges *false negatives* and count them as

$$n_k = \sum_{s \in C_k} \big|\boldsymbol{A}^{(k)} \setminus \boldsymbol{D}^{(s)}\big|, \tag{8}$$

where we interpret $\boldsymbol{D}^{(s)}$ and $\boldsymbol{A}^{(k)}$ as sets of edges, so the summand is the number of edges in mode k that are not in the network s. Similarly, we also require the number of edges that occur in the networks of cluster k but not in the mode—the number of *false positives*,

$$p_k = \sum_{s \in C_k} \big|\boldsymbol{D}^{(s)} \setminus \boldsymbol{A}^{(k)}\big|. \tag{9}$$

Like the cluster sizes S_k and edge counts per cluster M_k, the pairs (n_k, p_k) can be communicated to the receiver at a comparatively negligible cost, and we ignore them in our calculations.

To estimate the information content of this part of the transmission, we count the number of configurations of false-negative and false-positive edges in C_k. Focusing first on the false negatives—the edges that must be deleted—we count that of the $S_k M_k$ edges in the S_k copies of the mode of cluster k, n_k will be false-negative edges that can be configured in $\binom{S_k M_k}{n_k}$ ways. Similarly, using the shorthand $M_k^* = \binom{N}{2} - M_k$ to denote the unoccupied pairs of nodes in the mode k, there are $S_k M_k^*$ locations in which we must place p_k false-positive edges, for a total of $\binom{S_k M_k^*}{p_k}$ possible configurations of false-positive edges. The total information content required for transmitting the locations of the false-negative and false-positive edges of every network in cluster k is thus

$$\begin{aligned} \ell_k &:= \mathcal{L}(\{\mathcal{D}^{(s)} | s \in C_k\} | \boldsymbol{A}^{(k)}) \\ &= \log\binom{S_k M_k}{n_k} + \log\binom{S_k M_k^*}{p_k}, \end{aligned} \tag{10}$$

which we approximate as

$$\ell_k \approx S_k M_k H_b\left(\frac{n_k}{S_k M_k}\right) + S_k M_k^* H_b\left(\frac{p_k}{S_k M_k^*}\right). \tag{11}$$

Summing over all clusters,

$$\mathcal{L}(\mathcal{D} | \mathcal{A}, \mathcal{C}) = \sum_{k=1}^{K} \ell_k, \tag{12}$$

we obtain the total information content of the final step in the transmission process.

We obtain the total description length $\mathcal{L}(\mathcal{D})$ by adding the contributions of Eqs. (5), (6), and (12), as

$$\mathcal{L}(\mathcal{D}) = \mathcal{L}(\mathcal{A}) + \mathcal{L}(\mathcal{C}) + \mathcal{L}(\mathcal{D} | \mathcal{A}, \mathcal{C}). \tag{13}$$

This objective function allows for efficient optimization because we can express it as a sum of the cluster-level description lengths

$$\mathcal{L}_k(\boldsymbol{A}^{(k)}, C_k) = \mathcal{L}(\mathcal{A}^{(k)}) + S \log\left(\frac{S}{S_k}\right) + \ell_k, \tag{14}$$

giving

$$\mathcal{L}(\mathcal{D}) = \sum_{k=1}^{K} \mathcal{L}_k(\boldsymbol{A}^{(k)}, C_k). \tag{15}$$

Equations (4) and (10) provide explicit expressions for $\mathcal{L}(\mathcal{A}^{(k)})$ and ℓ_k.

Equation (15) gives the total description length of the data $\mathcal{D}$ under our multi-part transmission scheme. By minimizing this objective function we identify the best configurations of modes $\mathcal{A}$ and clusters $\mathcal{C}$. A good configuration $\{\mathcal{A}, \mathcal{C}\}$ will allow us to transmit a large portion of the information in $\mathcal{D}$ through the modes alone. If we use too many modes, the description length will increase as these are costly to communicate in full. And if we use too few, the description length will also increase because we will have to send lengthy messages describing how mismatched networks and modes differ. Hence, through the principle of parsimony, Eq. (15) favors descriptions with the number of clusters K as small as possible but not smaller.

