nature physics

SUPPLEMENTARY INFORMATION

DOI: 10.1038/NPHYS3374

# Percolation in real interdependent networks

Filippo Radicchi[1]

[1] *Center for Complex Networks and Systems Research, School of Informatics and Computing, Indiana University, Bloomington, USA**

## I. SITE PERCOLATION IN ISOLATED NETWORKS

The purpose of this section is twofold: (i) to illustrate in a simpler setting the heuristic approach we used in the main text to study site percolation in interdependent networks; (ii) to provide a connection between the exact results by Bollobás *et al.* for dense graphs [1], and the heuristic approaches for sparse graphs by Karrer *et al.* [2], and Hamilton and Pryadko [3].

Consider an undirected and unweighted graph composed of $N$ nodes and $E$ edges. The structure of the graph is encoded by the adjacency matrix $A$, a symmetric $N \times N$ matrix whose generic element $A_{ij} = 1$ if vertices $i$ and $j$ share an edge, and $A_{ij} = 0$, otherwise. Without loss of generality, we assume that, when all nodes are present in the network, the graph is formed by a single connected component. Let us consider an arbitrary value of the site occupation probability $p \in (0, 1)$, and indicate with $s_i$ the probability that the generic node $i$ is part of the largest cluster. The order parameter of the percolation transition is simply defined as the average of these probabilities over all nodes in the graph, i.e.,

$$P_\infty = \frac{1}{N} \sum_i s_i \, . \tag{S1}$$

Note that $s_i$ is a function of $p$, but, in the following, we omit this dependence for shortness of notation. As a first attempt, we can say that the probability $s_i$ for node $i$ to be part of the largest cluster is given by

$$s_i = p \, [1 - \prod_{j \in \mathcal{N}_i} (1 - s_j)] \, , \tag{S2}$$

where $\mathcal{N}_i$ is the set of neighbors of vertex $i$. The probability $s_i$ is written as the product of two contributions: (i) the probability that the node is occupied; (ii) the probability that at least one of its neighbors is part of the largest cluster. The attempt of Eq. (S3) relies on the so-called locally tree-like approximation [4]. In this ansatz, neighbors of node $i$ are not directly connected, and this allows us to consider the probabilities $s_j$ as independent variables. Tree-based approximations do not apply to regular lattices, but allow for surprisingly effective methods also in many real networks that are not strictly locally tree-like [2, 5]. Introducing the vectors $\vec{u}$ and $\vec{q}$, whose $i$-th components are respectively $u_i = \ln(1 - s_i)$ and $q_i = \ln(1 - s_i/p)$, we can write the set of coupled equations (S2) into the single vectorial equation

$$\vec{q} = A \, \vec{u} \tag{S3}$$

A trivial solution of Eq. (S3) is given by the configuration $\vec{u} = \vec{q} = \vec{0}$, corresponding to $\vec{s} = \vec{0}$ or $s_i = 0$ for all $i = 1, \ldots, N$. In the proximity of this configuration, we can make use of the truncated Taylor expansion $\ln(1 - x) = -x$, and Eq. (S3) can be approximated as

$$\vec{s} = pA\vec{s} \, , \tag{S4}$$

thus an eigenvalue/eigenvector equation. By the Perron-Frobenius theorem, the only solution having a physical meaning of this equation is obtained by setting $p = 1/\lambda$ and $\vec{s} = \vec{l}$, with $(\lambda, \vec{l})$ principal eigenpair of the adjacency matrix $A$. This tells us that the solution of Eq. (S2) is $s_i = 0$, for all $i = 1, \ldots, N$, if the site occupation probability is smaller than $1/\lambda$. In this region, the network is in the non-percolating regime. Slightly on the right of $1/\lambda$, the vector of probabilities $\vec{s}$ starts to grow in the direction of the principal eigenvector of the adjacency matrix, and the order parameter is not longer zero. For any value of the site occupation probability larger than $1/\lambda$, the network is in

*Electronic address: filiradi@indiana.edu.

DOI: 10.1038/NPHYS3374

the percolating phase. The site percolation threshold obtained using the heuristic approximation of Eq. (S2) is thus $p_c = 1/\lambda$. We remark that the equation $p_c = 1/\lambda$ has been previously obtained by Bollobás *et al.* in dense graph sequences [1]. Also, we stress that the set of heuristic Eqs. (S2) can be solved numerically to draw the percolation diagram of the network in this approximation.

The most serious limitation of Eq. (S2) is to introduce a positive feedback among probabilities. An increment in the probability $s_i$ produces an increase in the probabilities $s_j$ of the neighbors, which in turn causes an increment in the probability $s_i$, and so on. To avoid the presence of this self-reinforcement effect, we can introduce another set of heuristic equations according to which the probability that node $i$ belongs to the largest cluster is given by

$$s_i = p\,[\,1 - \prod_{j \in \mathcal{N}_i} (1 - r_{i\to j})\,]\,. \tag{S5}$$

These equations still rely on the locally tree-like ansatz. Here, $r_{i\to j}$ stands for the probability that node $j$ is part of the largest cluster disregarding whether vertex $i$ belongs to the largest cluster or not. We note that, while this quantity can be defined for any pair of nodes, only contributions given by adjacent vertices play a role in Eq. (S5). We can think $r_{i\to j}$ as one of the $2E$ components of a vector $\vec{r}$. In the definition of $\vec{r}$, every edge $(i, j)$ in the graph is responsible for two entries, i.e., $r_{i\to j}$ and $r_{j\to i}$. For consistency, the probability $r_{i\to j}$ is described by the following heuristic equation

$$r_{i\to j} = p\,[1 - \prod_{k \in \mathcal{N}_j \setminus \{i\}} (1 - r_{j\to k})\,]\,. \tag{S6}$$

The product of the r.h.s. of the last equation runs over all neighbors of node $j$ excluding vertex $i$. We note that the system of Eqs. (S6) is identical to the one obtained by Karrer *et al.* in Eq.(7) of the paper [2]. To see that just substitute $r_{i\to j}$ with their $1 - H_{i\leftarrow j}$. We note however that Karrer *et al.* wrote the same system of equations considering variables with a different meaning than $r_{i\to j}$, and for the bond percolation model. Their analogue of our Eq. (S5), however, does not contain the factor $p$ from which one recovers the correct result for bond percolation (see next section). It is convenient to rewrite Eq. (S6) as $\ln(1 - r_{i\to j}/p) = \sum_k A_{jk} \ln(1 - r_{j\to k}) - A_{ji} \ln(1 - r_{j\to i})$. Defining the vectors $\vec{w}$ and $\vec{t}$ such that their $(i \to j)$-th components are respectively given by $w_{i\to j} = \ln(1 - r_{i\to j})$ and $z_{i\to j} = \ln(1 - r_{i\to j}/p)$, the system of equations (S6) becomes equivalent to the vectorial equation

$$\vec{z} = M\,\vec{w}\,. \tag{S7}$$

The generic element of the $2E \times 2E$ square matrix $M$ is given by

$$M_{i\to j, k\to \ell} = \delta_{j,k}(1 - \delta_{i,\ell})\,, \tag{S8}$$

where the Kronecker delta function $\delta_{x,y} = 1$ if $x = y$, and $\delta_{x,y} = 0$, otherwise. Thus, the generic entry of the matrix $M$ is different from zero only if the ending node of the edge $i \to j$ corresponds to the starting vertex of the edge $k \to \ell$, but the starting and ending nodes $i$ and $\ell$ are different. This matrix is known as the non-backtracking matrix of the graph [6, 7]. A trivial solution of the preceding equation is given by $\vec{r} = \vec{0}$, which in turn leads to $\vec{s} = \vec{0}$. In proximity of this configuration, we can still make use of the truncated Taylor expansion of the logarithm, and approximate Eqs. (S5) and (S7) respectively as

$$s_i = p \sum_j A_{ij} r_{i\to j} \quad \text{and} \quad \vec{r} = p\,M\,\vec{r}\,. \tag{S9}$$

Using arguments similar to those applied to Eq. (S4), we can say that, according to Eq. (S9), the percolation threshold equals $p_c = 1/\mu$, with $\mu$ principal eigenvalue of the non-backtracking matrix of the graph, and that slightly on the right of the critical point the probability $s_i$ grows linearly with the sum of the components of the principal eigenvector of the non-backtracking matrix corresponding to edges pointing out from node $i$. We remark that the equation $p_c = 1/\mu$ has been previously derived by Karrer *et al.* [2], and Hamilton and Pryadko [3]. Also, we stress that the numerical solution of the heuristic Eqs. (S5) and (S6) provides a method to draw the entire percolation diagram, valid for the locally tree-like ansatz, for any finite network.

To summarize, the heuristics presented so far tell us two main interesting things. First, the difference between the two approaches resides only in the inclusion or exclusion of self-reinforcement effects among local variables. In this sense, Eqs. (S5) and (S6) represent an improvement to Eq. (S2), but both approaches are based on the same principles and approximations. This first observation serves to reunite recent predictions on percolation thresholds under the same theory [1–3]. Second, the way in which individual probabilities behave slightly on the right of the critical point allow us to understand why the prediction of the percolation threshold of Eq. (S9) may become inaccurate in networks with localized eigenstates of the non-backtracking matrix [8].

## II. BOND PERCOLATION IN ISOLATED NETWORKS

In the following, we will present a mapping between the site and bond percolation models in tree-like isolated networks. This mapping has been recently discovered by Radicchi and Castellano [9]. Here, we report the details only for illustrative purposes, and to show that the same mapping between site and bond percolation is valid also for interdependent networks (see next section).

