Supplementary Information

Disentangling Direct from Indirect Relationships in Association Networks

Naijia Xiao, Aifen Zhou, Megan L. Kempher, Benjamin Zhou, Zhou Jason Shi, Mengting Yuan, Xue Guo, Linwei Wu, Daliang Ning, Joy Van Nostrand, Mary K. Firestone*, and Jizhong Zhou*

*Mary K. Firestone

Email: mkfstone@berkeley.edu

*Jizhong Zhou

Email: jzhou@ou.edu

This PDF file includes:

Supplementary text

Figures S1 to S16

Tables S1 to S10

SI References

Supplementary Information Text

A: Details for Mathematical Problems Associated with Several Previous Approaches

A.1: Characteristics and key issues associated with several previous approaches

The main purpose of direct relationship inference is to find the direct association matrix **S**, when the total association matrix **G** is given. Several methods have been proposed to solve this problem, including Network Deconvolution (1) (ND), Global Silencing (2) (GS), and SPIEC-EASI (3). The general approach is to find a relationship between **G** and **S** first, and then develop an algorithm to solve **S** when **G** is given. In ND, the indirect influence corresponds to indirect paths of all lengths, i.e.

$$\mathbf{G} = \mathbf{I} + \mathbf{S} + \mathbf{S}^2 + \mathbf{S}^3 + \dots. \tag{A1}$$

Then $\mathbf{S} = (\mathbf{G} - \mathbf{I})\mathbf{G}^{-1}$ is used to solve **S** from **G**. Then eigen-decomposition is applied to obtain $\mathbf{G}^{-1}$. In GS, the association between *i* and *j* is split into two parts: association between *i* and one of *j*'s neighbors *k* and association between *k* and *j*, i.e. the off-diagonal terms of the matrix product **SG**. Using some approximations, **S** is given in terms of **G** as:

$$\mathbf{S} = \left(\mathbf{G} + \operatorname{diag}\left\{\mathbf{G}\left(\mathbf{G} - \mathbf{I}\right)\right\}\right)\mathbf{G}^{-1}. \tag{A2}$$

In SPIEC-EASI, **S** is assumed to be $\mathbf{G}^{-1}$; then $\mathbf{G}^{-1}$ is solved using a minimization process with penalty terms, assuming that $\mathbf{G}^{-1}$ is sparse. Differences between Eqs. (A1) and (A2) and the equations presented in ND and GS are due to whether diagonal terms in **G** are included.

Compared to traditional approaches, Network Deconvolution (ND) (1), Global Silencing (GS) (2) and SPIEC-EASI (3) have certain advantages. First, conceptually, while ND considers the indirect influences as flows of direct influences along the edges of the true network and

55 expresses them as a sum of an infinite power series of the direct correlation matrix, GS treats
56 measured correlations as small perturbations and derives a formula that resembles Modular
57 Response Analysis (MRA) (4, 5), and SPIEC-EASI uses either neighborhood selection or sparse
58 inverse covariance selection to estimate the interaction network. ND, GS, and SPIEC-EASI are
59 all capable of considering indirect paths of arbitrary lengths. In contrast, previous methods (6)
60 study local patterns of dependencies to recognize potential indirect edges and can only consider
61 indirect paths of limited length (usually 2). Theoretically, ND, GS, and SPIEC-EASI provide
62 more general frameworks for estimating direct influences from observed total measurements, and
63 hence it should be more applicable to network inferences in various applications. Below we
64 introduce some basic concepts about direct and indirect relationships in an association and
65 present the characteristics and key issues of these approaches.
66
67 A major problem in constructing association networks is that the observable total association G_{ij}
68 between node pair *i* and *j* contains not only direct interactions between *i* and *j*, but also indirect
69 interactions through other intermediate nodes (7). Connecting *i* and *j* with those intermediate
70 nodes forms an indirect influence path. Each segment of the indirect path is a direct link. Indirect
71 paths can be of any length larger than one. The length of a path is also referred to as its order.
72 Two paths can overlap partly. Any indirect paths can be constructed from their direct links by
73 attaching links sequentially and parallelly. Here sequential paths mean two nodes indirectly
74 connected through an intermediate node (Fig. S12a). Parallel paths mean two nodes linked
75 through two different paths, directly or indirectly (Fig. S12b).
76
77 In summary, problems with these existing methods are:

a. Ill-conditioning. The total association matrix **G** is usually either singular or ill-conditioned. Any solution involving $\mathbf{G}^{-1}$ is therefore highly unreliable. ND, GS, and SPIEC-EASI all used $\mathbf{G}^{-1}$ in their formulations.
b. Self-looping. None of these methods can eliminate all spurious indirect paths containing self-loops in their formulation. This leads to overestimating the effects of indirect associations.
c. Interaction strength overflow. Entries of the resulting direct association matrix **S**, which theoretically should always lie in the natural range [0,1] of association data, overflow outside [0,1] in practice.

In the following, these problems are discussed further in detail.

A.2: Ill-conditioning

The association matrix **G**, whose (*i*,*j*)-th entry G_{ij} represents the association strength between the *i*-th and the *j*-th node in the network, is either singular or ill-conditioned (8). Here singularity means that its inverse $\mathbf{G}^{-1}$, the matrix that makes $\mathbf{G}\mathbf{G}^{-1} = \mathbf{G}^{-1}\mathbf{G} = \mathbf{I}$ does not exist. Ill-conditioning means that its inverse $\mathbf{G}^{-1}$ is highly unreliable. The singularity of a matrix can be detected by checking if its rank is smaller than its size, or if its eigenvalues contain zeros.

For example, we can prove that an association matrix obtained using Pearson's correlation is singular when the number of samples m is smaller than the number of nodes in the network n, i.e. $m < n$. The Pearson's correlation coefficients between i and j is given by

$$G_{ij} = \frac{\mathrm{cov}\left(\mathbf{x}_i,\, \mathbf{x}_j\right)}{\sigma_{\mathbf{x}_i}\sigma_{\mathbf{x}_j}} = \frac{\left(\mathbf{x}_i - \mu_{\mathbf{x}_i}\right)^T \left(\mathbf{x}_j - \mu_{\mathbf{x}_j}\right)}{\sqrt{\mathbf{x}_i^T\mathbf{x}_i - m\mu_{\mathbf{x}_i}^2}\sqrt{\mathbf{x}_j^T\mathbf{x}_j - m\mu_{\mathbf{x}_j}^2}}, \tag{A3}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are two vectors storing the abundance or activity information of the i-th and j-th nodes in the network, cov($\mathbf{x}_i$, $\mathbf{x}_j$) is the covariance between $\mathbf{x}_i$ and $\mathbf{x}_j$

$$\mathrm{cov}\left(\mathbf{x}_i,\, \mathbf{x}_j\right) = \left(\mathbf{x}_i - \mu_{\mathbf{x}_i}\right)^T \left(\mathbf{x}_j - \mu_{\mathbf{x}_j}\right), \tag{A4}$$

$\sigma_{\mathbf{x}i}$ and $\sigma_{\mathbf{x}j}$ are the standard deviation of $\mathbf{x}_i$ and $\mathbf{x}_j$

$$\sigma_{\mathbf{x}_i} = \sqrt{\mathbf{x}_i^T\mathbf{x}_i - m\mu_{\mathbf{x}_i}^2},\ \text{and}\ \sigma_{\mathbf{x}_j} = \sqrt{\mathbf{x}_j^T\mathbf{x}_j - m\mu_{\mathbf{x}_j}^2}, \tag{A5}$$

$\mu_{\mathbf{x}i}$ and $\mu_{\mathbf{x}j}$ are the mean

$$\mu_{\mathbf{x}_i} = \frac{1}{m}\sum_{k=1}^{m} x_{ik},\ \text{and}\ \mu_{\mathbf{x}_j} = \frac{1}{m}\sum_{k=1}^{m} x_{jk}, \tag{A6}$$

and m is the number of samples (the length of $\mathbf{x}_i$ and $\mathbf{x}_j$). If column vectors $\left(\mathbf{x}_i - \mu_{\mathbf{x}_i}\right)/\sigma_{\mathbf{x}_i}$ are put together and named $\overline{\mathbf{X}}$, $\mathbf{G}$ can be rewritten as

$$\mathbf{G} = \overline{\mathbf{X}}^T\overline{\mathbf{X}}. \tag{A7}$$

For a network with n nodes reconstructed from m samples ($m < n$), the dimension of $\overline{\mathbf{X}}$ is $m \times n$, and the dimension of $\mathbf{G}$ is $n \times n$. Because $\mathbf{G}$ is a product of $\overline{\mathbf{X}}^T$ and $\overline{\mathbf{X}}$, the rank of $\mathbf{G}$ equals the rank of $\overline{\mathbf{X}}$ and is at most m. Thus, the rank of $\mathbf{G}$ is smaller than the dimension of $\mathbf{G}$ ($m < n$), rendering $\mathbf{G}$ singular.

For association measures other than Pearson's correlation, the corresponding association matrix **G** is ill-conditioned. The conditioning number n_{cond} of a matrix quantifies whether a matrix is ill-conditioned. n_{cond} is the ratio of the largest eigenvalue $|\sigma_{max}|$ and the smallest eigenvalue $|\sigma_{min}|$ of a matrix, that is, $= |\sigma_{max}|/|\sigma_{min}|$; n_{cond} describes the reliability of $\mathbf{G}^{-1}$ and can be interpreted as the maximal ratio possible between the error in the inverse $\mathbf{G}^{-1}$ and the error in **G** (9). Fig. S1a showed that, for a fixed number of samples (m = 20), as the size of the network increases, the conditioning number of **G** increases significantly, from [4.6, 27.32] (n = 5) to [1.78×10^5, 1.52×10^7] (n = 1,000), with an average increase of 2.70×10^5 from n = 5 to n = 1,000. Association measures include absolute value of Pearson correlation, absolute value of Spearman correlation, Kendall rank correlation, Bray-Curtis dissimilarity, distance correlation, and maximal information coefficients.

The results presented above indicated that an association matrix **G** is either singular or ill-conditioned, and its inverse $\mathbf{G}^{-1}$ is either non-existent or highly unreliable. The ill-conditioning of **G** is caused by the underdetermined nature of the network reconstruction problem, which is one of the obstacles that every network analysis method must consider (10, 11). Underdetermination means that the amount of information available is not enough to determine all the unknown variables (12), and it is usually because it is extremely difficult to survey enough replicate samples. For example, consider a network containing n nodes, and suppose m samples are collected about the abundance or activities of those nodes. The total number of pieces of available information is mn. In contrast, to reconstruct the complete pair-wise relationships between those n entities, the total number of unknown variables is at least $n(n-1)/2$ (when symmetry is assumed; in the case of asymmetry, the number is doubled). In practice, the number

of samples is far fewer than the number of entities in the network, that is, $m << n$. For example, in microbial community studies, the number of samples is usually in the magnitude of tens or hundreds, while the number of OTUs (operational taxonomic units) under consideration can vary from hundreds to thousands or even millions. Consequently, $mn << n(n-1)/2$ and the problem is severely underdetermined.

The consequences of underdetermination can be illustrated using a linear algebra problem. Let **A** be a given $m \times n$ matrix, **x** be an unknown $n \times 1$ column vector, and **b** be a given $m \times 1$ column vector that represents available information. Consider the following linear system

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \tag{A8}$$

When $m<n$, there is not enough information available to uniquely determines all the unknown variables, and hence the system is underdetermined. More specifically, because $m<n$, we can always find $\mathbf{x}_1$ satisfying $\mathbf{A}\mathbf{x}_1 = \mathbf{0}$. Given one solution $\mathbf{A}\mathbf{x}_0 = \mathbf{b}$ to Eq. (A8), we can construct an infinite number of solutions by letting $\mathbf{x} = \mathbf{x}_0 + a\mathbf{x}_1$, where a is an arbitrary real number. To choose the most plausible solution among those solutions, additional information must be used. For example, if we are interested in the solution with the least norm to Eq. (A8) when $m<n$, we can use the Moore–Penrose right pseudoinverse, which is a generalization of the inverse matrix:

$$\mathbf{x} = \mathbf{b}\mathbf{A}^T \left(\mathbf{A}\mathbf{A}^T\right)^{-1}. \tag{A9}$$

Additional information can be applied in the form of penalty terms in the optimization, also known as regularizations.

To address the singularity or ill-conditioning problem of **G**, usually additional assumptions are made regarding the properties of **G** and these assumptions are used to obtain an empirical

approximation of $\mathbf{G}^{-1}$. In ND (1), eigen-decomposition is used to reach a pseudo-inverse of **G**. In GS (2), **G** is modified according to its confidence level using a bootstrap randomization before direct inversion. In SPIEC-EASI, an optimization approach is adopted, and a penalty term is introduced to ensure the sparsity of $\mathbf{G}^{-1}$ (3, 13). All these methods turn to generic numerical analysis techniques to invert the association matrix **G**, without utilizing the intrinsic network structure provided in **G**.