This framework can be modified to accommodate populations of bipartite or directed networks. For the bipartite case, we make the transformations $\binom{N}{2} \to N_1 N_2$ and $M_k^* \to N_1 N_2 - M_k$, where N_1 and N_2 are the numbers of nodes in each of the two groups. This modification reduces the number of available positions for potential edges. Similarly, for the directed case, we can make the transformations $\binom{N}{2} \to N(N-1)$ and $M_k^* \to N(N-1) - M_k$, which increases the number of available edge positions.

Optimization and model selection. Since Eq. (15) has large support, is not convex, and has many local optima, a stochastic optimization method is a natural choice for finding reasonable solutions rapidly. We exploit the objective function's decoupling into a sum over clusters k and implement an efficient merge-split Monte Carlo method for the search[51,52]. The method greedily optimizes $\mathcal{L}(\mathcal{D})$ using moves that involve merging and splitting clusters of networks $D^{(s)} \in \mathcal{D}$.

Our merge-split algorithm minimizes the description length in Eq. (15) by performing one of the following moves selected uniformly at random and accepting the move as long as it results in a reduction of the description length (15):

1. **Reassignment**: Pick a network s at random and move it from its current cluster C_k to the cluster $C_{k'}$ that results in the greatest decrease in the description length. Compute the modes $\boldsymbol{A}^{(k)}$ and $\boldsymbol{A}^{(k')}$ that minimize the cluster-level description lengths $\mathcal{L}_k(\boldsymbol{A}^{(k)}, C_k)$ and $\mathcal{L}_{k'}(\boldsymbol{A}^{(k')}, C_{k'})$ using Eq. (14) and the procedure described below, conditioned on the networks in C_k and $C_{k'}$.
2. **Merge**: Pick two clusters $C_{k'}$ and $C_{k''}$ at random and merge them into a single cluster C_k. Compute the mode $\boldsymbol{A}^{(k)}$ that minimizes the cluster-level description length $\mathcal{L}_k(\boldsymbol{A}^{(k)}, C_k)$ using Eq. (14) and the procedure described below, conditioned on the networks in C_k. Finally, compute the change in the description length that results from this merge.
3. **Split**: Pick a cluster C_k at random and split it into two clusters $C_{k'}$ and $C_{k''}$ using the following 2-means algorithm. First assign every network in C_k to the cluster $C_{k'}$ or $C_{k''}$ at random. Refine the assignments by successively moving every network to the cluster $C_{k'}$ or $C_{k''}$ that results in a greater decrease in the description length and compute the modes $\boldsymbol{A}^{(k')}$ and $\boldsymbol{A}^{(k'')}$ that minimize the cluster-level description lengths $\mathcal{L}_{k'}(\boldsymbol{A}^{(k')}, C_{k'})$ and $\mathcal{L}_{k''}(\boldsymbol{A}^{(k'')}, C_{k''})$, conditioned on the networks now in $C_{k'}$ and $C_{k'}$. After convergence of the 2-means style algorithm, compute the change in the description length that results from this split of cluster C_k.
4. **Merge-split**: Pick two clusters at random, merge them as in move 2, then perform move 3 on this merged cluster. These two moves in direct succession help reassign multiple networks simultaneously; their addition to the move set improves the algorithm's performance.

Since these moves modify only one or two clusters, the change in the global description length $\mathcal{L}(\mathcal{D})$ can be recomputed quickly as updates to the cluster-level description lengths in Eq. (14). Every time a mode is needed for these calculations, we use the mode that minimizes the cluster-level description length $\mathcal{L}_k(\boldsymbol{A}^{(k)}, C_k)$ in Eq. (14). To find this optimal mode efficiently, we start with the "complete" mode

$$\boldsymbol{A}^{(k)}_{\text{comp}} = \bigcup_{s \in C_k} \boldsymbol{D}^{(s)}, \tag{16}$$

with an edge between nodes i and j if at least one network in the cluster contains the edge. We then greedily remove edges from $\boldsymbol{A}^{(k)}_{\text{comp}}$ in increasing order of occurrence in the networks of C_k—starting first with edges only found in a single network and going up from there—and update the cluster-level description length as we go. After removing all edges from $\boldsymbol{A}^{(k)}_{\text{comp}}$, the mode giving the lowest cluster-level description length is chosen as the mode for the cluster. This approach is locally optimal under a few assumptions about the sparsity of the networks and the composition of edges in the clusters (see Supplementary Note 2 for details).