Consider an undirected and unweighted graph composed of $N$ nodes and $E$ edges. Without loss of generality, we assume that, when all nodes and edges are present in the network, the graph is formed by a single connected component. We focus our attention on the bond percolation model. Here, nodes are always present while edges are occupied with probability $p$. For $p = 0$, there are no edges in the graph, each node is in its own cluster, and the relative size of the largest cluster is $P_\infty = 1/N$. For $p = 1$ instead, all edges are present, all nodes are in the same cluster, and $P_\infty = 1$. Let us consider an arbitrary value of the bond occupation probability $p \in (0, 1)$, and indicate with $v_i$ the probability that the generic node $i$ is part of the largest cluster. The order parameter of the percolation transition is simply defined as the average of these probabilities over all nodes in the graph, i.e.,

$$P_\infty = \frac{1}{N} \sum_i v_i \, . \tag{S10}$$

Note that $v_i$ is a function of $p$, but, in the following, we omit this dependence for shortness of notation. As a first attempt to describe the probability $v_i$, we can write the equation

$$v_i = 1 - \prod_{j \in \mathcal{N}_i} (1 - p\, v_j) \ , \tag{S11}$$

for all $i = 1, \ldots, N$. Eq. (S11) is formulated accordingly to the following straightforward argument. If node $j$ is in the set $\mathcal{N}_i$ of neighbors of vertex $i$, $pv_j$ is the probability that the connection between $i$ and $j$ is occupied, and node $j$ is part of the spanning cluster. Thus, the probability that node $i$ does not belong to the largest cluster, i.e., $1 - v_i$, is equal to the probability that none of its adjacent nodes, that are connected by an occupied edge, are part of the largest cluster of the graph. Note that Eq. (S11) assumes not only that the probabilities $v_j$ of all neighbors of node $i$ are uncoupled (i.e., the so-called locally tree-like approximation), but also that they do not depend on $v_i$. If we multiply both sides of Eq. (S11) for $p$, we obtain

$$p\, v_i = p\, [1 - \prod_{j \in \mathcal{N}_i} (1 - p\, v_j)] \, .$$

Now, if we call $pv_i = s_i$, we recover Eq. (S2), i.e., the equation describing the emergence of the largest cluster in site percolation. This simple relation extends all considerations deduced for site percolation to bond percolation.

The most serious limitation of Eq. (S11) is to introduce a positive feedback among probabilities. An increment in the probability $v_i$ produces an increase in the probabilities $v_j$ of the neighbors, which in turn causes an increment in the probability $v_i$, and so on. To avoid the presence of this self-reinforcement effect, we can change Eq. (S11) to

$$v_i = 1 - \prod_{j \in \mathcal{N}_i} (1 - p\, t_{i \to j}) \, . \tag{S12}$$

Here, $t_{i \to j}$ stands for the probability that node $j$ is part of the largest cluster discounting the contribution of node $i$. We note that while this quantity can be defined for any pair of nodes $i$ and $j$, only the contributions given by adjacent vertices, i.e., $A_{ij} = 1$, play a role in Eq. (S12). We can think to $t_{i \to j}$ as one of the components $2E$ of a vector $\vec{t}$, with every edge $(i, j)$ in the graph responsible for two entries ($t_{i \to j}$ and $t_{j \to i}$). In particular, the component $t_{i \to j}$ can be interpreted as the probability that following the edge $(i, j)$ in the direction $i \to j$, the node found at the end of the edge is part of the largest cluster. For consistency, the equation for the probability $t_{i \to j}$ is

$$t_{i \to j} = 1 - \prod_{k \in \mathcal{N}_j \setminus \{i\}} (1 - p\, t_{j \to k}) \, . \tag{S13}$$

The product of the r.h.s. of the last equation runs over all neighbors of node $j$ excluding vertex $i$. We note that the system of Eqs. (S6) is identical to the one obtained by Hamilton and Pryadko in Eq.(4) of the paper [3]. To see that just substitute $r_{i \to j}$ with their $1 - Q_{ij}$. We note however that Hamilton and Pryadko wrote the same system of equations considering the site percolation model, and not bond percolation as in our case. Now, if we multiply both sides of Eqs. (S12) and (S13) for $p$, and we call $p\, v_i = s_i$ and $p\, t_{i \to j} = r_{i \to j}$, we recover Eqs. (S5) and (S6) valid for site percolation. We can then extend to bond percolation all previous considerations obtained starting from Eqs. (S5) and (S6) for site percolation.

DOI: 10.1038/NPHYS3374

## III. BOND PERCOLATION IN INTERDEPENDENT NETWORKS

For interdependent networks, the analogy between site and bond percolation is still valid. If we forget self-reinforcement terms, we can write

$$v_i = Q_{\mathcal{AB}_i} + (1 - Q_{\mathcal{AB}_i})\, Q_{\mathcal{A}-\mathcal{B}_i}\, Q_{\mathcal{B}-\mathcal{A}_i}\,, \tag{S14}$$

where $Q_{\mathcal{X}} = 1 - \prod_{j\in\mathcal{X}}(1 - p\,v_j)$, and the sets $\mathcal{AB}_i$, $\mathcal{AB}_i$ and $\mathcal{B} - \mathcal{A}_i$ are defined as in the main text. In the previous equation, we supposed that the process of percolation involves all edges in the interconnected network, and shared edges among the layers are counted only once in the percolation model. In this model, the probability that node $i$ is part of the largest cluster of mutually connected nodes is given by: (i) the probability to be connected to the largest cluster thanks to at least one vertex that is connected to $i$ in both layers, and this is equal to $Q_{\mathcal{AB}_i}$; (ii) if the latter condition is not true, the probability that node $i$ is connected to the largest cluster through at least one node $k$ in layer $A$ and one node $\ell$ in layer $B$, with $k \neq \ell$, happening with probability $Q_{\mathcal{A}-\mathcal{B}_i}\, Q_{\mathcal{B}-\mathcal{A}_i}$. Please note that the decomposition of the interdependent network into an intersection graph, and two remainders is particularly intuitive in this percolation model. If we multiply Eq. (S14) for $p$, and again call $p\,v_i = s_i$, we recover Eq. (2) of the main text.

If we exclude self-reinforcement terms, we have

$$v_i = T_{\mathcal{AB}_i} + (1 - T_{\mathcal{AB}_i})\, T_{\mathcal{A}-\mathcal{B}_i}\, T_{\mathcal{B}-\mathcal{A}_i}\,, \tag{S15}$$

and

$$t_{i\to j} = T_{\mathcal{AB}_j\setminus\{i\}} + (1 - T_{\mathcal{AB}_j\setminus\{i\}})\, T_{\mathcal{A}-\mathcal{B}_j\setminus\{i\}}\, T_{\mathcal{B}-\mathcal{A}_j\setminus\{i\}}\,, \tag{S16}$$

with $T_{\mathcal{X}_i} = 1 - \prod_{j\in\mathcal{X}}(1 - p\,t_{i\to j})$. Again, by simply multiplying these equations for $p$, and setting $p\,v_i = s_i$ and $p\,t_{i\to j} = r_{i\to j}$, we recover Eqs. (3) and (4) of the main text.

## IV. TAYLOR EXPANSIONS

In the following, we will provide Taylor approximations of the various equations up to the second order. If we use the multidimensional Taylor expansion of the r.h.s. of Eq. (S2) around the trivial solution $\vec{s} = \vec{0}$ as

$$\begin{array}{l} [1 - \prod_{j\in\mathcal{N}_i}(1 - s_j)\,] \\ = \sum_k\ s_k\ \frac{d}{ds_k}\,[1 - \prod_{j\in\mathcal{N}_i}(1 - s_j)\,]\Big|_{\vec{s}=\vec{0}} + \\ \frac{1}{2}\sum_{k,v}\ s_k s_v\ \frac{d^2}{ds_k ds_v}\,[1 - \prod_{j\in\mathcal{N}_i}(1 - s_j)\,]\Big|_{\vec{s}=\vec{0}} + o(s_i^3) \\ = \sum_j A_{ij}\, s_j + \frac{1}{2}\sum_{j,k} A_{ij}A_{ik}s_j s_k(1 - \delta_{jk}) + o(s_i^3) \end{array}\,. \tag{S17}$$

In the equation above we used the fact that

$$\frac{d^2}{ds_k^2}\,[1 - \prod_{j\in\mathcal{N}_i}(1 - s_j)] = 0$$

because the variable $s_k$ can appear at maximum once in the product. In particular, Eq. (S4) is recovered by neglecting quadratic or higher order terms.