A.3: Self-looping

The second issue with existing methods is overestimating indirect influence due to spurious indirect paths containing self-loops (1). Self-loops are spurious paths that start and end at the same node. Allowing indirect paths to include self-loops will result an infinite number of indirect paths that contain one or more self-loops. For example, consider a simple network in Fig. S1b. A valid indirect path *B*-*C*-*D* (green dotted lines) connects node *B* and *D* through node *C*. However, indirect paths such as *B*-*A*-*B*-*D* (red dotted lines, containing a self-loop *B*-*A*-*B*) and *B*-*C*-*D*-*B*-*D* (purple dotted lines, containing self-loops *B*-*C*-*D*-*B* and *D*-*B*-*D*) are spurious and should be excluded in the calculation. If such paths are allowed, we can construct additional paths such as *B*-(*A*-*B*)$_n$-*D* and *B*-(*C*-*D*-*B*)$_n$-*D*, where the paths in the bracket are repeated *n* times. These paths are not useful and must be excluded in the calculation.

ND proposed to eliminate spurious indirect paths by deleting the diagonal terms in **S**, $\mathbf{S}^2$, $\mathbf{S}^3$, etc., where **S** is the direct association matrix (1). This approach can only eliminate spurious indirect paths that have the same starting and ending nodes, i.e. paths like *A*-*B*-*A*. For spurious paths

containing self-loops in the middle of the path, this approach does not work. For example, consider the spurious indirect path *B*-*A*-*B*-*D* in Fig. S1b, a self-loop *B*-*A*-*B* occurs in the middle of the path, and the starting node *B* and the ending node *D* are distinct. The incorrect association strength $S_{BA}S_{AB}S_{BD}$ will still be present in $\mathbf{S}^3$. This will result in an overestimation of the indirect association strength between *B* and *D*.

GS (2) follows a different approach and uses the matrix product **SG** to calculate the indirect associations. This approach also cannot eliminate all the spurious indirect paths. Consider the indirect influence between node *B* and *D* in Fig. S1b. The only valid indirect path is *B*-*C*-*D*. However, the corresponding entry in **SG** is $S_{BA}G_{AD} + S_{BB}G_{BD} + S_{BC}G_{CD} + S_{BD}G_{DD}$. $S_{BB}G_{BD} = S_{BD}G_{DD} = 0$, because $S_{BB} = G_{DD} = 0$; $S_{BA}G_{AD}$ includes two spurious indirect paths *B*-*A*-*B*-*D* and *B*-*A*-*B*-*C*-*D*, because G_{AD} contains two indirect paths *A*-*B*-*D* and *A*-*B*-*C*-*D*; $S_{BC}G_{CD}$ includes one valid indirect path *B*-*C*-*D* and one spurious indirect path *B*-*C*-*B*-*D*, because G_{CD} contains one direct path *C*-*D* and one indirect path *C*-*B*-*D*. Therefore, **SG** includes one valid indirect path (*B*-*C*-*D*) and three spurious indirect paths (*B*-*A*-*B*-*D*, *B*-*A*-*B*-*C*-*D*, and *B*-*C*-*B*-*D*). The common characteristic of these spurious paths is that the first and the third node in the path are identical. Therefore, GS also cannot eliminate the self-looping problem completely and eventually overestimates the indirect influence.

A.4: Interaction strength overflow

The third issue with the existing methods is associated with the rationale behind the formulation. ND postulates that all pair-wise indirect influences due to paths of length n ($n > 1$) can be

represented by the power $\mathbf{S}^n$, where **S** is the direct association matrix. The total observed matrix **G** contains both direct and indirect effects, and can be computed by summing **S** and all its powers:

$$\mathbf{G} = \mathbf{I} + \mathbf{S} + \mathbf{S}^2 + \mathbf{S}^3 + \ldots, \tag{A10}$$

which corresponds to all direct and indirect paths of all lengths. GS follows a different approach. Its derivation in the main text is based on treating the edge strength as small perturbations; however, the edge strength is later inconsistently calculated through correlation or other association measures. In its supplemental material, it was postulated that the indirect effects can be represented by the product **SG**, and the total observed matrix **G** is the sum of the direct matrix **S** and the indirect effects **SG**:

$$\mathbf{G} = \mathbf{S} + \mathbf{SG}. \tag{A11}$$

Both Eqs. (A10) and (A11) implicitly assume that if there are two or more paths connecting two nodes, the total association strength is the sum of association strengths of the individual paths using ordinary addition +. This assumption of the combinatorial rule for association data is fundamentally flawed, because it is incompatible the nature of association data; It results in the association strength overflowing outside the natural range [0,1]. Consider a simple network in Fig. S1c and use the rule presented in ND and GS. The indirect association strength of *B*-*C*-*D* is 0.6×0.7 = 0.42. The sum of the direct influence *B*-*D* and indirect influence *B*-*C*-*D* is 0.8+0.42 = 1.22 > 1. ND suggests that linearly rescaling of the results back to [0,1] can resolve this issue. This simplistic approach conceals the real problem here: it is not correct to use + to add individual direct/indirect association strengths together to obtain the total association strength. In the current example, the issue is that G_{BD} should be not equal to $S_{BD} + S_{BC}S_{CD}$. We need more systematic treatment to solve this association strength overflow problem.

B: Detailed mathematical framework for iDIRECT

In this section, we will address the problems with the existing methods outlined in Appendix A in a reverse order here. We will first describe iDIRECT on how to avoid interaction strength overflow (Appendix B.1), followed by a new strategy for eliminating self-looping (Appendix B.2). We will then introduce nonlinear solvers to minimize influences from underdetermination (Appendix B.3).

B.1: Strategies to address interaction strength overflow - copula-based additions

To address the problem of interaction strength overflow, we first introduce the concepts of sequential and parallel paths. A new operator $\oplus$ based on copulas is developed for parallel paths, replacing the ordinary addition + used in ND and GS. The proposed $\oplus$ is designed to give results that is consistent with common sense and guarantee to lie in the natural range [0,1] of association data. Then we describe an assembly strategy that uses sequential and parallel paths to calculate the strength of any indirect paths in a network. In this way, we solve the interaction strength overflow problem completely.

An indirect path consists of more than one segments (Appendix A.1), and each segment is a direct link. For sequential paths, two nodes are connected via an intermediate node. Consider an example in Fig. S12a, an indirect path *i*-*k*-*j* contains two segments, *i*-*k* with association strength *u*, and *k*-*j* with association strength *v*. The association strength of the indirect path *i*-*k*-*j* is intuitively assumed to be *uv*, or $u \otimes v$ as a generic notation, that is

$$u \otimes v = uv. \tag{B1}$$

For parallel paths, two nodes are connected via two different paths. For example, consider the four nodes in Fig. S12b. There are two paths connecting nodes *i* and *j*: one is *i*-k_1-*j* with association strength *u*, and the other is *i*-k_2-*j* with association strength *v*. The total association strength between *i* and *j* is termed as $u \oplus v$. An intuitive choice of $u \oplus v$ is $u + v$, which could result in association strength overflowing outside [0,1]. To avoid this, less intuitive but more desirable choices can be found, such as the one below,

$$u \oplus v = \frac{u + v - 2uv}{1 - uv}, \tag{B2}$$

which are based on Archimedean copulas. The realization of $u \oplus v$ in Eq. (B2) yields results that are guaranteed to lie within [0,1]. In the following subsections, we will discuss the following issues in detail: (i) sequential paths and the operator $\otimes$ (Appendix B.1.1), (ii) parallel paths and the operator $\oplus$ (Appendix B.1.2), and (iii) assembly strategies to use sequential and parallel paths (Appendix B.1.3).

B.1.1: Sequential paths

Sequential paths are one of the basic building blocks when studying indirect associations in networks. They occur when two nodes are indirectly linked via a third node. Consider a network of nodes *i*, *j*, and *k* (Fig. S12a). *i*-*k* and *k*-*j* are directly linked, with respective association strengths *u* and *v*. The association strength between *i* and *j* is determined by *u* and *v*, as well as the rule that relates them (a binary operation $\otimes$, in mathematical terms). Ordinary multiplication satisfies the requirements for $u \otimes v$, yet there exist other less intuitive choices.

The basic requirements for $u \otimes v$ are listed below:

1. *Lower bound*: $u \otimes 0 = 0 \otimes v = 0$, meaning if either of the two edges is not associated at all, the indirect association strength should be zero;
2. *Upper bound*: $u \otimes 1 = u$ and $1 \otimes v = v$, meaning if either of the edges is fully associated, the indirect association strength should equal the strength of the remaining edge;
3. *Monotonicity*: $u_1 \otimes v \leq u_2 \otimes v$, for all $u_1 \leq u_2$, and $u \otimes v_1 \leq u \otimes v_2$, for all $v_1 \leq v_2$, meaning that if either of the two associations is stronger, the resulting indirect association is also stronger, and *vice versa*.

The ordinary multiplication, $u \times v = uv$, satisfies the above three requirements and is the most intuitive choice. However, it is not the only choice. The minimum function, $\min\{u,v\}$ also satisfies those requirements. In fact, there are families of functions based on copulas (see Table S8) that satisfy those requirements. In short, a copula is a bivariate function that satisfies very similar conditions to the requirements for $u \otimes v$ (see Appendix C for details). For example, consider the Clayton family, letting $\theta = 0$ gives us $u \otimes v = uv$, and letting $\theta = +\infty$ yields $u \otimes v = \min\{u,v\}$ (Table S8). In general, for a given copula $C(u,v)$, we can construct a valid binary operator $\otimes$ for sequential paths such that $u \otimes v = C(u,v)$.

The operator $\otimes$ can be used repeatedly to calculate the indirect association strength between nodes that are connected via multiple intermediate nodes. For instance, nodes *i* and *j* in Fig. S14 are connected through *i*-*k*-*l*-*j*. Let the association strength of direct links *i*-*k*, *k*-*l*, and *l*-*j* be *u*, *v*,

and w, respectively. The indirect association strength between i and j is obtained by repeatedly applying the rule for sequential paths, that is, $u \otimes v \otimes w$.

Besides the three basic requirements discussed above, $u \otimes v$ should be commutative and associative. Commutativity requires $u \otimes v = v \otimes u$. Take the example in Fig. S12a. $u \otimes v$ means the association strength from i to j via k. $v \otimes u$ means the association strength from j to i via k. Therefore, commutativity means the association strength from i to j equals the association strength from j to i, that is $u \otimes v = v \otimes u$. Associativity requires $(u \otimes v) \otimes w = u \otimes (v \otimes w)$. Take the example in Fig. S14. $(u \otimes v) \otimes w$ means we first calculate the association from i to l via k. Then we use it to calculate the association strength from i to j via l. $u \otimes (v \otimes w)$ means we first calculate the association from k to j via l; then we use it to calculate the association strength from i to j via k. Therefore, associativity means these two approaches are equivalent, and the result is just the association strength between i and j via k and l. Operators satisfying commutativity and associativity are closely related to Archimedean copulas, which are discussed in detail in Appendix C.2. The intuitive realization $u \otimes v = uv$ satisfies both commutativity and associativity.

B.1.2: Parallel paths

Parallel paths are the other basic building blocks in indirect association calculation. They occur when two nodes are linked via two different paths. It can be extended to situations when more than two paths connect the two nodes of interest. The combined association strength is determined by the association strength of those paths and the rule that relates them. In the

following discussion, several plausible choices are proposed, and one of them is chosen as the default choice in iDIRECT.

Consider a network of node i, j, k_1, and k_2, in Fig. S12b. Node i and node j are indirectly linked via intermediate nodes k_1 and k_2. Let the association strengths of direct links i-k_1, k_1-j, i- k_2, and k_2-j be u_1, u_2, v_1, and v_2, respectively. Let $u = u_1 \otimes u_2$ be the association strength due to path i-k_1-j and $v = v_1 \otimes v_2$ be the indirect association due to path i-k_2-j. We are interested in the combined association strengths due to i-k_1-j and i-k_2-j, and the result is denoted as a binary operation $u \oplus v$. The binary operation $u \oplus v$ operates on the strengths of indirect paths that connects the same node pairs, u and v, and returns the total strength. Node k_1 and node k_2 in Fig. S12b are introduced merely to distinguish one path from the other. In practice, these paths can contain more than one intermediate node or none. In the former case, when the path has multiple intermediate nodes, the indirect association strength of the path is computed following the approach outlined in Appendix B.1.1; in the latter case, when the two nodes are directly linked, we just use its direct association strength.

The basic requirements for $u \oplus v$ are listed below:

1. *Lower bound*: $u \oplus 0 = u$ and $0 \oplus v = v$, meaning if either of the two paths is disconnected (equals zero), the total association strength should equal the strength of the remaining path;
2. *Upper bound*: $u \oplus 1 = 1 \oplus v = 1$, meaning if either of the path is fully associated, the total association strength is one. This requirement has not been considered before. Ordinary

addition does not satisfy this requirement. By satisfying it, iDIRECT differs significantly from all previous methods;

3. *Monotonicity*: $u_1 \oplus v \le u_2 \oplus v$, for all $u_1 \le u_2$ and $u \oplus v_1 \le u \oplus v_2$, for all $v_1 \le v_2$, meaning that if either of the two associations is stronger, the resulting total association is also stronger, and *vice versa*.