We run the algorithm by starting with K_0 initial clusters (this choice has a negligible effect on the results, see Supplementary Note 3) and stop when a specified number of consecutive moves all result in rejections, indicating that the algorithm has likely converged. The worst-case complexity of this algorithm is roughly $O(NS)$ (the worst case is a split move right at the start). Supplementary Note 2 details the entire algorithm, and Supplementary Note 3 provides additional tests of the algorithm, such as its robustness for different choices of K_0.

To diagnose the quality of a solution, we compute the *inverse compression ratio*

$$\eta(\mathcal{D}) = \mathcal{L}_{\text{MDL}}(\mathcal{D}) / \mathcal{L}_0(\mathcal{D}), \tag{17}$$

where $\mathcal{L}_{\text{MDL}}(\mathcal{D})$ is the minimum value of $\mathcal{L}(\mathcal{D})$ over all configurations of $\mathcal{A}, \mathcal{C}$, given by the algorithm after termination, and $\mathcal{L}_0$ is given in Eq. (2). Equation (17) tells us how much better we can compress the network population $\mathcal{D}$ by using our multi-step encoding than by using the naïve fixed-length code to transmit all networks individually. If $\eta(\mathcal{D})<1$, our model compresses the data $\mathcal{D}$, and if $\eta(\mathcal{D})>1$, it does not because we waste too much information in the initial transmission steps.

Contiguous clusters. In the previous section, we described a merge-split Monte Carlo algorithm to identify the clusters $\mathcal{C}$ and modes $\mathcal{A}$ that minimize the description length in Eq. (15). This algorithm samples the space of unconstrained partitions $\mathcal{C}$ of the network population $\mathcal{D}$. However, in many applications, particularly in longitudinal studies, we may only be interested in constructing contiguous clusters, where each cluster is now a set of networks where adjacent indexes $s \in \{1, \ldots, S\}$ indicate contiguity of some form (temporal, spatial, or otherwise). Such constraints reduce the space of possible clusterings $\mathcal{C}$ drastically, and we can minimize the description length exactly (up to the greedy heuristic for the mode construction) using a dynamic program[32,53,54].

Before we introduce an optimization for this problem, we require a small modification to Eq. (14) for the cluster-level description length to accurately reflect the constrained space of ordered partitions $\mathcal{C}$ that we are considering. In our derivation of the description length, we assumed that the receiver knows the sizes $\{S_k\}$ of the clusters in $\mathcal{C}$. If we transmit these sizes in the order of the clusters they describe, the receiver will also know the exact clusters $\mathcal{C}$, since knowing the sizes $\{S_k\}$ is equivalent to knowing the cluster boundaries in this contiguous case. We can therefore ignore the term $S\log(S/S_k)$ in Eq. (14) that tells us how much information is required to transmit the exact cluster configuration. This modification results in a new, shorter description length

$$\mathcal{L}^{(\text{cont})}_k(\boldsymbol{A}^{(k)}, C_k) = \mathcal{L}(\mathcal{A}^{(k)}) + \ell_k \tag{18}$$

and a new global objective

$$\mathcal{L}_{\text{cont}}(\mathcal{D}) = \sum_k \mathcal{L}^{(\text{cont})}_k(\boldsymbol{A}^{(k)}, C_k). \tag{19}$$

Since the objective in Eq. (19) is a sum of independent cluster-level terms, minimizing this description length for contiguous clusters admits a dynamic programming algorithm solution[32,53,54] that can identify the true optima in polynomial time.