We can take the Taylor expansion also for the r.h.s. of Eq. (2) of the main text. Let us first imagine that the intersection graph does not contain edges, so that Eq. (2) of the main text reads as

$$s_i = p\, S_{\mathcal{A}-\mathcal{B}_i}\, S_{\mathcal{B}-\mathcal{A}_i}\,.$$

Since $S_{\mathcal{A}-\mathcal{B}_i}$ and $S_{\mathcal{B}-\mathcal{A}_i}$ calculated at $\vec{s} = \vec{0}$ are zero, the first derivatives of the r.h.s. calculated in $\vec{s} = \vec{0}$ are automatically zero. The Taylor expansion of the r.h.s. is thus

$$\begin{array}{l} \frac{1}{2}\sum_j\sum_k\ s_j s_k\ \frac{d^2}{ds_j\, ds_k}\, S_{\mathcal{A}-\mathcal{B}_i}\, S_{\mathcal{B}-\mathcal{A}_i}\Big|_{\vec{s}=\vec{0}} + o(s_i^3) = \\ \frac{1}{2}\sum_j\sum_k\ s_j s_k\ \frac{d}{ds_j}\, S_{\mathcal{A}-\mathcal{B}_i}\ \frac{d}{ds_k} S_{\mathcal{B}-\mathcal{A}_i}\Big|_{\vec{s}=\vec{0}} + o(s_i^3) \end{array}\,,$$

where the second equality is justified by the fact that $S_{\mathcal{A}-\mathcal{B}_i}$ and $S_{\mathcal{B}-\mathcal{A}_i}$ are zero at $\vec{s}=\vec{0}$. Using the definitions of $S_{\mathcal{A}-\mathcal{B}_i}$ and $S_{\mathcal{B}-\mathcal{A}_i}$ we have that

$$\frac{d}{ds_j}S_{\mathcal{A}-\mathcal{B}_i}\Big|_{\vec{s}=\vec{0}} = A_{ij}(1-B_{ij})$$

and

$$\frac{d}{ds_j}S_{\mathcal{B}-\mathcal{A}_i}\Big|_{\vec{s}=\vec{0}} = B_{ij}(1-A_{ij}) \ ,$$

where $A$ and $B$ are the adjacency matrices of the network layers. In conclusion, we can approximate Eq. (10) in absence of the intersection term as

$$s_i = \frac{p}{2}\,[\sum_j s_j\, A_{ij}(1-B_{ij})]\,[\sum_j s_j\, B_{ij}(1-A_{ij})] + o(s_i^3) \ .$$

Using Eq. (S17), we can also insert the term accounting for the intersection graph, and write

$$\begin{array}{rl} s_i = & p\sum_j A_{ij}B_{ij}s_j + \frac{p}{2}\sum_{j,k} A_{ij}A_{ik}B_{ij}B_{ik}s_j s_k(1-\delta_{jk}) \\ & +\frac{p}{2}\,[\sum_j s_j\, A_{ij}(1-B_{ij})]\,[\sum_j s_j\, B_{ij}(1-A_{ij})] + o(s_i^3) \end{array} \ . \tag{S18}$$

Essentially, the r.h.s. is given by a linear term due to the neighbors of node $i$ in the intersection graph, plus two quadratic terms: one given by the product of the variables $s$ over all pairs of neighbors of node $i$ in the intersection graph, and the other by the product of the sum of the variables $s$ over all neighbors in the two remainders.

Note that the we used the fact that on the r.h.s. of Eq. (2) of the main text

$$(1-S_{\mathcal{A}\mathcal{B}_i})\,S_{\mathcal{A}-\mathcal{B}_i}\,S_{\mathcal{B}-\mathcal{A}_i} = S_{\mathcal{A}-\mathcal{B}_i}\,S_{\mathcal{B}-\mathcal{A}_i} + o(s_i^3) \ .$$

One could repeat a similar type of calculations also for the r.h.s. of Eq. (S6). In this case, the derivatives are taken over the variables $r_{i\to j}$, and calculated around the trivial solution $\vec{r}=\vec{0}$. We obtain

$$\begin{array}{l} [1-\prod_{k\in\mathcal{N}_j\setminus\{i\}}(1-r_{j\to k})] = \\ \sum_{\ell\to v} r_{\ell\to v}\ \frac{d}{dr_{\ell\to v}}[1-\prod_{k\in\mathcal{N}_j\setminus\{i\}}(1-r_{j\to k})]\Big|_{\vec{r}=\vec{0}} + \\ \frac{1}{2}\sum_{\ell\to v,f\to g} r_{\ell\to v}r_{f\to g}\ \frac{d^2}{dr_{\ell\to v}dr_{f\to g}}[1-\prod_{k\in\mathcal{N}_j\setminus\{i\}}(1-r_{j\to k})]\Big|_{\vec{r}=\vec{0}} + o(r_{i\to j}^3) \\ \simeq \sum_{k\neq i} A_{jk}r_{j\to k} + \frac{1}{2}\sum_{k\neq i,v\neq k,i} A_{jk}A_{jv}r_{j\to k}r_{j\to k} \\ = \sum_k A_{jk}(1-\delta_{ki})r_{j\to k} + \frac{1}{2}\sum_{k,v} A_{jk}(1-\delta_{ki})A_{jv}(1-\delta_{vi})r_{j\to v}r_{j\to v}(1-\delta_{kv}) \end{array} \ . \tag{S19}$$

In particular, Eq. (S9) can be recovered by neglecting quadratic or higher order terms.

We can further proceed, and consider Eq. (4) of the main text. We expand the r.h.s. up to the second order, thus

$$(1-R_{\mathcal{A}\mathcal{B}_j\setminus\{i\}})\,R_{\mathcal{A}-\mathcal{B}_j\setminus\{i\}}\,R_{\mathcal{B}-\mathcal{A}_j\setminus\{i\}} = R_{\mathcal{A}-\mathcal{B}_j\setminus\{i\}}\,R_{\mathcal{B}-\mathcal{A}_j\setminus\{i\}} + o(r_{i\to j}^3) \ .$$

We have

$$\begin{array}{l} \sum_{\ell\to v,f\to g} r_{\ell\to v}r_{f\to g}\ \frac{d^2}{dr_{\ell\to v}dr_{f\to g}}\,R_{\mathcal{A}-\mathcal{B}_j\setminus\{i\}}\,R_{\mathcal{B}-\mathcal{A}_j\setminus\{i\}}\Big|_{\vec{r}=\vec{0}} \\ = \sum_{\ell\to v,f\to g} r_{\ell\to v}r_{f\to g}\ \frac{d}{dr_{\ell\to v}}\,R_{\mathcal{A}-\mathcal{B}_j\setminus\{i\}}\ \frac{d}{dr_{f\to g}}R_{\mathcal{B}-\mathcal{A}_j\setminus\{i\}}\Big|_{\vec{r}=\vec{0}} \\ = [\sum_k A_{jk}(1-B_{jk})(1-\delta_{ki})r_{j\to k}]\,[\sum_k B_{jk}(1-A_{jk})(1-\delta_{ki})r_{j\to k}] \end{array} \ .$$

In the previous equation, we used the fact that one edge can appear at maximum once in the remainders of the layers.

Finally, putting everything together the former equation and Eq. (S19) written for the intersection graph, we can write

$$\begin{array}{rl} r_{i\to j} = & p\sum_k A_{jk}B_{jk}(1-\delta_{ki})r_{j\to k} \\ & +\frac{p}{2}\sum_{k,v} A_{jk}B_{jk}(1-\delta_{ki})A_{jv}B_{jv}(1-\delta_{vi})r_{j\to k}r_{j\to v}(1-\delta_{kv}) \\ & +\frac{p}{2}\,[\sum_k A_{jk}(1-B_{jk})(1-\delta_{ki})r_{j\to k}]\,[\sum_k B_{jk}(1-A_{jk})(1-\delta_{ki})r_{j\to k}] \\ & +o(r_{i\to j}^3) \end{array} \ . \tag{S20}$$

The r.h.s. is given by a linear term due to the neighboring edges of the link $i\to j$ in the intersection graph, plus two quadratic terms: one given by the product of the variables $r$ over all pairs of neighboring edges in the intersection graph, and the other by the product of the sum of the variables $r$ over all neighboring edges in the two remainders. If the quadratic and higher order terms are neglected, the equation tells us that the variables $r$ grow smoothly from zero at $p=1/\mu_I$, with $\mu_I$ largest eigenvalue of the non-backtracking matrix of the intersection graph.

DOI: 10.1038/NPHYS3374

## V. PERCOLATION EQUATIONS FOR COUPLED REGULAR GRAPHS

Consider the hypothetical case of a system composed of two coupled networks where the intersection graph and the remainders of the two layers are regular graphs. Essentially, every vertex is connected to $k \geq 2$ other nodes in the intersection, and $t \geq 2$ other vertices in each of the remainders. We will assume to work in the limit $N \to \infty$, so that no size dependence is included in the following calculations. Given the regularity of the topology, one can assume that the probability $r_{i\to j} = r$ is the same for all edges $i \to j$. Eqs. (4) of the main text reduce thus to the single equation

$$r = p\left\{1 - (1-r)^{k-1} + (1-r)^{k-1}[1-(1-r)^{t-1}]^2\right\} , \tag{S21}$$

For arbitrary values of $k$ and $t$, the solution of the previous equation cannot be written in a closed form, being the problem equivalent to finding the roots of a polynomial of arbitrarily large order. Thus, in the general case, the equation can be solved only numerically. Once such a solution is obtained, we can determine the probability $s$ that a generic node belongs to the largest component using Eq. (3) of the main text, thus

$$s = p\left\{1 - (1-r)^{k} + (1-r)^{k}[1-(1-r)^{t}]^2\right\} . \tag{S22}$$

Please note that in this case the percolation strength is $P_\infty = s$.