There are numerous functions that satisfy those basic requirements. In fact, given an arbitrary copula $C(u,v)$ (see Appendix C.1 for detail), we can construct a valid operator $\oplus$ such that $u \oplus v = 1 - C(1-u, 1-v)$. Here, we focus on three realizations that have simple explicit mathematical expressions. The first realization is

$$u \oplus v = u + v - uv. \tag{B3}$$

The second realization is the maximum function:

$$u \oplus v = \max\{u, v\}. \tag{B4}$$

The third realization is slightly more complicated than the previous two:

$$u \oplus v = \frac{u + v - 2uv}{1 - uv}. \tag{B5}$$

The realizations listed in Eqs. (B3-B5) satisfy commutativity and associativity, making the operations independent of the order that they are performed. As a side note, the independent copula $\Pi(u,v) = uv$ corresponds to $u \oplus v = u + v - uv$ in Eq. (B3); the upper Fréchet-Hoeffding bound $W(u,v)$ corresponds to $u \oplus v = \max\{u,v\}$ in Eq. (B4).

To compare the three different realizations of $\oplus$ and the ordinary addition +, Table S9 lists results of $u \oplus v$ (u, v = 0.1, 0.5, 0.9) when different realizations of $\oplus$ are used. Apparently, ($u+v-$

$2uv)/(1–uv)$ acts like $u+v–uv$ when both u and v are small and like $\max\{u,v\}$ when either u or v is large. We see that the three realizations of $\oplus$ in Eqs. (B3-B5) always produce results within the natural range [0,1] of association data; for ordinary addition $u+v$, the results can exceed 1.

A comparison of the contour plots of the different realizations of $u\oplus v$ is shown in Fig. S15, which also includes that of $u\otimes v = uv$. We noticed that $u+v–uv$ and $(u+v–2uv)/(1–uv)$ yield results close to $u+v$ when u and v are small, as the contour lines almost have a constant slope of -1 near $u = v = 0$ (red triangular box). This is intuitive: multiple paths with similarly weak associations result in a slightly stronger association. When one of the arguments, say u, is close to 1, and the other is small, $\max\{u,v\}$ and $(u+v–2uv)/(1–uv)$ yield result close to u, as indicated by the almost vertical contour lines near $u = 1$ (blue rectangular box). This is also desirable: a strong path should dictate the total association strength when other weaker paths exist. Finally, it is noteworthy that uv and $u+v–uv$ are mirrored by the straight line $u+v = 1$. This is because uv and $u+v–uv$ are both constructed from the same copula $C(u,v) = uv$ (see Appendix C.2 for more details).

Therefore, in this paper, we use $u\oplus v = (u+v–2uv)/(1–uv)$ because it has both desired qualities as discussed above: (i) multiple paths with similarly weak associations result in only a slightly stronger association; (ii) a strong path dictates the total association strength, even if there exist other weaker paths; and (iii) it is both commutative and associative, meaning the results do not depend on the operand order (commutative, $u\oplus v = v\oplus u$) or the operation order (associative, $(u\oplus v)\oplus w = u\oplus(v\oplus w)$).

B.1.3: Assembly strategies

The introduction of sequential and parallel paths enables us to compute the indirect association strength between two arbitrary nodes in a general network. To do so, the network needs to be decomposed into sequential paths and parallel paths. There are two available assembly strategies: the all-path sum (APS) and the two-step sum (TSP). These two strategies do not necessarily yield the same results, and their equivalence is closely related to the validity of the distributive law of the chosen binary operators $\otimes$ and $\oplus$.

Consider an illustrative network of 4 nodes, *A*, *B*, *C*, and *D* in Fig. S16a. *A*-*B*, *B*-*C*, *C*-*D*, and *B*-*D* are directly linked, and their respective association strengths are *u*, *v*, and *w* (blue solid lines). We are interested in the indirect association between node *A* and *C* (red dashed line). Two distinctive paths connecting *A* and *C* can be identified: *A*-*B*-*C* (green dotted lines) and *A*-*B*-*D*-*C* (purple dotted lines).

There are two strategies to calculate the association strength between *A* and *C*:

1. *All-path sum* (*APS*): the association strength of each path is calculated, and then the sum is computed, that is, $(u \otimes v) \oplus (u \otimes w)$. The scenario is visualized in Fig. S16b, where the path *A*-*B*-*C* is highlighted in green dotted lines, and the path *A*-*B*-*D*-*C* is highlighted in purple dotted lines;
2. *Two-step product* (*TSP*): the paths are divided into two parts: part 1 is *A*-*B* (shared by both) and part 2 is *B*-*C* and *B*-*D*-*C* (different for each path). Then the product of the two

parts is computed, that is, $u\otimes(v\oplus w)$. The scenario is visualized in Fig. S16c, where part 1 is highlighted in green dotted lines, and part 2 is highlighted in purple dotted lines.

Ideally these two strategies should be equivalent, which requires $\otimes$ to be distributive over $\oplus$. The only operator $\oplus$ that satisfies this condition, however, is $u\oplus v = \max\{u,v\}$. The following is a concise proof. Let u, v, and w be three arbitrary numbers, and $\otimes$ be distributive over $\oplus$. Per distributivity,

$$u \otimes (v \oplus w) = (u \otimes v) \oplus (u \otimes w)\,. \tag{B6}$$

Without loss of generality, let v be 1, and Eq. (B6) becomes

$$u\otimes(v\oplus 1) = (u\otimes v)\oplus(u\otimes 1)\,, \;\; \Rightarrow \;\; u\otimes 1 = (u\otimes v)\oplus(u\otimes 1)\,, \tag{B7}$$

where $v\oplus 1 = 1$ (upper bound property of $\oplus$) has been used. Now let $x = u\otimes 1$ and $y = u\otimes v$, then $x \geq y$ ($1 \geq v$ and monotonicity of $\otimes$). Eq. (B7) becomes

$$x = y \oplus x, \quad \Rightarrow \quad x \oplus y = x. \tag{B8}$$

In other words, the operator $\oplus$ returns the larger of the two inputs, which is the definition of the maximum function. Therefore, the maximum function $u\oplus v = \max\{u,v\}$ is the only one that satisfies the distributive law.

Any choice for $\oplus$ other than the maximum function will result in a violation of the distributive law of $\otimes$ over $\oplus$. However, the maximum function does not have an inverse operator. This means that given a function value $w = \max\{u,v\}$ and one of the argument u, it is not always possible to recover the other argument v: v can be any value in $[0,u]$ if $u = w$. In contrast, if we choose $u\oplus v = (u+v-2uv)/(1-uv)$, an inverse operator $\ominus$ can be defined as:

$$u \ominus v = \frac{u - v}{1 - 2v + uv}. \tag{B9}$$

And $(u \oplus v) \ominus v = (u \ominus v) \oplus v = u$. Having an inverse operator $\ominus$ makes the development of the nonlinear solvers detailed in Appendix B.3 possible and is a highly desirable feature. Discussion in Appendix B.1.3.1 describes the extent of the impact of violating the distributive law and shows that the TSP strategy can minimize this impact. Therefore, we will use the TSP strategy in developing iDIRECT.

B.1.3.1: Comparison of the two assembly strategies

To compare the two assembly strategies, we measure the deviation of the chosen binary operators $\otimes$ and $\oplus$ from satisfying the distributive law by the difference

$$\Delta = (u \otimes v) \oplus (u \otimes w) - u \otimes (v \oplus w). \tag{B10}$$

Table S10 compares values of Δ using three different combinations of u, v, and w. Consider the difference Δ when $u \otimes v = uv$ and $u \oplus v = (u+v-2uv)/(1-uv)$. The minimal difference $\Delta_{min} = 0$, and the maximal difference $\Delta_{max} = 3-2\sqrt{2}$, which occurs at $u = \sqrt{2}-1$, $v = 1$, $w = 1$. The difference Δ is always positive, meaning the APS strategy always gives results larger than those from the TSP strategy.

A closer investigation into the difference between these two strategies is provided in Fig. S16d. Consider the same 4-node network in Fig. S16a, and consider the indirect association strength G_{AC} between node *A* and *C*. Let the association strength be $u = 0.5$ for link *A*-*B*, and $v = w = 1$ for link *B*-*C* and *B*-*C*-*D*, respectively. Thus, node *B*, *C*, and *D* are assumed to have perfect

association, and we would expect $G_{AC} = G_{AB} = u$. However, the APS strategy gives $G_{AC} = u \oplus u$ (green box), while the TSP strategy gives $G_{AC} = u$ (purple box). Unless $\oplus$ is the maximal function, $u \oplus u > u$. Therefore, the TSP strategy is preferred because it can capture what is intuitively expected, that is, $G_{AC} = u$.

B.2: Strategies to address the self-looping problem - transitivity matrix

To address the self-looping problem, iDIRECT introduces a transitivity matrix; its (i,k,j)-th component, $T_{i,kj}$, represents the association strength between node k and j, excluding paths passing node i. To demonstrate how the transitivity matrix eliminates all spurious self-looping paths, consider the indirect association between two nodes i and j through one of i's neighbors k (Fig. S12c). Using the TSP strategy, its indirect association strength is $S_{ik}T_{i,kj}$, where S_{ik} is the direct association strength between i and k and the first step in TSP. Here, we use $T_{i,kj}$ to represent the association strength between k and j for the second step in TSP instead of G_{kj}, because G_{kj} includes the influences of indirect paths passing i, while $T_{i,kj}$ explicitly excludes those paths in its definition. Consequently, $S_{ik}G_{kj}$ the includes influences of spurious self-looping paths such as i-k-...-i-…-j, where self-loops in the form of i-k- ... -i occurs; in contrast, $S_{ik}T_{i,kj}$ eliminates the influences of those spurious paths completely. In the following subsections, we will discuss (i) calculation of the transitivity matrix (Appendix B.2.1) and (ii) a relationship between direct association **S** (collection of all direct association strengths S_{ij}) and total association **G** (collection of all total association strength G_{ij}) (Appendix B.2.2).

B.2.1: Calculation of the transitivity matrix

To calculate the transitivity matrix $T_{i,kj}$, one can directly express $T_{i,kj}$ from the direct association matrix **S** per its definition – association of indirect paths connecting node *k* and *j* without passing node *i*. However, this approach requires listing all the paths connecting all node pairs at all lengths. Its computational complexity is well beyond the capacity of current computers. Instead, we use an indirect approach. Consider three nodes *i*, *j*, and *k* in Fig. S12d. The green dashed lines represent $T_{i,kj}$, $T_{j,ki}$, and $T_{k,ij}$, that is, associations between two nodes excluding paths passing the remaining third node. The total association G_{kj} between *k* and *j* consists of indirect paths between *k* and *j* not passing *i*, whose association strength is $T_{i,kj}$ (using the definition of $T_{i,kj}$), and indirect paths between *k* and *j* passing *i*, whose association strength is $T_{j,ki}T_{k.ij}$ (dividing the paths into two steps, *k*-*i* and *i*-*j*, and using the TSP strategy). The sum of these two terms (using the binary operator $\oplus$ for parallel paths) is the total association G_{kj}:

$$G_{kj} = T_{i,kj} \oplus (T_{j,ki}T_{k,ij})\,. \tag{B11}$$

In the same spirit, we can obtain two other equations about G_{ki} and G_{ij}. Combining these three equations enables us to use the following three nonlinear equations to solve for the three unknown variables $T_{i,kj}$, $T_{j,ki}$, and $T_{k,ij}$

$$\begin{cases} G_{kj} = T_{i,kj} \oplus (T_{j,ki}T_{k,ij})\,; \\ G_{ki} = T_{j,ki} \oplus (T_{k,ij}T_{i,kj})\,; \\ G_{ij} = T_{k,ij} \oplus (T_{i,kj}T_{j,ki})\,, \end{cases} \tag{B12}$$

where the symmetry of the transitivity matrix is used, that is, $T_{i,kj} = T_{i,jk}$, $T_{j,ki} = T_{j,ki}$, and $T_{k,ij} = T_{k,ji}$. We can iterate over all possible combinations of *i*, *j*, and *k* to obtain all entries of the transitivity matrix. Specifically, for each node *i*, we need to iterate *j* and *k* over all *i*'s neighbors. The total number of entries to calculation is $n\bar{d}(\bar{d}-1)/2$, where *n* is the number of nodes, and $\bar{d}$ is the average connectivity.