The algorithm is constructed by recursing on $\mathcal{L}^{(i)}_{\text{MDL}}$, the minimum description length of the first i networks in $\mathcal{D}$ according to Eq. (19). Since the objective function decomposes as a sum over clusters, for any $j \in [1, S]$, the MDL can be calculated as

$$\mathcal{L}^{(j)}_{\text{MDL}} = \min_{i \in [1,j]} \left\{ \mathcal{L}^{(i-1)}_{\text{MDL}} + \mathcal{L}^{(\text{cont})}_k([i,j]) \right\}, \tag{20}$$

where we set the base case to $\mathcal{L}^{(0)}_{\text{MDL}} = 0$ and define $\mathcal{L}^{(\text{cont})}_k([i,j])$ as the description length of the cluster of networks with indices $\{i, \ldots, j\}$, according to Eq. (18) with the mode computed with the greedy procedure described in the previous section. Once we recurse to $\mathcal{L}^{(S)}_{\text{MDL}}$, we have found the MDL of our complete dataset, and keeping tab of the minimizing i in Eq. (20) for every j allows us to reconstruct the clusters.

In practice, the recursion can be implemented from the bottom up, starting with $\mathcal{L}^{(1)}_{\text{MDL}}$, then $\mathcal{L}^{(2)}_{\text{MDL}}$, and so on. The computational bottleneck for calculating $\mathcal{L}^{(j)}_{\text{MDL}}$ is finding the modes of a cluster j times for each evaluation of Eq. (20) (once for each $i = 1, \ldots, j$), leading to an overall complexity $O(jN \log N)$ for this step. Summing over $j \in [1, S]$, the overall time complexity of the dynamic programming algorithm is $O(S^2 N \log N)$, which we verify numerically in Supplementary Note 3.

Data availability

The datasets used in this paper are available at https://github.com/aleckirkley/MDL-network-population-clustering.

Code availability

The algorithm presented in this paper is available at https://github.com/aleckirkley/MDL-network-population-clustering.

Received: 16 January 2023; Accepted: 15 June 2023;
Published online: 22 June 2023

References

1. Eagle, N., Pentland, A. S. & Lazer, D. Inferring friendship network structure by using mobile phone data. *Proc. Natl Acad. Sci. USA* **106**, 15274–15278 (2009).
2. Lehmann, S. in *Temporal Network Theory*, 25–48 (Springer, 2019).
3. Sporns, O. *Networks of the Brain* (MIT Press, 2010).
4. Stark, C. et al. Biogrid: a general repository for interaction datasets. *Nucleic Acids Res.* **34**, D535–D539 (2006).
5. Kivelä, M. et al. Multilayer networks. *J. Complex Netw.* **2**, 203–271 (2014).
6. Butts, C. T. Network inference, error, and informant (in)accuracy: a Bayesian approach. *Soc. Netw.* **25**, 103–140 (2003).
7. Newman, M. E. J. Estimating network structure from unreliable measurements. *Phys. Rev. E* **98**, 062321 (2018).
8. Young, J.-G., Cantwell, G. T. & Newman, M. Bayesian inference of network structure from unreliable data. *J. Complex Netw.* **8**, cnaa046 (2020).
9. Peixoto, T. P. Reconstructing networks with unknown and heterogeneous errors. *Phys. Rev. X* **8**, 041011 (2018).
10. Priebe, C. E., Sussman, D. L., Tang, M. & Vogelstein, J. T. Statistical inference on errorfully observed graphs. *J. Comput. Graph. Stat.* **24**, 930–953 (2015).
11. Arroyo, J. et al. Inference for multiple heterogeneous networks with a common invariant subspace. *J. Mach. Learn. Res.* **22**, 1–49 (2021).
12. Tang, R. et al. Connectome smoothing via low-rank approximations. *IEEE Trans. Med. Imaging* **38**, 1446–1456 (2018).