One case that can be easily solved is the lowest possible order of the polynomial, which is obtained by setting $k = t = 2$ in Eq. (S21), leading to

$$r = p\left[r + (1-r)r^2\right] . \tag{S23}$$

One trivial solution is always $r = 0$. The non trivial solution $r > 0$ is obtained by solving the quadratic equation

$$r^2 - r - 1 + 1/p = 0 , \tag{S24}$$

thus,

$$r = \frac{1 \pm \sqrt{5 - 4/p}}{2} .$$

For the percolation problem, the solution having physical meaning is the one with the minus sign because $r$ cannot decrease as $p$ grows. The solution of the percolation problem is $r = 0$ for $p < 4/5$, and $r > 0$ for $p > 4/5$. In particular, $r = 1/2$ for $p = 4/5$. Plugging this solution into Eq. (S22) we have: $s = 0$ for $p < p_c$, $s > 0$ for $p \geq p_c$, with $p_c = 4/5$. In particular, the height of the jump in the order parameter is $s = 57/80$ for $p = p_c$.

For an arbitrary value of $k$, we can imagine that there is always a corresponding values of $t$, namely $t_c$, where we can observe a change in the nature of the percolation transition: for $t < t_c$, the order parameter changes smoothly; for $t > t_c$, the order parameter has instead a discontinuity. As we remarked, $t_c$ cannot be determined analytically but can be obtained with sufficient accuracy from the numerical solutions of the equations. The numerical results presented in Fig. S1 show that $t_c \simeq -0.47 + 1.12k$. In the region $t > t_c$, we note also that the jump in the order parameter becomes less and even less noticeable as $t$ and $k$ increase, as we expect for dense graphs. We can further note that the largest eigenvalue of the non-backtracking matrix of the network remainders is $\mu_{A,B} = t - 1$, while for the intersection graph is $\mu_I = k - 1$. If we insert this definitions in the former equation for $t_c$, we find the boundary between the first and second-order percolation transition is roughly given, although not perfectly, by the equation $\mu_I - \mu_{A,B} = 0$.

## VI. NUMERICAL SOLUTION OF THE EQUATIONS

Numerical solutions of the system of Eqs. (4) of the main text are obtained by iteration. We calculate the value of the variables $\vec{r}^{(t)}$ for iteration $t$ using the values of the variables $\vec{r}^{(t-1)}$ calculated at iteration $t-1$. The algorithm starts from a given initial configuration $\vec{r}^{(t=0)}$. Below, we will provide more details about the initial configuration we used depending on the application. In the iteration of the equations, we consider the solution achieved with sufficient accuracy, if at iteration $t_f$, we have

$$\frac{1}{2E}\sum_{i\to j} |r_{i\to j}^{(t_f)} - r_{i\to j}^{(t_f-1)}| < \epsilon .$$

$E$ equals the sum of all edges in the intersection, and in the remainders of the layers. In our numerical simulations, we set $\epsilon = 10^{-7}$. The solution $\vec{r}^{(t_f)}$ is then plugged into Eqs. (3) of the main text to evaluate the vector of probabilities $\vec{s}$ for individual nodes, and these values are finally used to compute the percolation strength $P_\infty$ with Eq. (1) of the main text.

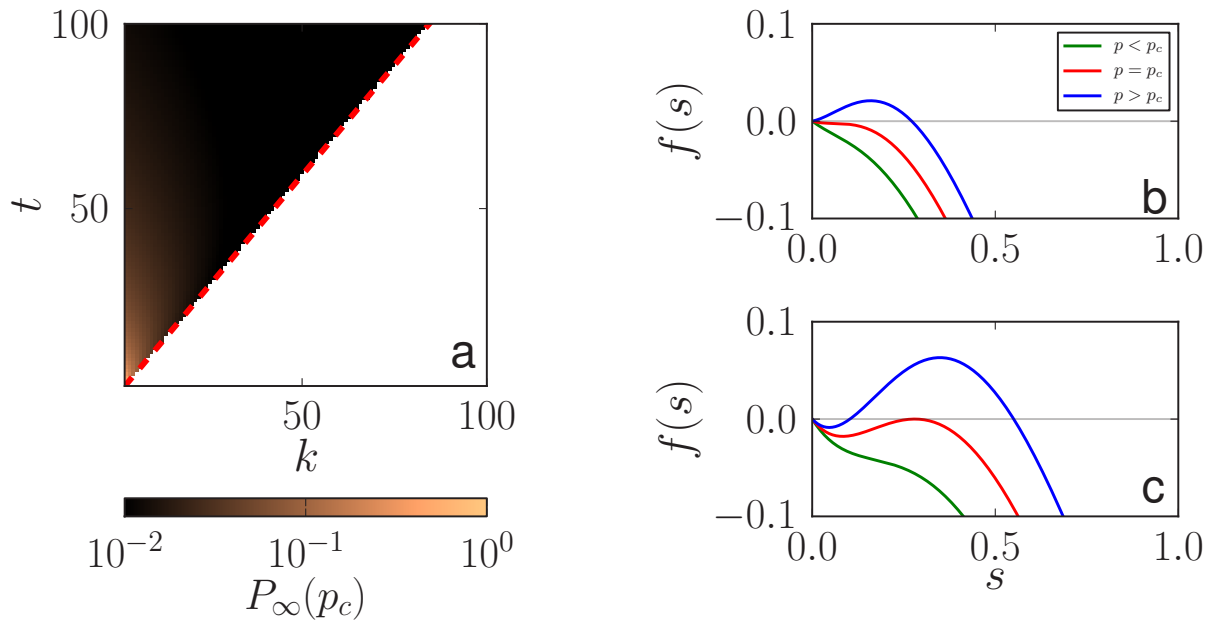


Figure S1: **a)** Height of the discontinuous jump in the percolation strength at criticality $P_\infty(p_c)$ for a system of two interdependent networks whose intersection is given by a regular graph with valency $k$, and whose remainders of the layers are still given by regular graphs with valency $t$. The white region in the plot denotes null jumps in the percolation strength. The red line approximates the boundary between the regions with different critical behavior. The line is given by the best fit of a simple linear regression model with the data points. The equation of the line is $t_c = -0.47 + 1.19\,k$. **b)** Graphical solution of Eqs. (S21) and (S22) for $k = t = 4$. The solution of the problem is *de facto* determined by finding, as $p$ varies, the roots of the polynomial $f(s) = p\left\{1-(1-r)^k + (1-r)^k[1-(1-r)^t]^2\right\} - s$, where $r$ is the solution of the polynomial equation $r = p\left\{1-(1-r)^{k-1} + (1-r)^{k-1}[1-(1-r)^{t-1}]^2\right\}$. For $k = t = 4$, a novel root, different from $s = 0$, emerges smoothly at $p_c$ (i.e., there is an infinitesimal gap between the value of the new root and $s = 0$). **c)** For $k = 2$ and $t = 4$ instead, the new root $s > 0$ of the former equation suddenly appears at the critical point (i.e., there is a finite gap between the value of the new root and $s = 0$).

### A. Implementation of a single iteration

At iteration $t$, the inputs of our numerical method are the vector $\vec{r}^{(t-1)}$, whose $2E$ entries correspond to the value of the variables $r_{i\to j}^{(t-1)}$ obtained at iteration $t-1$, and the value of occupation probability $p$. To speed up the execution, it is convenient to store, in a separate vector, the position (i.e., intersection, remainder $A$, or remainder $B$) of every edge. The output of a single iteration is the vector $\vec{r}^{(t)}$ whose $2E$ entries represent the value of the variables $r_{i\to j}^{(t)}$ at iteration $t$ of the algorithm. To be more specific, we use the following algorithmic procedure:

1. We define three vectors $\vec{u}$, $\vec{a}$, and $\vec{b}$, with $N$ components each, where $N$ is the total number of nodes present in the network. At the beginning of the iteration all the components of these vectors are set equal to one. This step has computational complexity equal to $N$.

2. For every edge $i \to j$, we have three possibilities:

   (a) If the edge $i \to j$ belongs to the intersection, then we update the $i$th entry of the vector $\vec{u}$, i.e., $u_i \to u_i\,(1 - r_{i\to j}^{(t-1)})$;

   (b) If the edge $i \to j$ belongs to the remainder of layer $A$, then $a_i \to a_i\,(1 - r_{i\to j}^{(t-1)})$;

   (c) If the edge $i \to j$ belongs to the remainder of layer $B$, then $b_i \to b_i\,(1 - r_{i\to j}^{(t-1)})$.

   Since we have to run this procedure on every edge $i \to j$, this step has computational complexity equal to $2E$.

3. For every edge $i \to j$, we determine:

DOI: 10.1038/NPHYS3374

(a) $R_{\mathcal{AB}_j\setminus\{i\}} = 1 - u_j/(1 - r_{j\to i}^{(t-1)})]$ if the edge $i \to j$ belongs to the intersection, and $R_{\mathcal{AB}_j\setminus\{i\}} = 1 - u_j$, otherwise.

(b) $R_{\mathcal{A}-\mathcal{B}_j\setminus\{i\}} = 1 - a_j/(1 - r_{j\to i}^{(t-1)})]$ if the edge $i \to j$ belongs to the remainder of layer $A$, and $R_{\mathcal{A}-\mathcal{B}_j\setminus\{i\}} = 1 - a_j$, otherwise.

(c) $R_{\mathcal{B}-\mathcal{A}_j\setminus\{i\}} = 1-b_j/(1-r_{j\to i}^{(t-1)})]$ if the edge $i \to j$ belongs to the remainder of layer $B$, and $R_{\mathcal{A}-\mathcal{B}_j\setminus\{i\}} = 1-b_j$, otherwise.