B.2.2: Relationship between direct and total associations

Combining the results above, the total association strength G_{ij} between any two nodes *i* and *j* in the network consists of the direct association strength S_{ij} between *i* and *j* and the indirect association strength. The indirect association strength includes many parallel paths, each of which starts from *i*, ends at *j*, and passes one of *i*'s neighbors k_2, k_3, …, k_d (intermediate nodes, Fig. S12c). Using the operator $\oplus$ for parallel paths, the total association

$$G_{ij} = S_{ij} \oplus S_{ik_2}T_{i,k_2j} \oplus S_{ik_3}T_{i,k_3j} \oplus \ldots \oplus S_{ik_d}T_{i,k_dj}. \tag{B13}$$

The intermediate nodes k_2, k_3, …, k_d are directly linked to the starting node *i*, with association strengths S_{ij}, S_{ik2}, S_{ik3}, …, S_{ikd}, respectively. They are indirectly linked to the ending node *j*, with association strengths $T_{i,k2j}$, $T_{i,k3j}$, …, $T_{i,kdj}$, respectively. k_1 is reserved so that k_1=*j*. We can iterate *j* over k_2, k_3, …, k_d to get other sets of equations and express them in a matrix form:

$$\begin{bmatrix} G_{ik_1} \\ G_{ik_2} \\ \vdots \\ G_{ik_d} \end{bmatrix} = \begin{bmatrix} S_{ik_1} \\ S_{ik_2} \\ \vdots \\ S_{ik_d} \end{bmatrix} \oplus \left(\begin{bmatrix} 1 & T_{i,k_1k_2} & \cdots & T_{i,k_1k_d} \\ T_{i,k_2k_1} & 1 & \cdots & T_{i,k_2k_d} \\ \vdots & \vdots & \ddots & \vdots \\ T_{i,k_dk_1} & T_{i,k_dk_2} & \cdots & 1 \end{bmatrix} \otimes \begin{bmatrix} S_{ik_1} \\ S_{ik_2} \\ \vdots \\ S_{ik_d} \end{bmatrix} \right) \tag{B14}$$

$$\Rightarrow \quad \mathbf{G}_i = \mathbf{S}_i \oplus (\mathbf{T}_i \otimes \mathbf{S}_i)\,,$$

where $\mathbf{G}_i$ and $\mathbf{S}_i$ are collections of G_{ik} and S_{ik} with *i* being fixed; $\mathbf{T}_i$ is a collection of $T_{i,kj}$ with *i* being fixed. Then we can iterate *i* over all nodes in the network. The total number of equations to solve is $n\bar{d}$, where *n* is the number of nodes, and $\bar{d}$ is the average connectivity. Eq. (B14) is the new relationship between total and direct association strengths **G** and **S** that we discovered, which forms the foundation of our formulation. Eq. (B14) superficially resembles GS formulation, but there are two important modifications: (i) replacing the normal addition "+" with a new copula-based operator $\oplus$, which guarantees the result to lie in the natural range [0,1]

of association data; and (ii) replacing the total association strength G_{kj} with a transitivity matrix $T_{i,kj}$, completely eliminating the influence of spurious indirect paths containing self-loops.

B.3: Strategies to minimize the underdetermination problem - nonlinear solvers

B.3.1: Division into subsystems

We divided the whole system into smaller subsystems to minimize the impact of underdetermination. Therefore, we do not need to invert the total association matrix **G** and avoid the entailing ill-conditioning problem. Specially, we developed a nonlinear solver, the T-solver (Appendix B.3.2), to solve transitivity matrix $\mathbf{T}_i$ when **G** is given using Eq. (B12), and another nonlinear solver, the S-solver (Appendix B.3.3) to solve direct association **S** when **G** and $\mathbf{T}_i$ are given by Eq. (B14). Below we use a network containing n entities and constructed from m samples as an example to illustrate the way the whole system is divided into subsystems and the effectiveness of our approach to minimize the underdetermination problem.

We apply the T-solver first, using **G** as inputs and obtaining $\mathbf{T}_i$. We consider subsystems containing a node i and its neighbors j and k. Assuming the network has an average degree $\bar{d}$, the total number of entries in $\mathbf{T}_i$ in those subsystems is $n\bar{d}(\bar{d}-1)/2$, noticing that $T_{i,jk}$ is symmetric with respect to j and k. If the network follows a power-law distribution of connectivity, $\bar{d}(\bar{d}-1)/2$ is usually small and $n\bar{d}(\bar{d}-1)/2 < nm$. Thus, this system is not underdetermined. For each subsystem, we solve for three variables $T_{i,kj}$, $T_{j,ki}$, and $T_{k,ij}$ and it is also not underdetermined. Then we apply the S-solver, using **G** and $\mathbf{T}_i$ as inputs and obtaining **S**. We consider subsystems containing a node i and all its neighbors. The total number of S_{ij} in those subsystems is $n\bar{d}$. When

$\bar{d} < m$ and $n\bar{d} < nm$, this system is also not underdetermined. For a subsystem containing node *i* and its *d* neighbors, the total number of unknown variables *d* is smaller than the total available information *md*, and it is not underdetermined, too.

Below are some of the numbers of the networks under warming and control from the soil microbial community study to justify the validity of our approaches to minimize the underdetermination problem. Note that (i) $nm < n(n-1)/2$ for both networks, indicating that the problem is underdetermined, and the association matrix is ill-conditioned, (ii) $nm > n\bar{d}(\bar{d}-1)/2$ and $nm > n\bar{d}$ for both networks, suggesting that iDIRECT successfully minimized the underdetermination problem.

	n	m	$\bar{d}$	nm	$n(n-1)/2$	$n\bar{d}(\bar{d}-1)/2$	$n\bar{d}$
Warming network	559	120	6.12	67,080	155,961	12,186	3,421
Control network	317	120	3.72	38,040	50,086	2,786	1,179

B.3.2: The T-solver

The T-solver is used to solve for the transitivity matrix among three nodes *i*, *j*, and *k*, that is, $T_{i,kj}$, $T_{j,ki}$, and $T_{k,ij}$. Entries in the transitivity matrix can be solved using Eq. (B12), when the total association matrix is given. For simplicity, let $T_1 = T_{i,kj}$, $T_2 = T_{j,ki}$, $T_3 = T_{k,ij}$, $G_1 = G_{kj}$, $G_2 = G_{ki}$, and $G_3 = G_{ij}$. Eq. (B12) becomes

$$\begin{cases} G_1 = T_1 \oplus (T_2 T_3)\,; \\ G_2 = T_2 \oplus (T_3 T_1)\,; \\ G_3 = T_3 \oplus (T_1 T_2)\,. \end{cases} \tag{B15}$$

In practice, the binary operator is implemented using the associated generator function (see Appendix C.2 for detail) for efficiency. The above equation is transformed into:

$$\begin{cases} \psi\left(1-G_1\right)=\psi\left(1-T_1\right)+\psi\left(1-T_2T_3\right); \\ \psi\left(1-G_2\right)=\psi\left(1-T_2\right)+\psi\left(1-T_3T_1\right); \\ \psi\left(1-G_3\right)=\psi\left(1-T_3\right)+\psi\left(1-T_1T_2\right), \end{cases} \tag{B16}$$

where $\psi(t)$ is the generator function associated with $\oplus$. Eq. (B16) can be solved using standard Newton's method. For convenience, let $\mathbf{T}$ be the vector with components T_1, T_2, and T_3. The residue vector $\mathbf{R}$ and the Jacobian matrix $\mathbf{J}$ are given by

$$\begin{aligned} \mathbf{R}(\mathbf{T}) &= \begin{bmatrix} \psi\left(1-G_1\right)-\psi\left(1-T_1\right)-\psi\left(1-T_2T_3\right) \\ \psi\left(1-G_2\right)-\psi\left(1-T_2\right)-\psi\left(1-T_3T_1\right) \\ \psi\left(1-G_3\right)-\psi\left(1-T_3\right)-\psi\left(1-T_1T_2\right) \end{bmatrix}, \\ \mathbf{J}(\mathbf{T})=\frac{\partial \mathbf{R}(\mathbf{T})}{\partial \mathbf{T}} &= \begin{bmatrix} \psi'\left(1-T_1\right) & T_3\psi'\left(1-T_2T_3\right) & T_2\psi'\left(1-T_2T_3\right) \\ T_3\psi'\left(1-T_3T_1\right) & \psi'\left(1-T_2\right) & T_1\psi'\left(1-T_3T_1\right) \\ T_2\psi'\left(1-T_1T_2\right) & T_1\psi'\left(1-T_1T_2\right) & \psi'\left(1-T_3\right) \end{bmatrix}. \end{aligned} \tag{B17}$$

$\mathbf{T}$ in the brackets highlights that $\mathbf{R}$ and $\mathbf{J}$ depend on $\mathbf{T}$. Then the update formula for $\mathbf{T}$ is

$$\mathbf{T}^{(i+1)}=\mathbf{T}^{(i)}-\left(\mathbf{J}(\mathbf{T}^{(i)})\right)^{-1}\mathbf{R}(\mathbf{T}^{(i)}). \tag{B18}$$

The superscripts (i) means variables at the current iteration. The initial values for T_1, T_2, and T_3 are $T_1 = G_1$, $T_2 = G_2$, and $T_3 = G_3$. The algorithm converges very fast. The residue norm $|\mathbf{R}^{(i)}|$ and the relative increment $|\Delta\mathbf{T}^{(i)}|/|\mathbf{T}^{(i)}|$ become $<10^{-10}$ after 3 iterations in most cases, where $|\bullet|$ denotes the norm of a vector.

B.3.3: The S-solver

The S-solver is used to solve for the direct association strength between node i and all its neighbors k_1, k_2, …, and k_d in the network, that is, S_{ik1}, S_{ik2}, …, and S_{ikd}. For convenience, they

are referred to as S_1, S_2, …, and S_d in this subsection. The same abbreviation applies to the total association strength and the transitivity matrix. For example, G_1 means G_{ik1} now, and T_{12} means $T_{i,k1k2}$ now. Eq. (B14) becomes:

$$\begin{cases} G_1 = S_1 \oplus (T_{12}S_2) \oplus \ldots \oplus (T_{1d}S_d)\,; \\ G_2 = (T_{21}S_1) \oplus S_2 \oplus \ldots \oplus (T_{2d}S_d)\,; \\ \quad \ldots \quad \ldots \quad \ldots \\ G_d = (T_{d1}S_1) \oplus (T_{d2}S_2) \oplus \ldots \oplus S_d, \end{cases} \tag{B19}$$

which is equivalent to the following set of equations,

$$\begin{cases} \psi\left(1-G_1\right) = \psi\left(1-S_1\right) + \psi\left(1-T_{12}S_2\right) + \ldots + \psi\left(1-T_{1d}S_d\right); \\ \psi\left(1-G_2\right) = \psi\left(1-T_{21}S_1\right) + \psi\left(1-S_2\right) + \ldots + \psi\left(1-T_{2d}S_d\right); \\ \quad \ldots \quad \ldots \quad \ldots \\ \psi\left(1-G_d\right) = \psi\left(1-T_{d1}S_1\right) + \psi\left(1-T_{d2}S_2\right) + \ldots + \psi\left(1-S_d\right). \end{cases} \tag{B20}$$

$\psi(t)$ is the generator function associated with the binary operator $\oplus$. There are d unknowns (from S_1 to S_d) and d equations in Eq. (B20). Standard Newton's method can be used to solve the nonlinear system above. For convenience, let $\mathbf{S}$ be the vector with components S_1, S_2, …, and S_d. The residue vector $\mathbf{R}$ and the Jacobian matrix $\mathbf{J}$ are given by

$$\begin{aligned} \mathbf{R}(\mathbf{S}) &= \begin{bmatrix} \psi\left(1-G_1\right) - \psi\left(1-S_1\right) - \psi\left(1-T_{12}S_2\right) - \ldots - \psi\left(1-T_{1d}S_d\right) \\ \psi\left(1-G_2\right) - \psi\left(1-T_{21}S_1\right) - \psi\left(1-S_2\right) - \ldots - \psi\left(1-T_{2d}S_d\right) \\ \ldots \\ \psi\left(1-G_d\right) - \psi\left(1-T_{d1}S_1\right) - \psi\left(1-T_{d2}S_2\right) - \ldots - \psi\left(1-S_d\right) \end{bmatrix}, \\ \mathbf{J}(\mathbf{S}) &= \frac{\partial \mathbf{R}(\mathbf{S})}{\partial \mathbf{S}} = \begin{bmatrix} \psi'\left(1-S_1\right) & T_{12}\psi'\left(1-T_{12}S_2\right) & \ldots & T_{1d}\psi'\left(1-T_{1d}S_d\right) \\ T_{21}\psi'\left(1-T_{21}S_1\right) & \psi'\left(1-S_2\right) & \ldots & T_{2d}\psi'\left(1-T_{2d}S_d\right) \\ \vdots & \vdots & \ddots & \vdots \\ T_{d1}\psi'\left(1-T_{d1}S_1\right) & T_{d2}\psi'\left(1-T_{d2}S_2\right) & \ldots & \psi'\left(1-S_d\right) \end{bmatrix}. \end{aligned} \tag{B21}$$

$\mathbf{S}$ in the brackets highlights that $\mathbf{R}$ and $\mathbf{J}$ depend on $\mathbf{S}$. Then the update formula for $\mathbf{S}$ is

$$\mathbf{S}^{(i+1)} = \mathbf{S}^{(i)} - \left(\mathbf{J}(\mathbf{S}^{(i)})\right)^{-1} \mathbf{R}(\mathbf{S}^{(i)}). \tag{B22}$$

The superscripts (i) means variables at the current iteration. The initial values for S_1, S_2, …, and S_d are $S_1 = G_1$, $S_2 = G_2$, …, and $S_d = G_d$. The advantages of using Eq. (B20) instead of Eq. (B19) are twofold. First, introducing the generator functions $\psi(t)$ associated with the binary operator $\oplus$ enhances computational efficiency significantly (see Appendix C.3 for details). Second, using Eq. (B20) makes computation of the Jacobian matrix very easy, as seen in Eq. (B21). These improvements make iDIRECT fast enough to solve complex networks.

C: Connection to copulas

Copulas in probability theory (14) are bivariate functions satisfying several requirements. They are closely related to the two binary operators $\otimes$ and $\oplus$ for sequential paths and parallel paths introduced in iDIRECT (Appendix B.1). For a given copula $C(u,v)$, the function $C(u,v)$ can be used as a realization for $u \otimes v$, and $1-C(1-u,1-v)$ can be used as a realization for $u \oplus v$. Using copulas, probability-based interpretations of the corresponding binary operators $\otimes$ and $\oplus$ are attained (Appendix C.1). In addition, if the copula $C(u,v)$ is Archimedean (15), corresponding realizations of $\otimes$ or $\oplus$ are both commutative and associative (Appendix C.2). Furthermore, the generator function for each Archimedean copula can be used to enhance the computational efficiency of iDIRECT (Appendix C.3). These subjects are discussed in detail in the following subsections.