13. Lunagómez, S., Olhede, S. C. & Wolfe, P. J. Modeling network populations via graph distances. *J. Am. Stat. Assoc.* **116**, 2023–2040 (2021).
14. Wang, L. et al. Common and individual structure of brain networks. *Ann. Appl. Stat.* **13**, 85–112 (2019).
15. Young, J.-G., Valdovinos, F. S. & Newman, M. Reconstruction of plant–pollinator networks from observational data. *Nat. Commun.* **12**, 3911 (2021).
16. Banks, D. & Carley, K. Metric inference for social networks. *J. Classif.* **11**, 121–149 (1994).
17. Newman, M. E. J. Network structure from rich but noisy data. *Nat. Phys.* **14**, 542–545 (2018).
18. Le, C. M. et al. Estimating a network from multiple noisy realizations. *Electron. J. Stat.* **12**, 4697–4740 (2018).
19. La Rosa, P. S. et al. Gibbs distribution for statistical analysis of graphical data with a sample application to fcMRI brain images. *Stat. Med.* **35**, 566–580 (2016).
20. Young, J.-G., Kirkley, A. & Newman, M. E. J. Clustering of heterogeneous populations of networks. *Phys. Rev. E* **105**, 014312 (2022).
21. Stehlé, J. et al. High-resolution measurements of face-to-face contact patterns in a primary school. *PLoS ONE* **6**, e23176 (2011).
22. Peel, L. & Clauset, A. Detecting change points in the large-scale structure of evolving networks. In *Proc. 29th International Conference on Artificial Intelligence (AAAI)*, 2914–2920 (2015).
23. Peixoto, T. P. & Gauvin, L. Change points, memory and epidemic spreading in temporal networks. *Sci. Rep.* **8**, 15511 (2018).
24. De Domenico, M., Nicosia, V., Arenas, A. & Latora, V. Structural reducibility of multilayer networks. *Nat. Commun.* **6**, 6864 (2015).
25. Nielsen, A. M. & Witten, D. The multiple random dot product graph model. Preprint at https://arxiv.org/abs/1811.12172 (2018).
26. Wang, S., Arroyo, J., Vogelstein, J. T. & Priebe, C. E. Joint embedding of graphs. *IEEE Trans. Pattern Anal. Mach. Intell.* **43**, 1324–1336 (2019).
27. Stanley, N., Shai, S., Taylor, D. & Mucha, P. J. Clustering network layers with the strata multilayer stochastic block model. *IEEE Trans. Netw. Sci. Eng.* **3**, 95–105 (2016).
28. Signorelli, M. & Wit, E. C. Model-based clustering for populations of networks. *Stat. Model.* **20**, 9–29 (2020).
29. Mantziou, A., Lunagómez, S. & Mitra, R. Bayesian model-based clustering for multiple network data. Preprint at https://arxiv.org/abs/2107.03431 (2021).
30. Yin, F., Shen, W. & Butts, C. T. Finite mixtures of ERGMs for ensembles of networks. *Bayesian Anal.* **17**, 1153–1191 (2022).
31. Durante, D., Dunson, D. B. & Vogelstein, J. T. Nonparametric Bayes modeling of populations of networks. *J. Am. Stat. Assoc.* **112**, 1516–1530 (2017).
32. Patania, A., Allard, A. & Young, J.-G. Exact and rapid linear clustering of networks with dynamic programming. Preprint at https://arxiv.org/abs/2301.10403 (2023).
33. Rojas, A., Calatayud, J., Kowalewski, M., Neuman, M. & Rosvall, M. A multiscale view of the Phanerozoic fossil record reveals the three major biotic transitions. *Commun. Biol.* **4**, 309 (2021).
34. Vinh, N. X., Epps, J. & Bailey, J. Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. *J. Mach. Learn. Res.* **11**, 2837–2854 (2010).
35. Kirkley, A. Spatial regionalization based on optimal information compression. *Commun. Phys.* **5**, 249 (2022).
36. Eriksson, A., Edler, D., Rojas, A., de Domenico, M. & Rosvall, M. How choosing random-walk model and network representation matters for flow-based community detection in hypergraphs. *Commun. Phys.* **4**, 133 (2021).
37. Coupette, C., Dalleiger, S. & Vreeken, J. Differentially describing groups of graphs. In *The Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI-22)*, 3959–3967 (AAAI, 2022).