We finally compute $r_{i\to j}^{(t)} = pR_{\mathcal{AB}_j\setminus\{i\}}+p(1-R_{\mathcal{AB}_j\setminus\{i\}})R_{\mathcal{A}-\mathcal{B}_j\setminus\{i\}}R_{\mathcal{A}-\mathcal{B}_j\setminus\{i\}}$. Since we have to run this procedure on every edge $i \to j$, this step has computational complexity equal to $2E$.

### B. Drawing the phase diagram

The algorithmic procedure described above allows to find the solution to the equations for any given value of $p$. If one wants to draw the entire phase diagram in the interval $[p_1, p_2]$ with precision $dp$, a good way to reduce the computational time is to start from the upper bound $p = p_2$, and find the solution of the equations by iteration starting from the initial configuration $\vec{r}^{(t=0)} = \vec{1}$, i.e., all components of the vector are equal to one. Denote the solution obtained for the upper bound as $\vec{r}(p_2)$. Then,

1. Decrease $p \to p - dp$.
2. Solve the equations starting from the configuration $\vec{r}^{(t=0)} = \vec{r}(p+dp)$, i.e., the solution of the equations obtained at the previous value of $p$.
3. Go back to point 1 until $p \geq p_1$.

This recipe accounts for the fact that the value of the variables $r_{i\to j}$ cannot increase as $p$ decreases. Using as initial configuration the solution obtained at the previous step decreases substantially the number of iterations required to find the new solution. In particular, if, at a particular value of $p^*$, we find $\vec{r}(p^*) = \vec{0}$, then for $p \leq p^*$ the solution $\vec{r}(p) = \vec{0}$ is obtained with a single iteration of the algorithm. The phase diagrams plotted in the figures of the main text have been obtained using $p_1 = dp$, $p_2 = 1 - dp$, and $dp = 10^{-3}$. Estimates of the computational time required to draw the entire percolation diagram for some interdependent networks composed of graph models are provided in Fig. S2. We note that the time required to achieve convergence in the iterative part of the algorithm introduces only a factor $\ln(E)$, and thus it does not change dramatically the computational complexity of the algorithm.

### C. Identification of point discontinuities in the phase diagram

This section is devoted to the description of the methods we implemented to identify the eventual presence of a point discontinuity in the order parameter $P_\infty$ as a function of the occupation probability $p$. The function $P_\infty(p)$ has a point discontinuity in $p = p_c$ if the left and right-hand limits of the function in $p = p_c$ are different

$$P_\infty(p_c^-) = \lim_{p\to p_c^-} P_\infty(p) \neq \lim_{p\to p_c^+} P_\infty(p) = P_\infty(p_c^+) \, .$$

We implemented two different numerical procedures for the identification of such discontinuity. In both, we assumed that $P_\infty$ has at maximum a single discontinuity. This fact is supported by numerical evidence in all networks we considered in our paper.

The first procedure applies to discontinuities of $P_\infty$ starting off from zero to a finite value, i.e., $P_\infty(p_c^-) = 0$. The identification of the jump in the order parameter can be efficiently performed by adopting a binary search. The interval $[p_l, p_L]$ where the discontinuity is located is progressively reduced by changing the value of the lower bound $p_l$ with the constraint $P_\infty(p_l) = 0$, and the one of the upper bound $p_L$ subjected to the constraint $P_\infty(p_L) > 0$. This procedure is computationally efficient, and allows for the location of the discontinuous jump, if present, in a relatively fast way and with sufficiently high accuracy. We use this procedure in the finite-size scaling analyses of Fig. 2b, 2c, 2f, and 2g of the main text, and Fig. S3. In our analysis, the position of the jump is obtained with an accuracy equal to $10^{-5}$. In this case, we indicate the height of the jump with $P_\infty(p_c)$ without specifying the fact that this value corresponds to the right-hand limit of the function $P_\infty$ at $p = p_c$.

The second procedure that we implemented is required when the jump in the order parameter does not start off from zero, i.e., $P_\infty(p_c^-) > 0$. This is for example visible in Fig. 2d of the main text. In such a case, the binary search

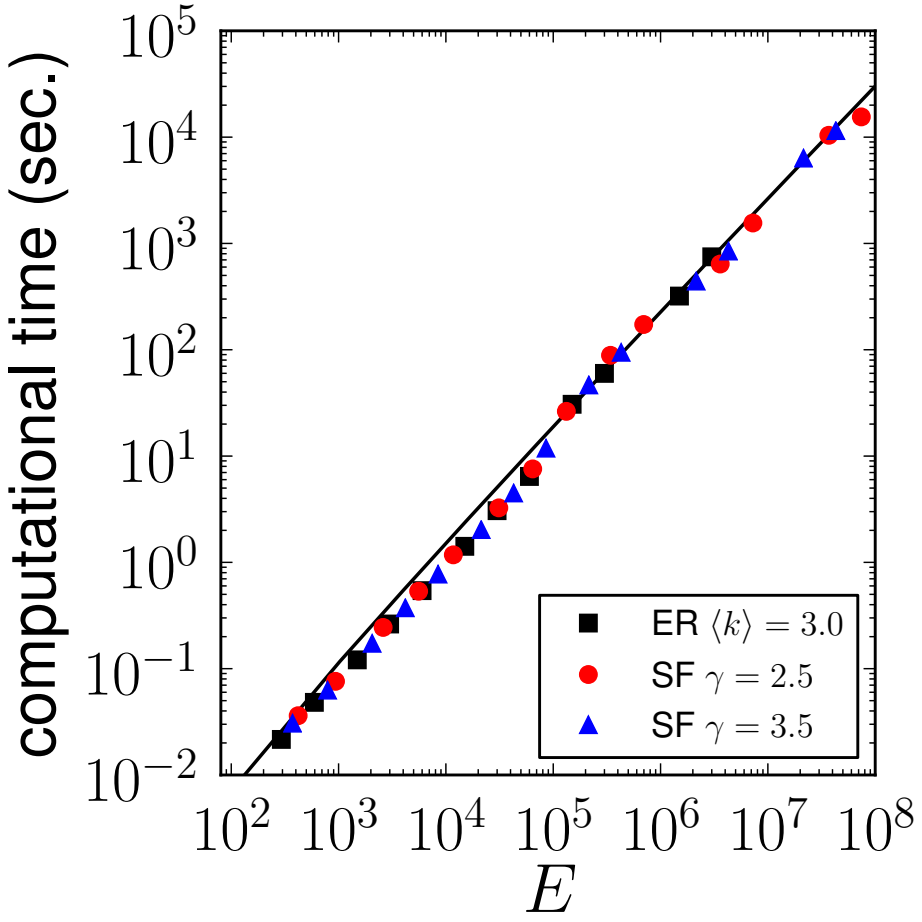


Figure S2: Estimate of the computational time required to draw the entire percolation diagram for an interdependent network formed by two Erdős-Rényi graphs with average degree $\langle k \rangle = 3.0$ (black circles), and scale-free graphs with minimal degree $k_{min} = 3$, and degree exponent $\gamma = 2.5$ (red circles) and $\gamma = 3.5$ (blue triangles). Results have been obtained on an `Intel(R) Xeon(R) CPU E5-2695 v2 2.40GHz` machine. Each point corresponds to the average value over at least ten realizations of the network models (only five realizations for networks with size $N > 10^6$). Standard deviations have size comparable with the symbol size. We draw the percolation diagram by considering any value of $p$ in the interval $[0, 1]$ with a precision $dp = 10^{-3}$. Numerical convergence of the iterative algorithm is achieved with precision $\epsilon = 10^{-7}$. The computational time $\tau$ is plotted as a function of the number of edges $E$ in the network. We test the scaling $\tau \sim E \ln(E)$ (black line).

described above cannot be used to locate the position of the discontinuity in the order parameter. We therefore apply an extensive search for such discontinuity by drawing the entire phase diagram. We use this more computationally expensive procedure in the finite-size scalings of Figs. S4 and S5. Also here, the position of the jump is obtained with an accuracy equal to $10^{-5}$. In this case, we indicate the height of the jump with $P_\infty(p_c^+) - P_\infty(p_c^-)$ specifying the fact that this value corresponds to the difference between the right and left-hand limits of the function $P_\infty$ at $p = p_c$.

## VII. NUMERICAL SIMULATIONS OF THE PERCOLATION MODEL

In several figures of the main text, we compare the results obtained with the solution of our heuristic equations, and those obtained instead by directly simulating the percolation process and estimating the size of the largest cluster of mutually connected nodes. Here, we report details regarding the algorithm implemented in our simulations of the percolation model. The algorithm takes as inputs the occupation probability $p$, and the network layers $A$ and $B$. As a first step, nodes are marked as occupied with probability $p$. The remaining vertices are instead marked as non occupied. Then, we use the following procedure:

DOI: 10.1038/NPHYS3374

1. Identify the nodes that belong to the largest cluster in layer $A$, and those that belong to the largest cluster in layer $B$. These clusters can be formed only by nodes that are marked as "occupied". Edges attached to "non occupied" nodes cannot be used to form clusters.

2. Every node belonging to the largest cluster in layer $A$ but not to the largest cluster in layer $B$ is marked as non occupied. The same happens to all vertices within the largest cluster in $B$, but not in $A$.