C.1: Introduction to copulas

A copula $C(u,v)$ (14) is a bivariate function defined on [0,1]×[0,1], taking values in [0,1], and satisfying the following conditions:

1. *Lower bound*: $C(u,0) = C(0,v) = 0$;
2. *Upper bound*: $C(u,1) = u$ and $C(1,v) = v$;
3. 2-*increasing*: $C(u_1,v_1)-C(u_1,v_2)-C(u_2,v_1)+C(u_2,v_2) \geq 0$, for all $0 \leq u_1 \leq u_2 \leq 1$ and $0 \leq v_1 \leq v_2 \leq 1$.

The 2-increasing property of a copula guarantees the monotonicity with respect to both arguments.

Informally, a copula can be interpreted as a joint distribution function with uniform marginal distributions. According to Sklar's theorem (14), if X and Y are two random variables with joint distribution function $F_{XY}(x,y)$ and marginal distribution functions $F_X(x)$ and $F_Y(y)$, then the following function is a copula:

$$C\left(u,\, v\right) = F_{XY}\left(F_X^{-1}(u),\, F_Y^{-1}(v)\right). \tag{C1}$$

The lower bound and upper bound properties of a copula guarantee that the corresponding cumulative distribution function takes values between 0 and 1; the 2-increasing property guarantees that the corresponding joint distribution density function $f_{XY}(x,y)$ is always positive.

All copulas $C(u,v)$ are bounded by the following two copulas

$$M\left(u,\, v\right) = \max\left\{u + v - 1,\, 0\right\}, \quad W\left(u,\, v\right) = \min\left\{u,\, v\right\}, \tag{C2}$$

such that $M(u,v) \leq C(u,v) \leq W(u,v)$. The functions $M(u,v)$ and $W(u,v)$ are called the lower and upper Fréchet-Hoeffding bounds (14). Another important copula is the independent copula $\Pi(u,v) = uv$, which is associated with two independent random variables, hence the name. For any given copula $C(u,v)$, we can construct a valid operator $\otimes$ for sequential paths (Appendix B.1.1) such that $u \otimes v = C(u,v)$ and a valid operator $\oplus$ for parallel paths (Appendix B.1.2) such that $u \oplus v = 1 - C(1-u,1-v)$. Table S7 lists several commonly used copulas $C(u,v)$ and their corresponding binary operations $u \otimes v$ and $u \oplus v$.

Probability-based interpretations of $\otimes$ and $\oplus$ can be proposed using copulas. As an example, consider the sequential paths *i*-*k* and *k*-*j* in Fig. S12a. Suppose the links are switched on and off randomly, and the association strength of a link is the probability of the link being switched "on". Let X be the event that link *i*-*k* is "on", Y be the event that link *k*-*j* is "on", and u and v be the probabilities of event X and Y. Therefore, the indirect association strength between *i* and *j*, $u \otimes v$, is the probability that both *i*-*k* and *k*-*j* are "on". If X and Y are independent, $u \otimes v = uv$; if X and Y are not entirely independent, $u \otimes v$ varies between the lower Fréchet-Hoeffding bound $M(u,v) = \max\{u+v-1,0\}$ and the upper Fréchet-Hoeffding bound $W(u,v) = \min\{u,v\}$. Similar interpretations can be made for parallel paths. Consider two nodes *i* and *j* connected via two intermediate nodes k_1 and k_2 and two different paths *i*-k_1-*j* and *i*-k_2-*j* (see Fig. S12b). Let X be the event that path *i*-k_1-*j* is "on", Y be the event that path *i*-k_2-*j* is "on", and u and v be the probabilities of event X and Y. The total association strength between *i* and *j*, $u \oplus v$, is the probability that either path *i*-k_1-*j* or *i*-k_2-*j* is "on". If X and Y are independent, $u \oplus v = u+v-uv$; if X and Y are not entirely independent, $u \oplus v$ varies between $1 - M(1-u,1-v) = \max\{u,v\}$ and $1 - W(1-u,1-v) = \min\{u+v,1\}$.

630

631 C.2: Archimedean copulas

632

633 A special class of copulas, the Archimedean copulas (15), are commutative and associative. An

634 Archimedean copula allows the following representation

$$C(u, v) = \psi^{-1}\{\psi(u) + \psi(v)\}, \quad \text{(C3)}$$

635 where ψ: [0,1]↦[0,∞) is the generator function and ψ^{-1} is its inverse. The generator function is a

636 continuous, strictly non-decreasing, and convex function with $\psi(1) = 0$. The commutativity and

637 associativity of Archimedean copulas are automatically satisfied by construction. Table S8 lists

638 some of the most important families of Archimedean copulas and their generator functions, as

639 well as the range of the parameter. Due to the important role they play in the implementation of

640 the binary operator ⊕ for parallel paths, two derived functions $\psi_\theta(1–t)$ and $1–\psi_\theta^{-1}(t)$ are listed in

641 the table, too. When $\theta = 0$ for the Ali-Mikhail-Haq (16), Clayton (17), and Frank families, or $\theta =$

642 1 for the Gumbel and Joe families, the corresponding copula $C(u,v) = uv$ is the independent

643 copula $\Pi(u,v)$, and the corresponding generator function is $\psi(t) = –\ln t$.

644

645 The Clayton family is of special interest because $C_{-1}(u,v)$ is the lower Fréchet- Hoeffding bound

646 $M(u,v) = \max\{u+v–1,0\}$, $C_0(u,v)$ is the independent copula $\Pi(u,v) = uv$, $C_1(u,v)$ corresponds to

647 our preferred choice of $(u+v–2uv)/(1–uv)$ for parallel paths in Eq. (B2), and the limiting case

648 $C_\infty(u,v)$ is the upper Fréchet-Hoeffding bound $W(u,v) = \min\{u,v\}$.

649

650 C.3: Computational efficiency enhancement

651

In practice, the binary operator $\otimes$ is realized as multiplication $\times$, and the implementation is straightforward. However, for the binary operator $\oplus$, the implementation is more difficult. Direct implementation of formulas as presented in Table S7 is not computationally efficient, especially when we want to add multiple terms together.

For example, to compute the sum $u_1 \oplus u_2 \oplus \ldots \oplus u_n$, if we use $u \oplus v = (u+v-2uv)/(1-uv)$ directly, the total number of arithmetic operations is $7\times(n-1) = 7n-7$, where 7 is the number of arithmetic operations for each application of $\oplus$, and $n-1$ is the number of times the binary operation $\oplus$ is performed. Alternatively, if we rewrite the binary operator $\oplus$ in terms of the generator function ψ of the corresponding $C(u,v)$, that is

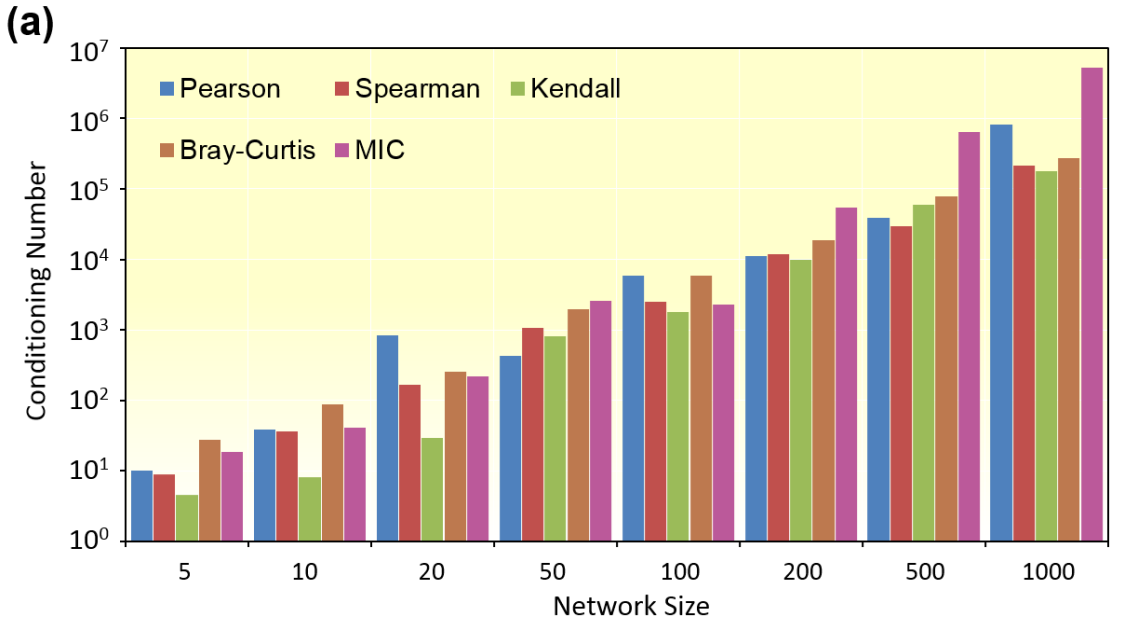
$$u \oplus v = 1 - C\left(1-u,\, 1-v\right) = 1 - \psi^{-1}\left\{\psi(1-u) + \psi(1-v)\right\}. \qquad \text{(C4)}$$

The sum $u_1 \oplus u_2 \oplus \ldots \oplus u_n$ can be expressed in terms of ψ as

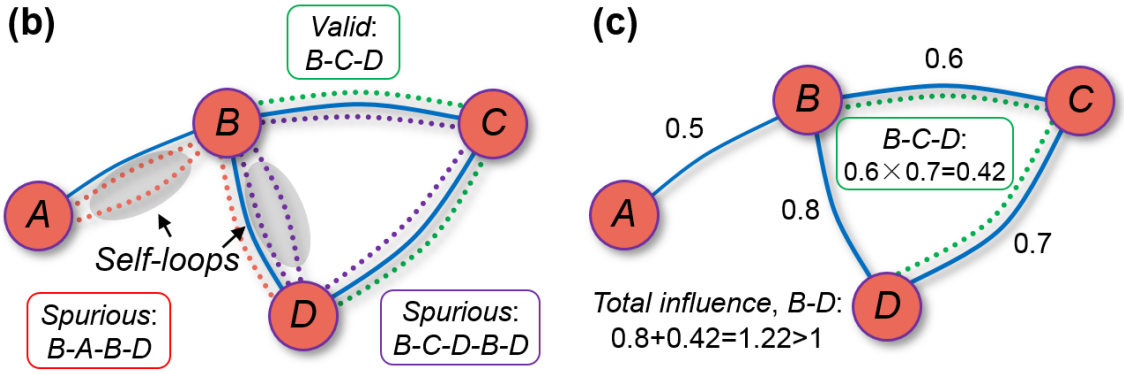
$$u_1 \oplus u_2 \oplus \ldots \oplus u_n = 1 - \psi^{-1}\left\{\psi(1-u_1) + \psi(1-u_2) + \ldots + \psi(1-u_n)\right\}. \qquad \text{(C5)}$$

In other words, $u_1 \oplus u_2 \oplus \ldots \oplus u_n$ is calculated in three steps: (i) use $\psi(1-t)$ to transform each term $u_1, u_2, \ldots, u_n$; (ii) add the transformed terms together; (iii) use $1-\psi^{-1}(t)$ to obtain the result. For $u \oplus v = (u+v-2uv)/(1-uv)$, the corresponding functions are: $\psi(1-t) = t/(1-t)$ and $1-\psi^{-1}(t) = t/(1+t)$. The total number of operations using Eq. (C5) is $2\times n+(n-1)+2 = 3n+1$, where 2 is the number of arithmetic operations for each use of $\psi(1-t)$, $n-1$ is the number of additions performed, 2 is the number of arithmetic operations for the use of $1-\psi^{-1}(t)$. This approach leads to a 133% efficiency increase over directly applying $u \oplus v = (u+v-2uv)/(1-uv)$ when n is large.

673



674



675

676 **Fig. S1. Key issues associated with several previous approaches. (a)** Ill-conditioning of the
677 association matrix. The conditioning number of a matrix is the ratio between the largest and
678 smallest eigenvalues of the matrix. Various network sizes (n = 5, 10, 20, ..., 500, 1,000) and
679 different association measures [Pearson correlation (blue), Spearman's correlation (red), Kendall
680 rank correlation (green), Bray-Curtis dissimilarity (brown), and Maximal Information
681 Coefficients (MIC, purple)] were considered. The number of samples were fixed (m = 20). **(b)**

Self-looping. A valid indirect path B-C-D (dotted green lines), spurious path B-A-B-D (dotted red lines), which contains a self-loop B-A-B, and spurious path B-C-D-B-D (dotted purple lines), which contains self-loops B-C-D-B and D-B-D. Blue solid lines mean direct links. Self-loops are highlighted with grey areas. **(c)** Interaction strength overflow. Direct addition of association strength of direct path B-D (0.8) and indirect path B-C-D (0.6×0.7=0.42) results in total association strength being 0.8+0.42=1.22, which is outside the natural range [0,1] of association data.