38. Rissanen, J. Modeling by the shortest data description. *Automatica* **14**, 465–471 (1978).
39. Hansen, M. H. & Yu, B. Model selection and the principle of minimum description length. *J. Am. Stat. Assoc.* **96**, 746–774 (2001).
40. Georgieva, O., Tschumitschew, K. & Klawonn, F. Cluster validity measures based on the minimum description length principle. In *Proc. International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, 82–89 (Springer, 2011).
41. Tabor, J. & Spurek, P. Cross-entropy clustering. *Pattern Recognit.* **47**, 3046–3059 (2014).
42. Li, T., Ma, S. & Ogihara, M. Entropy-based criterion in categorical clustering. In *Proc. Twenty-First International Conference on Machine Learning*, 68 (Association for Computing Machinery, 2004).
43. Narasimhan, M., Jojic, N. & Bilmes, J. A. Q-clustering. *Adv. Neural Inf. Process. Syst.* **18**, 979–986 (2005).
44. Rosvall, M. & Bergstrom, C. T. An information-theoretic framework for resolving community structure in complex networks. *Proc. Natl Acad. Sci. USA* **104**, 7327–7331 (2007).
45. Peixoto, T. P. Hierarchical block structures and high-resolution model selection in large networks. *Phys. Rev. X* **4**, 011047 (2014).
46. Koutra, D., Kang, U., Vreeken, J. & Faloutsos, C. Vog: summarizing and understanding large graphs. In *Proc. 2014 SIAM International Conference on Data Mining*, 91–99 (SIAM, 2014).
47. Wegner, A. E. Subgraph covers: an information-theoretic approach to motif analysis in networks. *Phys. Rev. X* **4**, 041026 (2014).
48. Bloem, P. & de Rooij, S. Large-scale network motif analysis using compression. *Data Min. Knowl. Discov.* **34**, 1421–1453 (2020).
49. Young, J.-G., Petri, G. & Peixoto, T. P. Hypergraph reconstruction from network data. *Commun. Phys.* **4**, 135 (2021).
50. Bouritsas, G., Loukas, A., Karalias, N. & Bronstein, M. Partition and code: learning how to compress graphs. *Adv. Neural Inf. Process. Syst.* **34**, 18603–18619 (2021).
51. Kirkley, A. & Newman, M. E. J. Representative community divisions of networks. *Commun. Phys.* **5**, 40 (2022).
52. Peixoto, T. P. Merge-split Markov chain Monte Carlo for community detection. *Phys. Rev. E* **102**, 012305 (2020).
53. Jackson, B. et al. An algorithm for optimal partitioning of data on an interval. *IEEE Signal Process. Lett.* **12**, 105–108 (2005).
54. Bellman, R. *Dynamic Programming* (Princeton University Press, 1957).
55. Cohen, K. M., Finney, S. C., Gibbard, P. L. & Fan, J.-X. The ICS international chronostratigraphic chart. *Episodes* **36**, 199–204 (2013).
56. Raup, D. M. & Sepkoski Jr, J. J. Mass extinctions in the marine fossil record. *Science* **215**, 1501–1503 (1982).

Acknowledgements

A.K. was supported in part by the HKU-100 Start Up Grant. M.R. was supported by the Swedish Research Council, Grant No. 2016-00796.

Author contributions

A.K. designed the study and methodology; A.K., A.R., M.R., and J.G.Y. designed the experiments; A.K. and J.G.Y. performed the experiments; A.R. and M.R. provided new datasets, and A.K. wrote the manuscript. All authors reviewed, edited and approved the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42005-023-01270-5.

Correspondence and requests for materials should be addressed to Alec Kirkley.

Peer review information *Communications Physics* thanks Nicolo Ruggeri, Alexander Gates and the other anonymous reviewer(s) for their contribution to the peer review of this work.

Reprints and permission information is available at http://www.nature.com/reprints

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