3. Go back to point 1.

The procedure ends when the number of nodes simultaneously within the largest clusters of both layers does not longer change. This number provides an approximate for the size of the largest cluster of mutually connected nodes. The procedure described above does not necessarily guarantee that the largest cluster of mutually connected nodes is correctly quantified. This cluster in fact could be in principle given by combinations of smaller clusters within the layers. The problem may arise more frequently for values of $p$ smaller than the percolation threshold. We expect, however, that the problem mentioned above becomes negligible when $p > p_c$. The comparison between ours and numerical results obtained previously [10, 11] justifies the approximation made in the numerical estimation of the size of the largest cluster of mutually connected nodes.

## VIII. GENERATION OF RANDOM NETWORKS

The generation of a single instance of the Erdős-Rényi model with $N$ nodes and average degree $\langle k \rangle$ is obtained by connecting each pair of nodes with probability $\langle k \rangle / (N-1)$.
To generate a random network with $N$ nodes and power-law degree distribution

$$P(k) \begin{cases} \sim k^{-\gamma} & \text{, if } k \in [k_{min}, \sqrt{N}] \\ = 0 & \text{, oth.} \end{cases} , \tag{S25}$$

we make use of the so-called uncorrelated configuration model (UCM) [12, 13]. In Eq. (S25), $k_{min}$ indicates the the minimal degree imposed in the network. The support $[k_{min}, \sqrt{N}]$ of the degree distribution is chosen in such a way that the resulting network has no degree-degree correlations, and is always composed of a single connected component. In the generation of a single instance of the network model, we first assign degrees to the nodes according to the prescribed $P(k)$. Then, we attach pairs of nodes at random, preserving their pre-imposed degrees, but not allowing for multiple connections and self-loops.

## IX. REAL NETWORKS

In each of the networks listed in Table S1, we restrict our attention only on the portion of the system with nodes present in the both layers, and we exclude all vertices that appear only in one of the network layers but not in the other.

## X. FINITE-SIZE SCALING ANALYSIS

In Figs. 2g and 2f of the main text, we perform a finite-size scaling analysis on coupled random scale-free networks. This section serves to illustrate how we obtain the best estimates of the percolation threshold and the height of the jump of the order parameter valid in the infinite limit size.

For given values of the network size $N$ and degree exponent $\gamma$, we generate a model of the system. We then estimate the critical point $p_c(N)$ and the height of the jump $P_\infty[p_c(N)]$ using our equations, and the procedure described above for the identification of the discontinuity point in the order parameter. The values of $p_c(N)$ and $P_\infty[p_c(N)]$ used in our fits are given by average values over several realizations of the network model (from 1000 realizations for small $N$ values, up to 10 realizations for large values of $N$). To extrapolate the asymptotic value $p_c$, we make the hypothesis that $p_c(N) = p_c + aN^{-\alpha}$. Then, we perform a simple linear regression fit between $\log[p_c(N) - p]$ and $\log(N)$ to test the former hypothesis, where $p$ is a free parameter ranging in the interval $[0, \min(p_c(N))]$. We determine the best estimate of $p_c$ as the value of $p$ for which the squared coefficient of variation $R^2$ of the linear regression model is maximal (see Fig. S3). The slope of the best linear fit determines the best estimate of the decay exponent $\alpha$. The same exact technique is used to determine $P_\infty(p_c)$ and $\beta$ for the relation $P_\infty[p_c(N)] = P_\infty(p_c) + bN^{-\beta}$. Results are visualized in Fig. S3, and in Fig. 2f and 2g of the main text. Best estimates of the parameters of the linear fits are reported in Table S2.

| network | layer A | layer B | $N$ | $N_I$ | $E_I$ | $\mu_I$ | $N_A$ | $E_A$ | $\mu_A$ | $N_B$ | $E_B$ | $\mu_B$ | ref. | url |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *H. sapiens* | direc. inter. | phys. assoc. | 9553 | 4958 | 11965 | 22.63 | 8626 | 30412 | 27.40 | 8960 | 56220 | 87.62 | [14, 15] | url |
| | direc. inter. | coloc. | 4465 | 653 | 1362 | 6.22 | 4063 | 18329 | 33.93 | 3954 | 13371 | 53.10 | [14, 15] | url |
| | phys. assoc. | coloc. | 5202 | 1057 | 2218 | 12.76 | 5086 | 40280 | 83.55 | 4536 | 15377 | 57.10 | [14, 15] | url |
| *C. elegans* | elect. | chem. mon. | 238 | 49 | 111 | 3.84 | 209 | 662 | 5.81 | 229 | 374 | 9.20 | [15] | url |
| | elect. | chem. pol. | 252 | 91 | 162 | 4.43 | 199 | 349 | 5.65 | 246 | 1293 | 16.34 | [15] | url |
| | chem. mon. | chem. pol. | 259 | 242 | 630 | 9.23 | 173 | 257 | 3.59 | 222 | 946 | 12.49 | [15] | url |
| *US air transp.* | Delta | American | 84 | 49 | 68 | 3.55 | 79 | 374 | 18.44 | 76 | 190 | 12.47 | - | url |
| | Delta | United | 82 | 39 | 56 | 3.33 | 78 | 348 | 17.45 | 78 | 226 | 14.70 | - | url |
| | American | United | 73 | 42 | 68 | 4.92 | 68 | 161 | 12.14 | 69 | 202 | 14.65 | - | url |

Table S1: Summary data for the real networks analyzed in the main text. From left to right, we include: name of the network, type of connections on layer A and layer B, total number of nodes $N$ in the system, number of nodes $N_I$ in the largest component of the intersection graph, total number of edges $E_I$ in the intersection graph, largest eigenvalue $\mu_I$ of the non-backtracking matrix for the largest component of the intersection graph, number of nodes $N_A$ in the largest component of the remainder of layer A, total number of edges $E_A$ in the remainder of layer A, largest eigenvalue $\mu_A$ of the non-backtracking matrix for the largest component of the remainder of layer A, number of nodes $N_B$ in the largest component of the remainder of layer B, total number of edges $E_B$ in the remainder of layer B, largest eigenvalue $\mu_B$ of the non-backtracking matrix for the largest component of the remainder of layer B, reference to papers where the dataset was first studied, hyperlink to the web site where data have been obtained.

| $\gamma$ | $p_c$ | $\alpha$ | $R^2$ | $P_\infty(p_c)$ | $\beta$ | $R^2$ |
|---|---|---|---|---|---|---|
| 2.3 | 0.17 | 0.29 | 1.00 | 0.01 | 0.32 | 1.00 |
| 2.7 | 0.25 | 0.42 | 1.00 | 0.07 | 0.37 | 1.00 |
| 3.5 | 0.34 | 0.68 | 0.99 | 0.16 | 0.61 | 0.99 |

Table S2: Values of the best estimates for the parameters of the power-law fits performed in Fig. S3 and Figs. 2f and 2g of the main text. For every value of the degree exponent $\gamma$, we report the results obtained for the finite-size scaling of the percolation threshold, i.e., $p_c(N) = p_c + aN^{-\alpha}$, and those valid for the finite-size scaling of the height of the order parameter at criticality $P_\infty[p_c(N)] = P_\infty(p_c) + bN^{-\beta}$. Also, we show the respective values of the squared coefficients of variation of the two fits.

## XI. IDENTICAL NETWORK LAYERS WITH RANDOMIZED NODE LABELS

To understand more quantitatively how the percolation in interdependent network layers changes its features, we take advantage of a simple model inspired by a recent work by Bianconi and Dorogovtsev [16]. The graphs used in both layers are identical. However, the labels of the nodes of one of the two layers, say layer $A$, are randomized with probability $q$. The randomization procedure is obtained in the following way. Nodes are initially labeled in the same way in both layers. With probability $q$, the label of every node in layer A is exchanged with the one of another randomly chosen node. In the unperturbed configuration for $q = 0$, all edges are in the intersection graph. For $q = 1$ instead, we expect that the number of edges in the intersection is much smaller than the number of edges in the remainders of the layers.

In our analysis, we consider two different models to generate the structure of the graph: ER graphs with average degree $\langle k \rangle = 3$, and scale-free graphs with degree exponent $\gamma = 2.5$ and minimal degree $k_{min} = 3$. In Figs. 2d and 2h of the main text, we present a comparison between the results of numerical simulations and those obtained with the numerical solution of Eqs. (3) and (4) of the main text for both models with fixed size $N = 10^4$, and various values of $q$. We note a transition from a smooth to an abrupt behavior in the order parameter as a function of $q$. Here, we provide a more detailed analysis of the behavior of the model. In particular, we study its properties with a finite-size scaling analysis. Note that the eventual discontinuity in the order parameter does not generally start off from zero, thus we make use of the second strategy described above to determine the location and height of the jump in the percolation strength. Fig. S4a displays the quantity $P_\infty(p_c^+) - P_\infty(p_c^-)$ as a function of $q$ for several network sizes $N$. We multiply the height of the jump by the factor $\mu_{A,B} - \mu_I$. $\mu_I$ is the largest eigenvalue of the non-backtracking matrix associated to the largest connected component of the intersection graph. $\mu_{A,B} = \min\{\mu_A, \mu_B\}$ is the smallest among the largest eigenvalues of the non-backtracking matrices associated to the largest connected component of the remainders of networks $A$ and $B$. This factor is able to account for variations in the height of the jump as $N$ and $q$