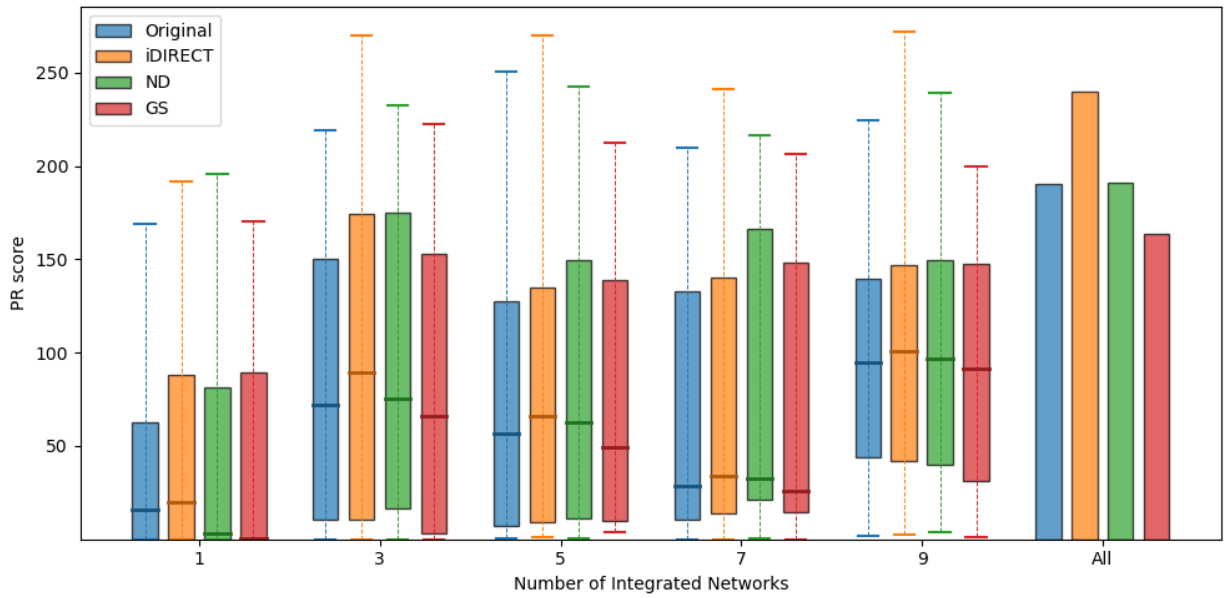
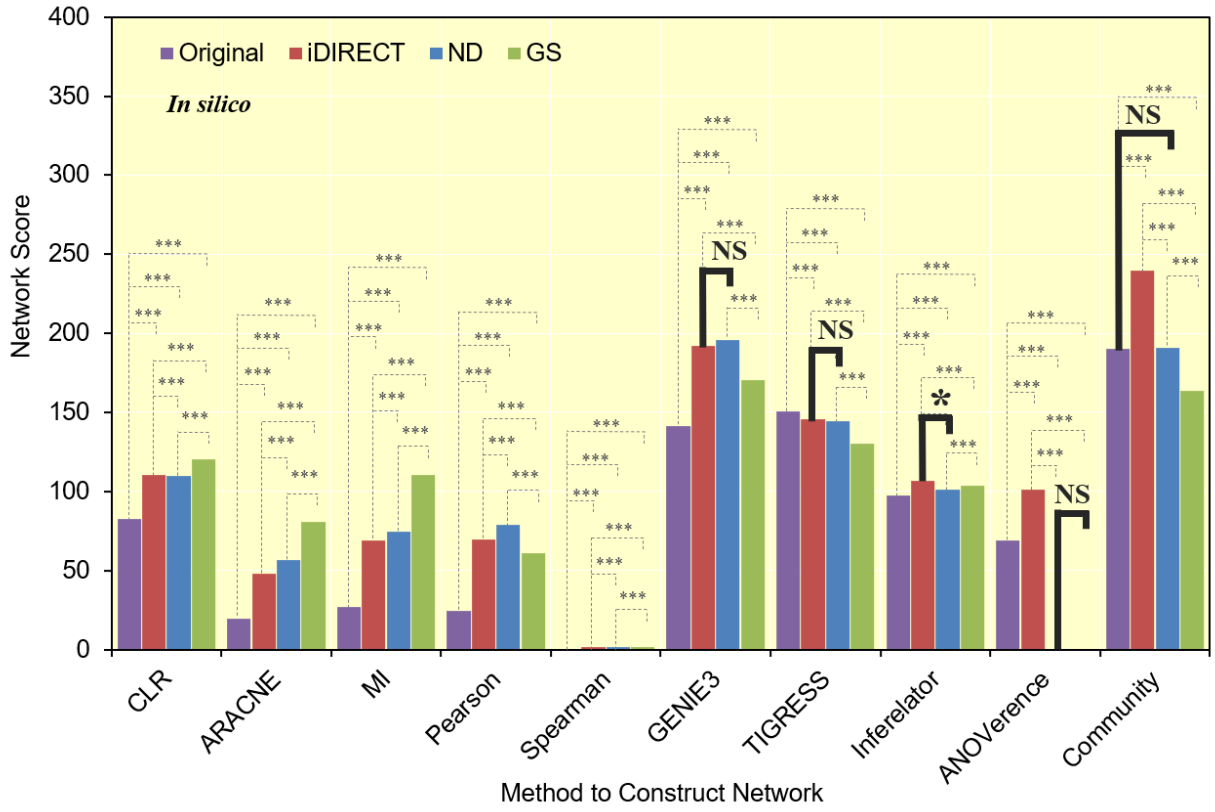


Fig. S2. Improvement of community network score when only a subset of submissions were included. The y-axis represents the Precision-Recall score. The first group of boxes depict the performance distribution of individual submissions (n = 1). The thick bar in the middle represents the mean, the top and bottom of the box represents the 75% and 25% quantile, and the two short bars represent the maximum and the minimum. Subsequent groups of boxes show the performance when n > 1 randomly sampled submissions (n = 3, 5, 7, 9) are integrated. The last group of bars shows the performance when all submissions are integrated. The original, iDIRECT-, ND-, and GS-processed scores are represented by different colors.



704

705 **Fig. S3. Significance of the difference between the scores of the *in silico* network** from the
706 DREAM5 network inference challenge. The x-axis represents different methods to construct the
707 network, and the y-axis represents the corresponding scores. Scores from the original network
708 (purple bars), iDIRECT-processed network (red bars), ND-processed network (blue bars), and GS-
709 processed network (green bars) are represented by different colors. Most of the pairs of scores are
710 significantly different (***, $p < 0.001$), except the five pairs that are highlighted by thicker lines
711 (NS means $p \geq 0.05$ and * means $0.01 \leq p < 0.05$). The significance level is calculated based on
712 Student's t-tests and standard deviations of network scores obtained by randomly switching
713 weights for the first 3,000 edges of each submission. Note that the numbers for Spearman
714 (2.26×10^{-5} for original, 2.90×10^{-3} for iDIRECT, 1.25×10^{-3} for ND, and 2.10×10^{-3} for GS) are too
715 small to show.

716

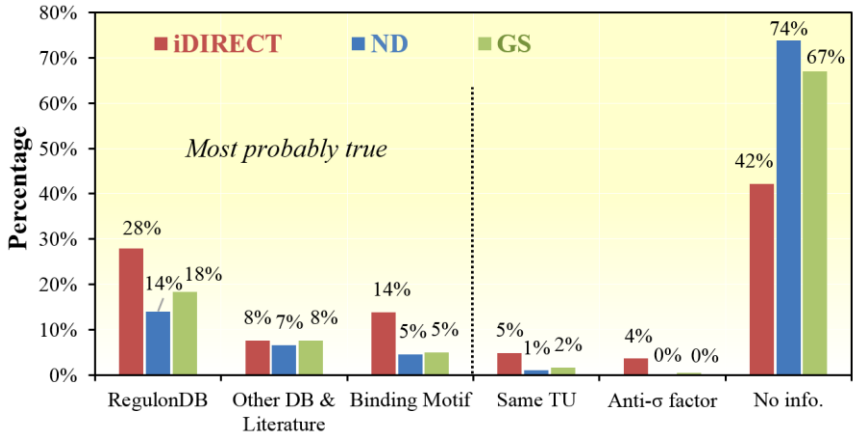


Fig. S4. Assessment of the consistency of the top 500 links identified by iDIRECT with biological evidence. The supporting evidences of the top 500 links in iDIRECT solution with the highest direct association strengths were searched via online databases or available literature. Links with supporting evidence that are most likely true (listed in RegulonDB, found in online databases or literature, or having a binding motif in the promoter region) are separated from links that are unlikely true or lack enough information to decide (involving genes in the same operon, or involving an anti-sigma factor, or no information in the literature).

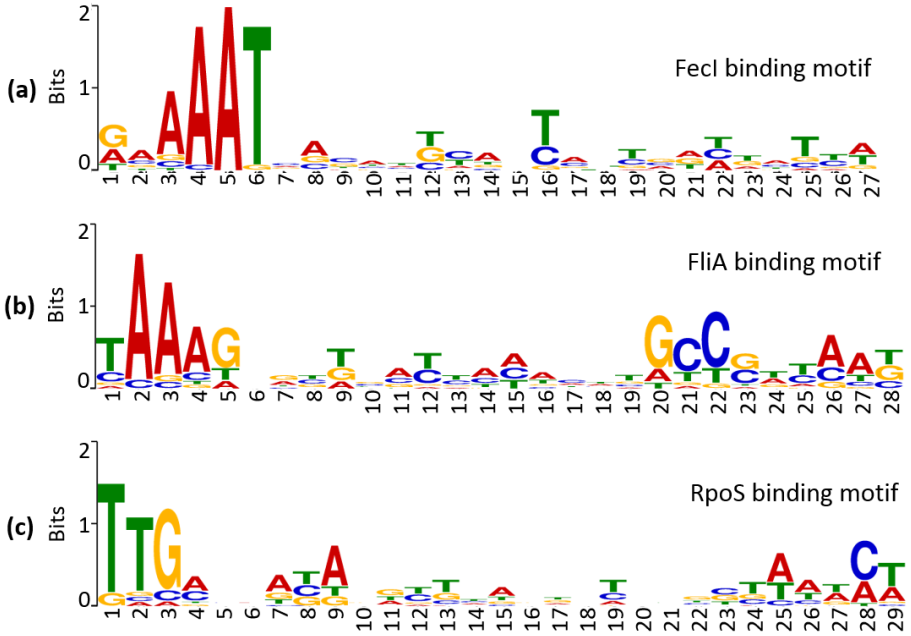


Fig. S5. Consensus sequences in the manually identified binding motifs. Binding motifs of three key regulatory factors identified by iDIRECT are examined. **(a).** FecI binding motif; **(b).** FliA binding motif; **(c).** RpoS binding motif. The consensus sequences of the binding motifs were identified using the MEME Suite from http://meme-suite.org/index.html.

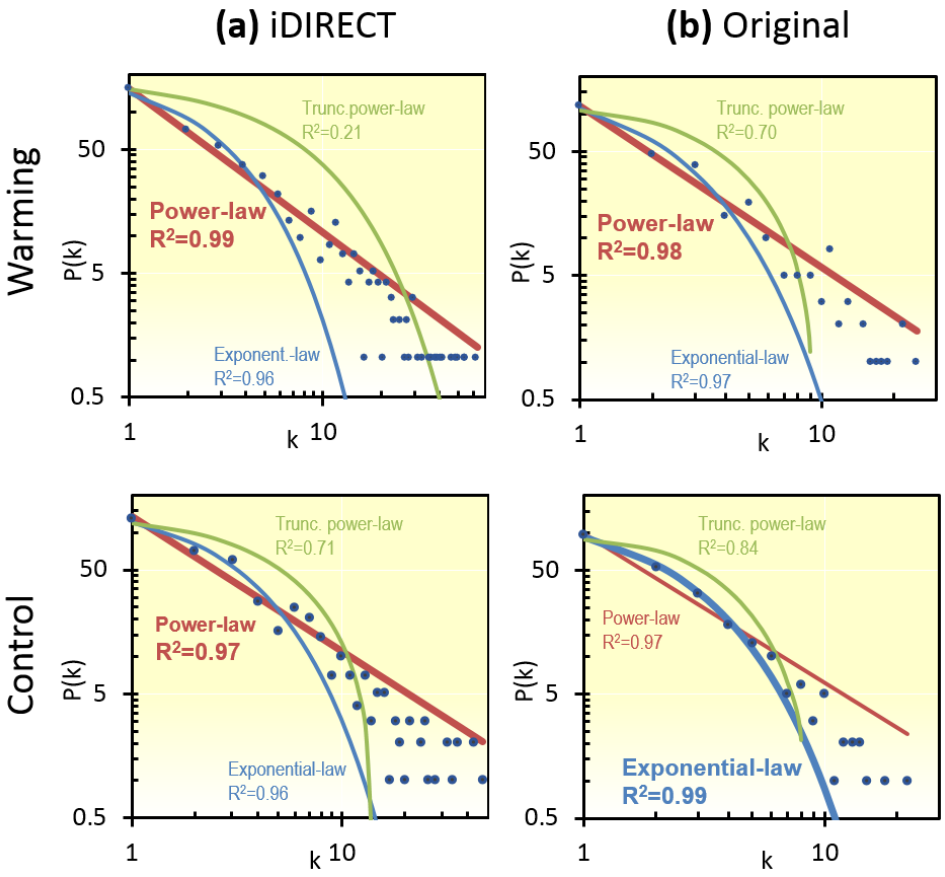


Fig. S6. Degree distribution for microbial molecular ecological networks under warming and control. (a) iDIRECT-processed networks. **(b)** Original networks. The node degree k is plotted against the probability P(k) in a log-log scale. Circular dots were data points, and solid lines represented different regression models (red: power-law, blue: exponential law, and green: truncated power-law). The regression models with the best fitting were highlighted with thicker lines.