DOI: 10.1038/NPHYS3374

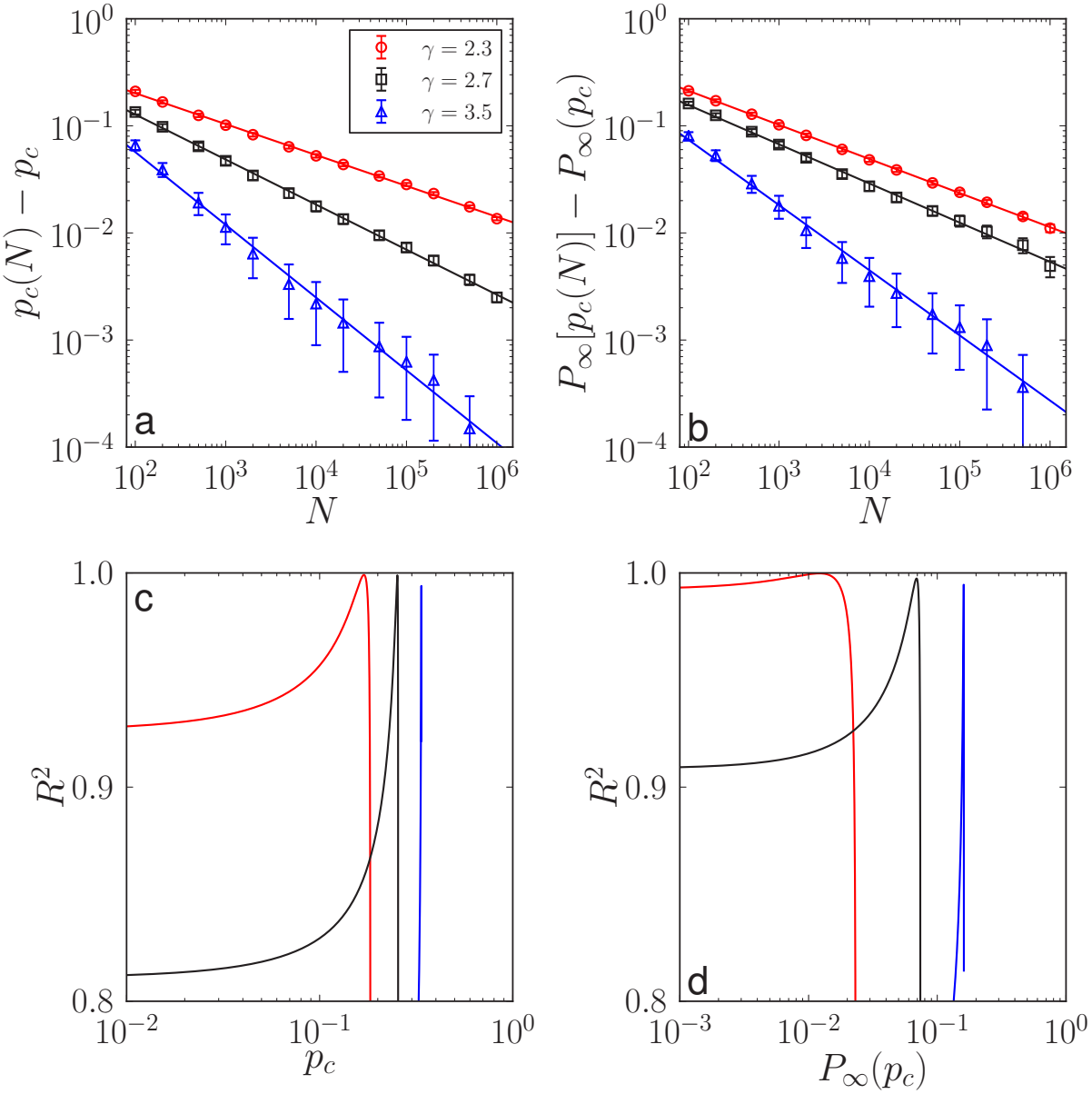


Figure S3: Finite-size analysis of scale-free networks with degree distribution $P(k) \sim k^{-\gamma}$ if $k \in [3, \sqrt{N}]$, and $P(k) = 0$, otherwise. **a)** As the size of the network grows, the pseudo-critical threshold $p_c(N)$ gets closer to the critical threshold $p_c$ in a power-law fashion, i.e., $p_c(N) - p_c \sim N^{-\alpha}$ (solid lines). Points indicate average values obtained over at least 10 independent realizations of the network model, error bars quantify instead the standard deviation of the measures across different realizations. The best estimates of $p_c$ and $\alpha$, as well as the $R^2$ of the linear fit, are reported in Table S2. **b)** Same as in panel a but for the height of the jump of the percolation strength at pseudo-criticality. In this case, the solid line stand for the best estimate of the scaling $P_\infty[p_c(N)] - P_\infty(p_c) \sim N^{-\beta}$. The best estimates of $P_\infty(p_c)$ and $\beta$, and the $R^2$ value of the linear fit are tabulated in Table S2. **c)** Coefficient of variation $R^2$ for the linear fits in panel a. The value of $p_c$ corresponding to the peak of $R^2$ is assumed to be the best estimate of the asymptotic value of the percolation threshold. **d)** Same as in panel c, but for the linear fits of panel b.

vary. We note that in the ER model, all curves collapse one on the top of the other. Further a nonvanishing jump is visible if $\mu_I - \mu_{A,B} < 0$. For the scale-free models instead, all curves crosses at the same value of $q$, namely $q_c$. The height of the jump in the limit $N \to \infty$ is non vanishing for all values of $q > q_c$, while becomes zero for $q < q_c$. This fact is proven extrapolating the asymptotic values $p_c$ and $P_\infty(p_c)$ (see Fig. S5 and Table S3). Differently from the case of the ER model, $q_c$ does not correspond to the point where there is an inversion in the sign of $\mu_I - \mu_{A,B}$. This fact

can be intuitively explained by accounting for the heterogeneity of the degree distribution. Hub-to-hub connections appear too frequently in the intersection graph, unless the probability of relabeling the hubs becomes sufficiently high.