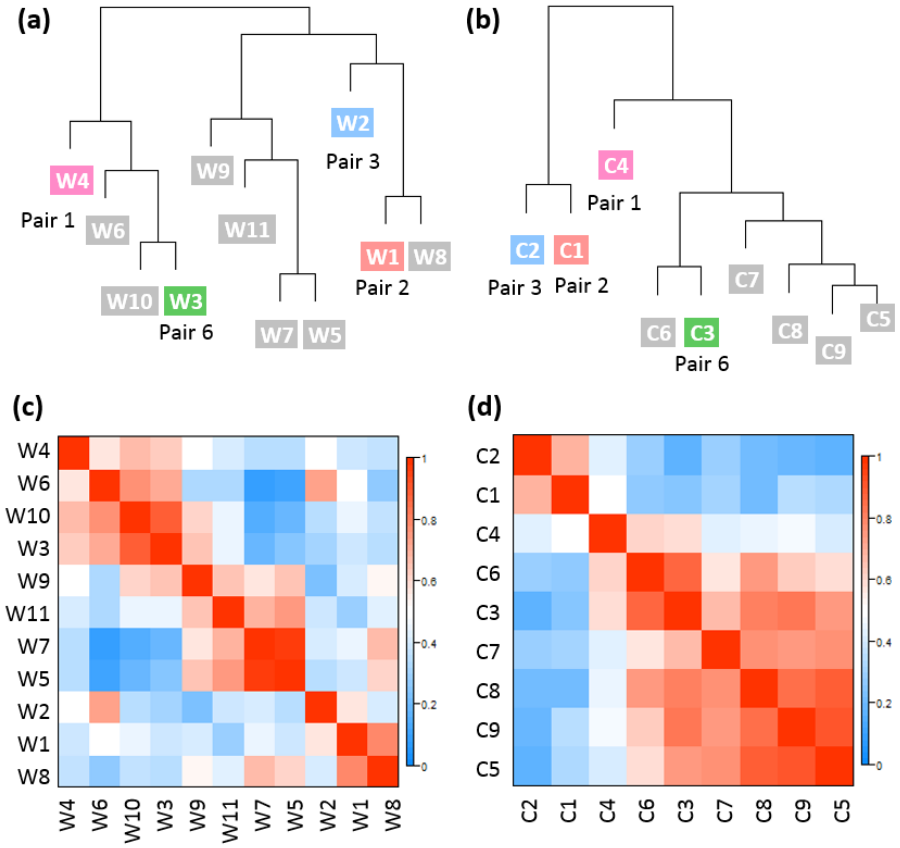


Fig. S7. Module-level higher-order organizations of iDIRECT-processed networks. (a,b) The clustering dendrograms under warming **(a)**, or control **(b)** show the relationships among eigengenes from different modules. Module pairs between warming and control identified by Fisher's exact test (Table S6) were highlighted with same colors. **(c, d)** The heat maps under warming **(c)** and control **(d)** display the correlations between eigengenes of different modules.

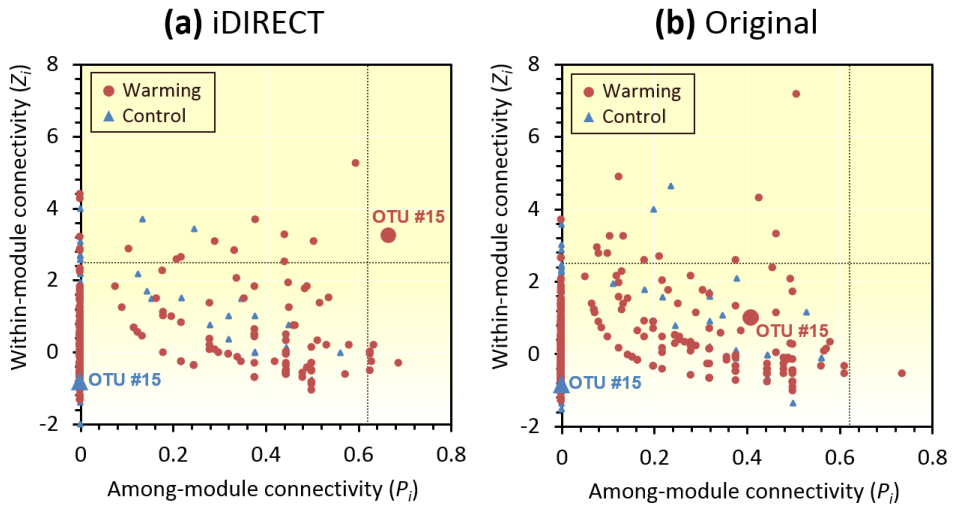


Fig. S8. Comparison of OTU topological roles under warming and control. (a) iDIRECT-processed networks; **(b)** Original networks. The among-module connectivity (P_i) was plotted against the within-module connectivity (Z_i). The nodes are categorized into: peripheral ($P_i < 0.6$, $Z_i < 2.5$), module hub ($P_i < 0.6$, $Z_i \geq 2.5$), connector ($P_i \geq 0.6$, $Z_i < 2.5$) and network hub ($P_i \geq 0.6$, $Z_i \geq 2.5$). Each symbol represents an OTU under control (blue triangular dot) or warming (red circular dot). Locations of the network hub in iDIRECT-processed network, OTU #15, were highlighted. There are 21 module hubs, with 16 for warming network, 8 for control network, and 3 shared by both. There are 5 connectors for the warming network. After the application of iDIRECT, a new network hub, 7 new module hubs, and 5 new connectors appear, while 4 old module hub and 1 old connector disappear.

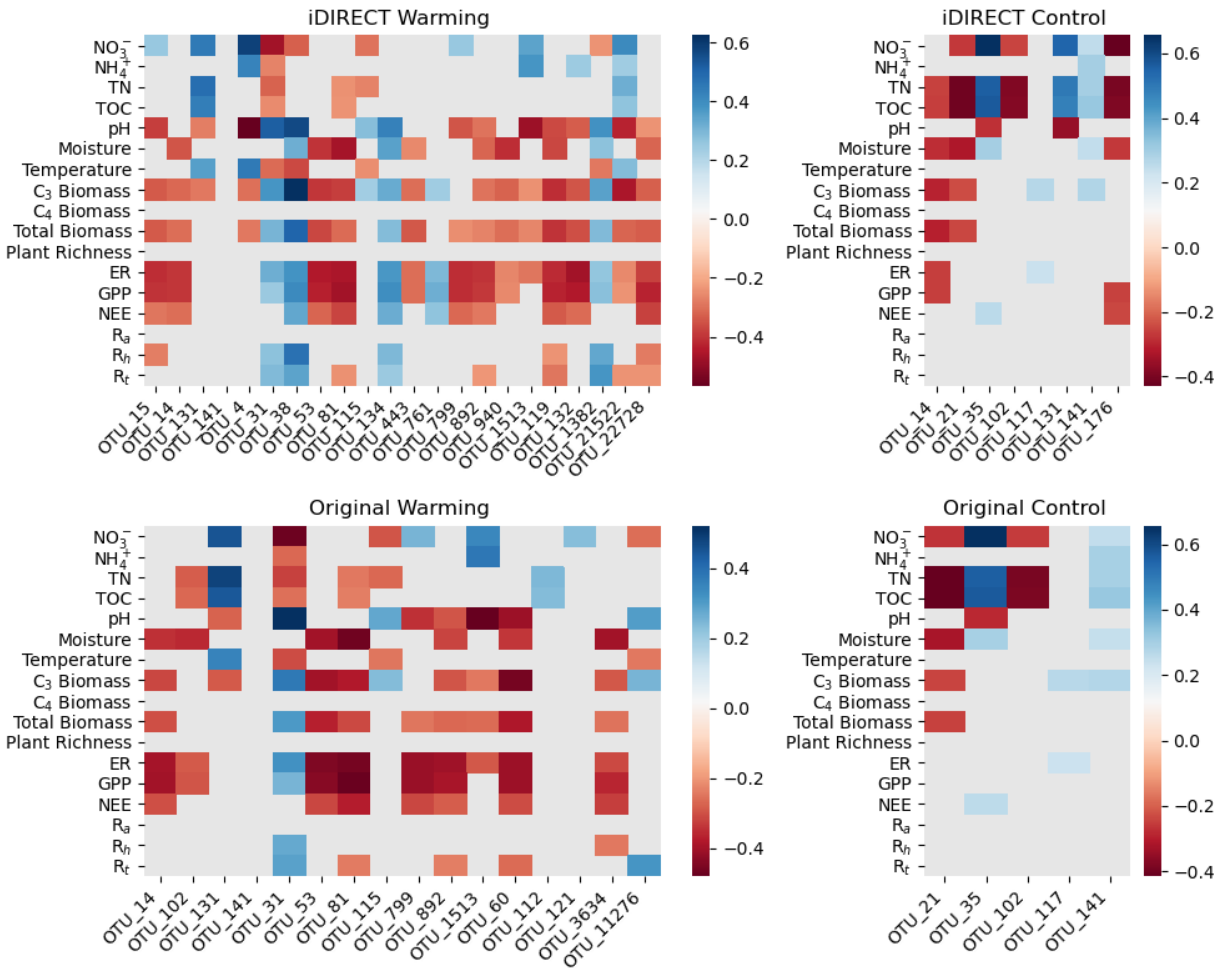


Fig. S9. Comparison of correlations between keystone OTUs abundance and soil, plant and ecosystem functioning variables. The keystone OTUs were either obtained from the original or iDIRECT-processed networks under warming or control. The keystone OTUs are defined as either module hub ($P_i < 0.6$, $Z_i \geq 2.5$), or connector ($P_i \geq 0.6$, $Z_i < 2.5$) or network hub ($P_i \geq 0.6$, $Z_i \geq 2.5$), with P_i being the among-module connectivity and Z_i being the within-module connectivity. The metadata include total nitrogen (TN), total organic carbon (TOC), ecosystem respiration (ER), gross primary productivity (GPP), net ecosystem exchange (NEE, difference between GPP and ER), autotrophic respiration (R_a), heterotrophic respiration (R_h), total soil respiration (R_t), etc. Only significantly correlated pairs ($p < 0.01$) are shown. iDIRECT-

779 processed networks show more significant correlations (42.2% in warming and 29.4% in control)
780 than original networks (33.8% in warming and 27.1% in control).
781

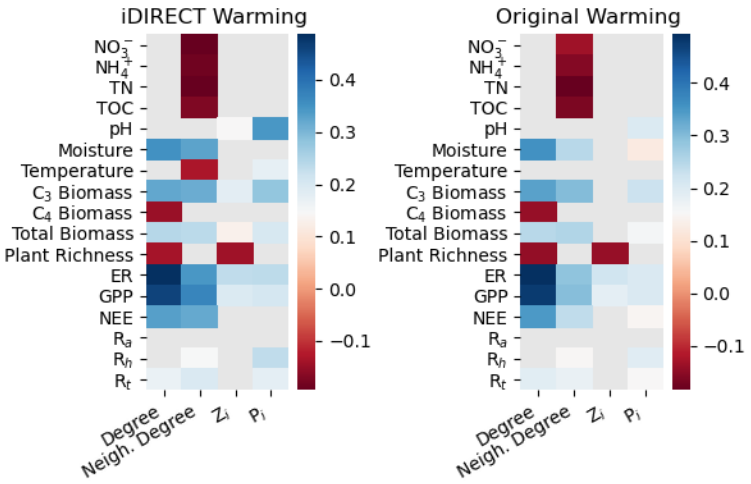


Fig. S10. Comparison of correlations between OTU significance and network properties under warming. The networks are constructed with or without iDIRECT-processing. The x-axis represents nodal network properties, the y-axis represents environmental traits, and the colors represent the Pearson correlation coefficients r between OTU significance and network properties. The OTU significance is calculated and defined as the square of Pearson correlation coefficient (r2) of OTU abundance profile with environmental traits. The nodal network properties considered include node degree, average neighboring node degree, the among-module connectivity (P_i), and the within-module connectivity (Z_i). Only significantly correlated pairs ($p < 0.01$) are shown. More significant correlations are observed between iDIRECT-processed network properties and OTU significance (left panel, 52.9% of all possible pairs) than those between original network properties and OTU significance (right panel, 48.5% of all possible pairs).