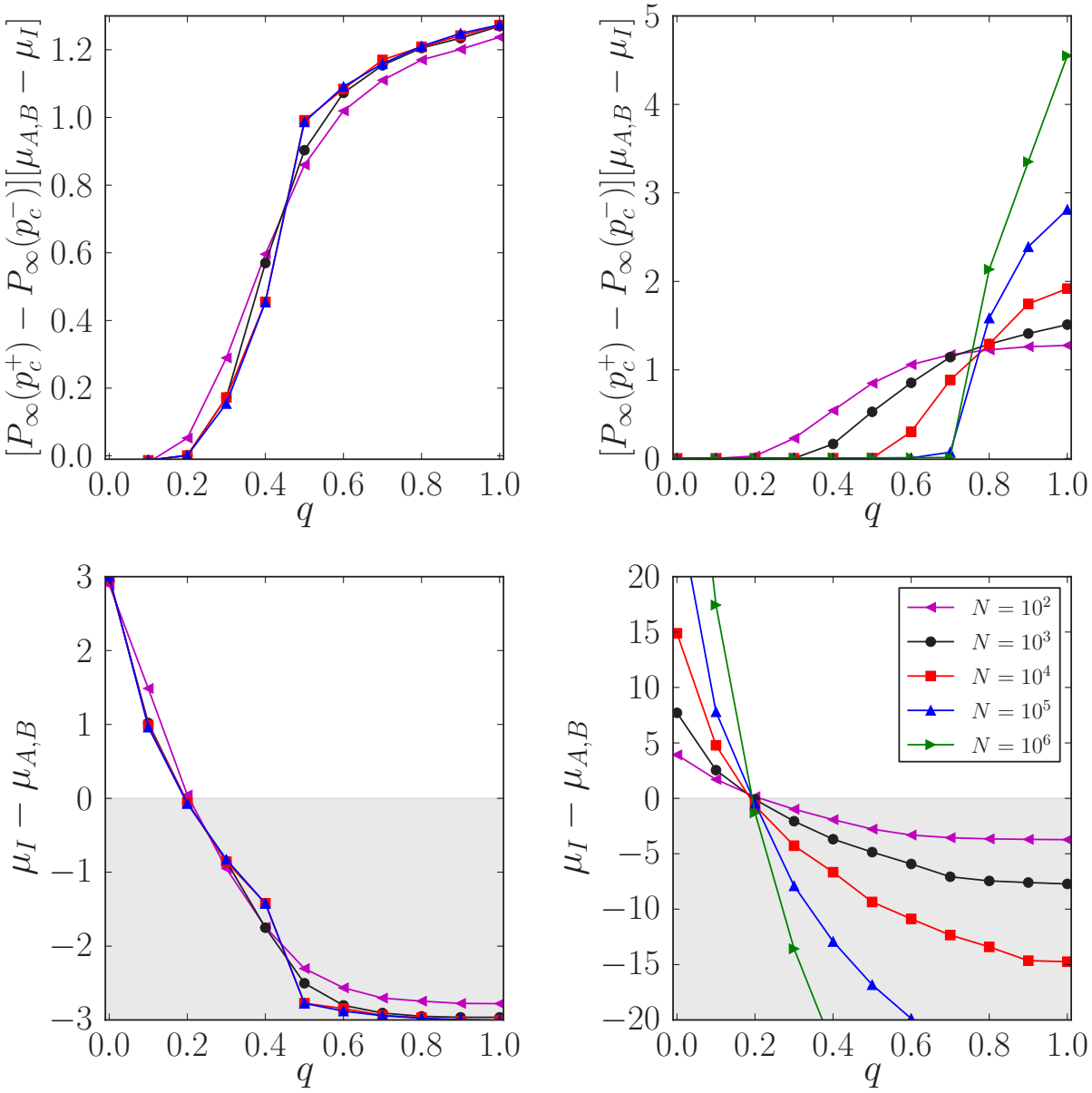


Figure S4: **a)** Height of the jump of the order parameter around the discontinuity point for two identical interdependent ER models, with average degree $\langle k \rangle = 3$, where the nodes of one the layers are relabeled with probability $q$. The jump in the order parameter does not always start off zero, so in this case we explicitly write the height of such a jump as the difference between the right and left-hand limits of the function $P_\infty$, i.e., $P_\infty(p_c^+) - P_\infty(p_c^+)$. Points represents average values of the height of such jumps, computed by solving numerically Eqs. (3) and (4) of the main text, over at least 10 realizations of the model. The height of the jump is multiplied by the factor $\mu_{A,B} - \mu_I$. $\mu_I$ is the largest eigenvalue of the non-backtracking matrix associated to the largest connected component of the intersection graph. $\mu_{A,B}$ is the smallest among the largest eigenvalues of the non-backtracking matrices associated to the largest connected component of the remainders. **b)** Same as in panel a but for two identical interdependent scale-free networks, with exponent $\gamma = 2.5$ and minimal degree $k_{min} = 3$, where the nodes of one the layers are relabeled with probability $q$. **c)** $\mu_I - \mu_{A,B}$ for the same networks as in panel a. The gray area denotes the region of values below zero. **d)** Same as in panel c, but the networks analyzed in panel b.

DOI: 10.1038/NPHYS3374

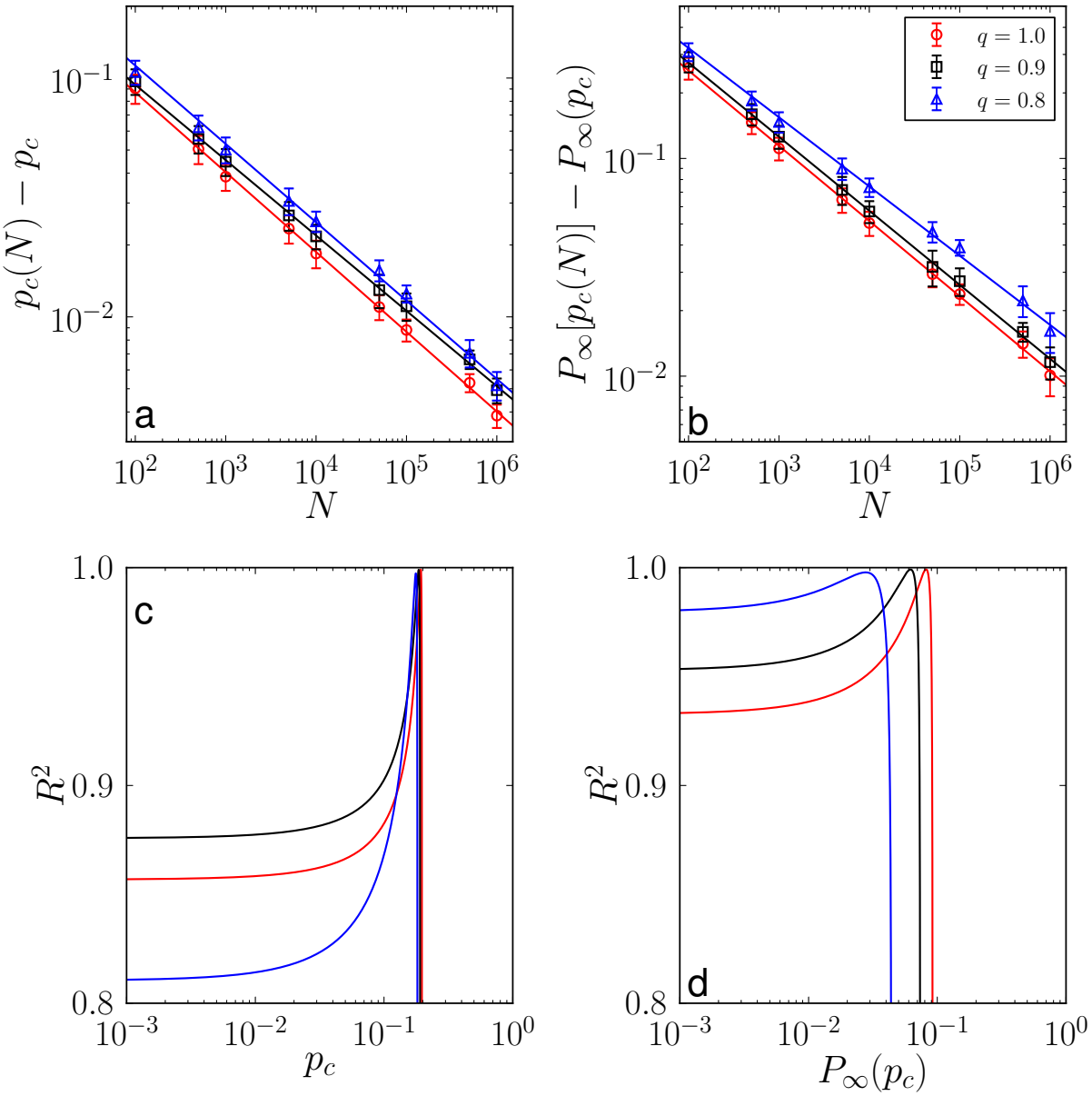


Figure S5: Finite-size analysis for the scale-free networks studied in Fig. S4. **a)** As the size of the network grows, the pseudo-critical threshold $p_c(N)$ gets closer to the critical threshold $p_c$ in a power-law fashion, i.e., $p_c(N) - p_c \sim N^{-\alpha}$ (solid lines). Points indicate average values obtained over at least 10 independent realizations of the network model, error bars quantify instead the standard deviation of the measures across different realizations. The best estimates of $p_c$ and $\alpha$, as well as the $R^2$ of the linear fit, are reported in Table S3. **b)** Same as in panel a but for the height of the jump of the percolation strength at pseudo-criticality. In this case, the solid line stand for the best estimate of the scaling $P_\infty[p_c(N)] - P_\infty(p_c) \sim N^{-\beta}$. The best estimates of $P_\infty(p_c)$ and $\beta$, and the $R^2$ value of the linear fit are tabulated in Table S3. **c)** Coefficient of variation $R^2$ for the linear fits in panel a. The value of $p_c$ corresponding to the peak of $R^2$ is assumed to be the best estimate of the asymptotic value of the percolation threshold. **d)** Same as in in panel c, but for the linear fits of panel b.

| $q$ | $p_c$ | $\alpha$ | $R^2$ | $P_\infty(p_c)$ | $\beta$ | $R^2$ |
|---|---|---|---|---|---|---|
| 1.0 | 0.19 | 0.33 | 1.00 | 0.08 | 0.35 | 1.00 |
| 0.9 | 0.19 | 0.32 | 1.00 | 0.06 | 0.34 | 1.00 |
| 0.8 | 0.18 | 0.33 | 1.00 | 0.03 | 0.32 | 1.00 |

Table S3: Values of the best estimates for the parameters of the power-law fits performed in in Fig. S5. For every value of the mixing probability $q$, we report the results obtained for the finite-size scaling of the percolation threshold, i.e., $p_c(N) = p_c + aN^{-\alpha}$, and those valid for the finite-size scaling of the height of the order parameter at criticality $P_\infty[p_c(N)] = P_\infty(p_c) + bN^{-\beta}$. Also, we show the respective values of the squared coefficients of variation of the two fits.

DOI: 10.1038/NPHYS3374

[1] B. Bollobás, C. Borgs, J. Chayes, O. Riordan, et al., The Annals of Probability **38**, 150 (2010).
[2] B. Karrer, M. E. J. Newman, and L. Zdeborová, Physical Review Letters **113**, 208702 (2014).
[3] K. E. Hamilton and L. P. Pryadko, Physical Review Letters **113**, 208701 (2014).
[4] S. N. Dorogovtsev, A. V. Goltsev, and J. F. Mendes, Reviews of Modern Physics **80**, 1275 (2008).
[5] S. Melnik, A. Hackett, M. A. Porter, P. J. Mucha, and J. P. Gleeson, Phys. Rev. E **83**, 036112 (2011).
[6] K.-i. Hashimoto, Automorphic forms and geometry of arithmetic varieties. pp. 211–280 (1989).
[7] F. Krzakala, C. Moore, E. Mossel, J. Neeman, A. Sly, L. Zdeborová, and P. Zhang, Proceedings of the National Academy of Sciences **110**, 20935 (2013).
[8] F. Radicchi, Physical Review E **91**, 010801 (2015).
[9] F. Radicchi and C. Castellano, submitted (2015).
[10] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, and S. Havlin, Nature **464**, 1025 (2010).
[11] J. Gao, S. V. Buldyrev, H. E. Stanley, and S. Havlin, Nature physics **8**, 40 (2012).
[12] M. Molloy and B. Reed, Random structures & algorithms **6**, 161 (1995).
[13] M. Catanzaro, M. Boguñá, and R. Pastor-Satorras, Physical Review E **71**, 027103 (2005).
[14] C. Stark, B.-J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers, Nucleic acids research **34**, D535 (2006).
[15] M. De Domenico, M. A. Porter, and A. Arenas, Journal of Complex Networks p. cnu038 (2014).
[16] G. Bianconi and S. N. Dorogovtsev, arXiv preprint arXiv:1411.4160 (2014).