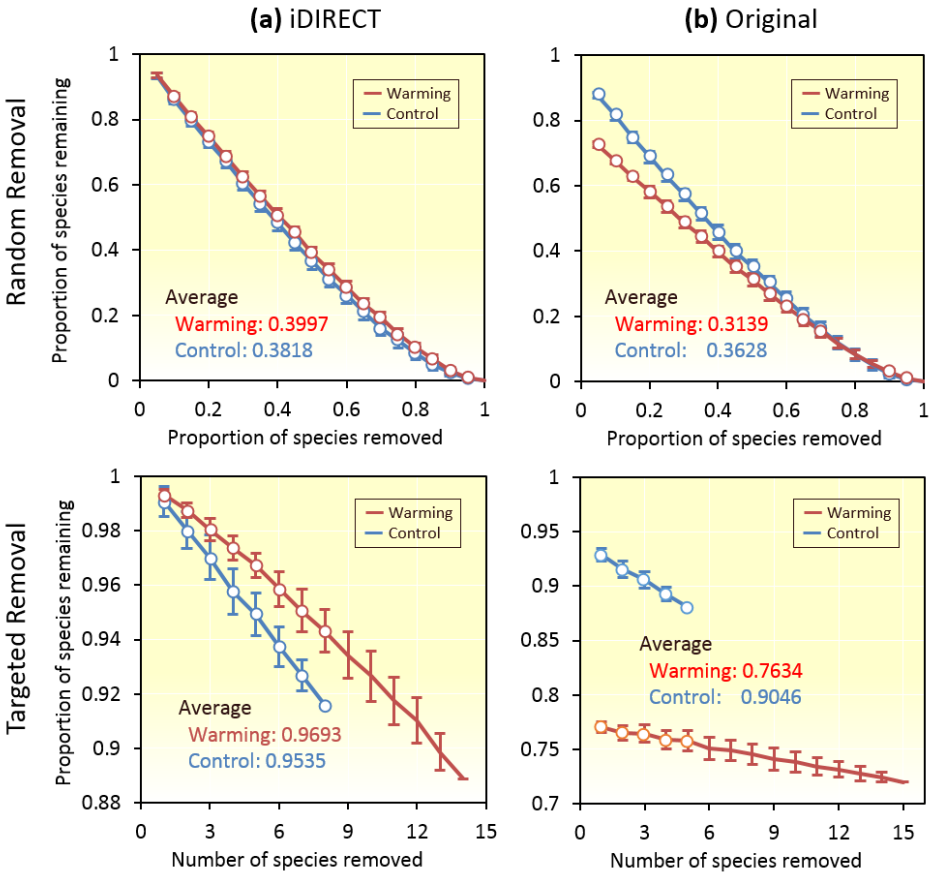


Fig. S11. Robustness analysis of original and iDIRECT-processed networks. (a) iDIRECT-processed networks. **(b)** Original networks. Network robustness to species deletion under control (blue) or warming (red) was represented by simulated microbial species extinction triggered by random species removal or targeted species removal. Warming, red; Control, blue. Error bars represented standard deviation of 100 repetitions of each simulation. Empty dots meant significant difference between warming and control with $p < 0.05$.

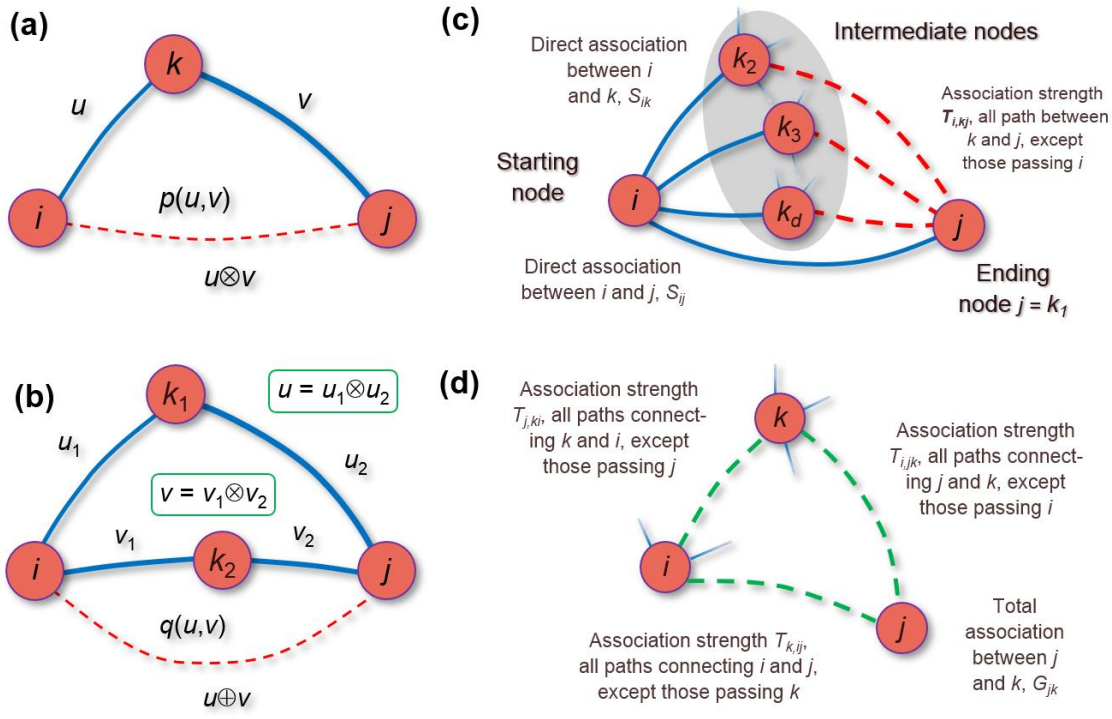


Fig. S12. Sequential paths, parallel paths, and strategies to eliminate self-looping. (a) Sequential paths: Node *i* and node *j* are indirectly linked through an intermediate node *k*. The indirect association strength between *i* and *j* is $u \otimes v$, where *u* and *v* are association strength of each path, respectively. **(b)** Parallel paths: Node *i* and *j* are indirectly linked via two distinctive paths passing node k_1 or node k_2. The combined strength of these two paths is $u \oplus v$, where *u* and *v* are association strength of each path. **(c)** Indirect association between two nodes through intermediate nodes. The starting node is *i*, the ending node is *j*, and the intermediate nodes k_i are neighbors of *i*. The indirect association between *i* and *j* via one of the intermediate node *k* is the product of the direct association S_{ik} between *i* and *k* and the association $T_{i,kj}$ between k_i and *j* except those passing *i*. Spurious paths due to self-looping are removed because they are excluded in the definition of $T_{i,kj}$. **(d)** Calculation of the transitivity matrix. The total association G_{kj} between *k* and *j* is the sum of $T_{i,kj}$ (associations between node *k* and *j* without passing node *i*) and $T_{j,ki}T_{k,ij}$ (associations between *k* and *j* passing node *i*).

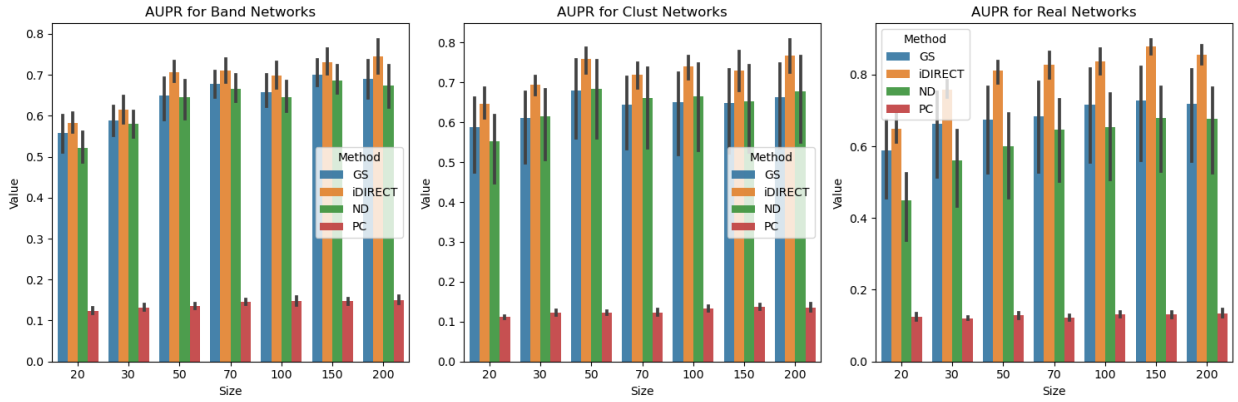


Fig. S13. Area Under Precision-Recall curves (AUPR) for different network types (band-like: left panel, clustered: middle panel, scale-free: right panel). The x-axis represents the sample size used in the study, and the y-axis represents the performance as measured by AUPR. Different methods (GS, iDIRECT, ND and PC) are represented as bars with different colors. The error bar represents 95% confidence level from 10 runs each.

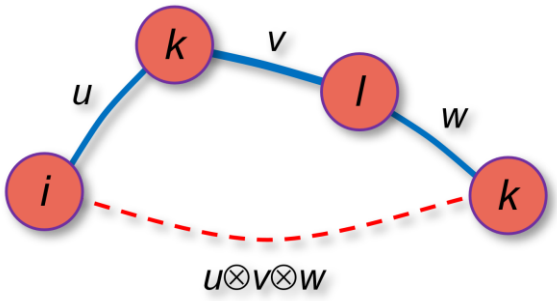


Fig. S14. Nodes connected via multiple intermediate nodes. Node *i* and *j* are indirectly connected via intermediate nodes *k* and *l*. The indirect association strength between *i* and *j* is $u \otimes v \otimes w$, where *u*, *v*, and *w* are the association strengths of direct links *i*-*k*, *k*-*l*, and *l*-*j*, respectively.

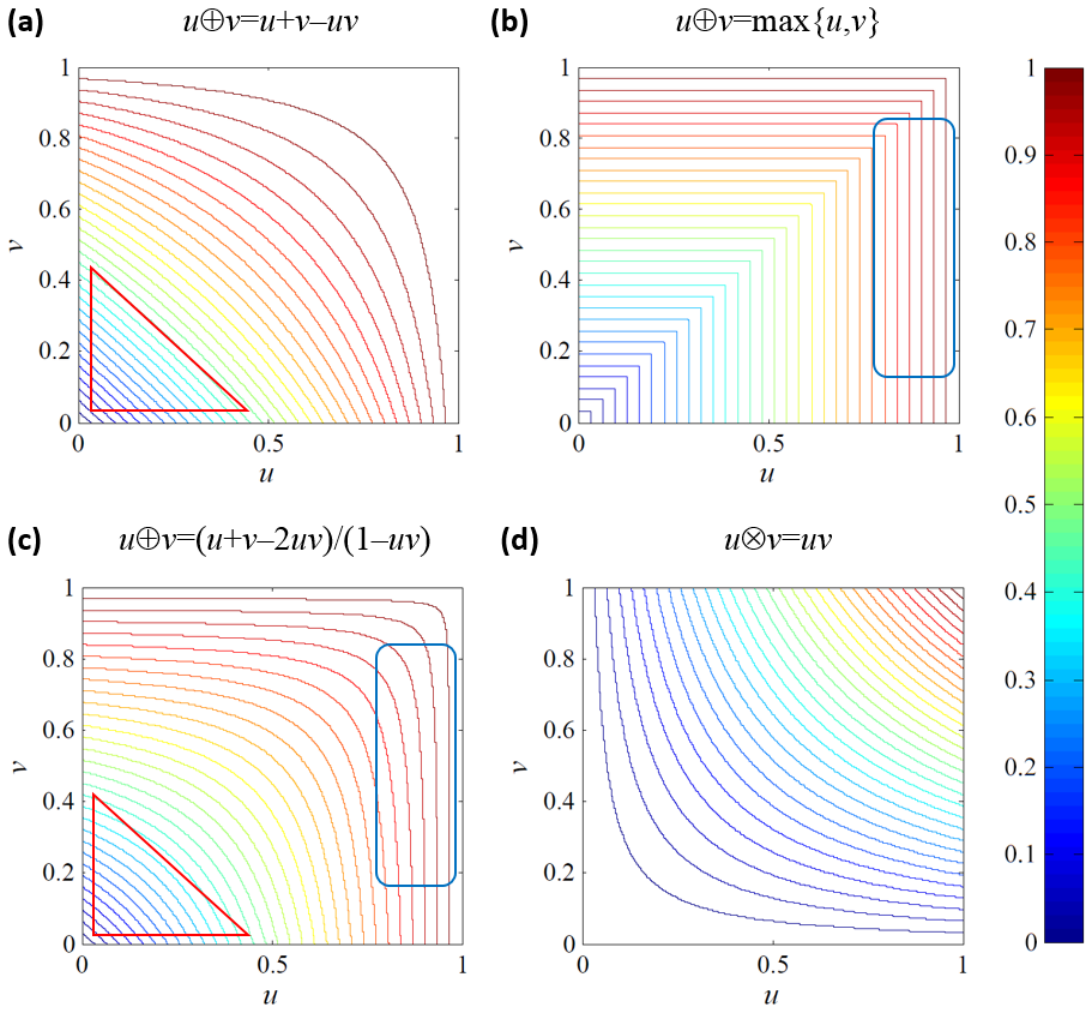


Fig. S15. Contour plot of copula-based functions. (a, b, c). Different choices of the binary operation $u \oplus v$ for parallel paths. Color bar for the contour plot was shown on the right. The red triangular boxes highlighted the areas where the contour lines had almost a constant slope of −1; the blue rectangular boxes highlighted the areas where the contour lines were almost vertical. **(d)** Ordinary multiplication $u \otimes v = uv$ for sequential paths.

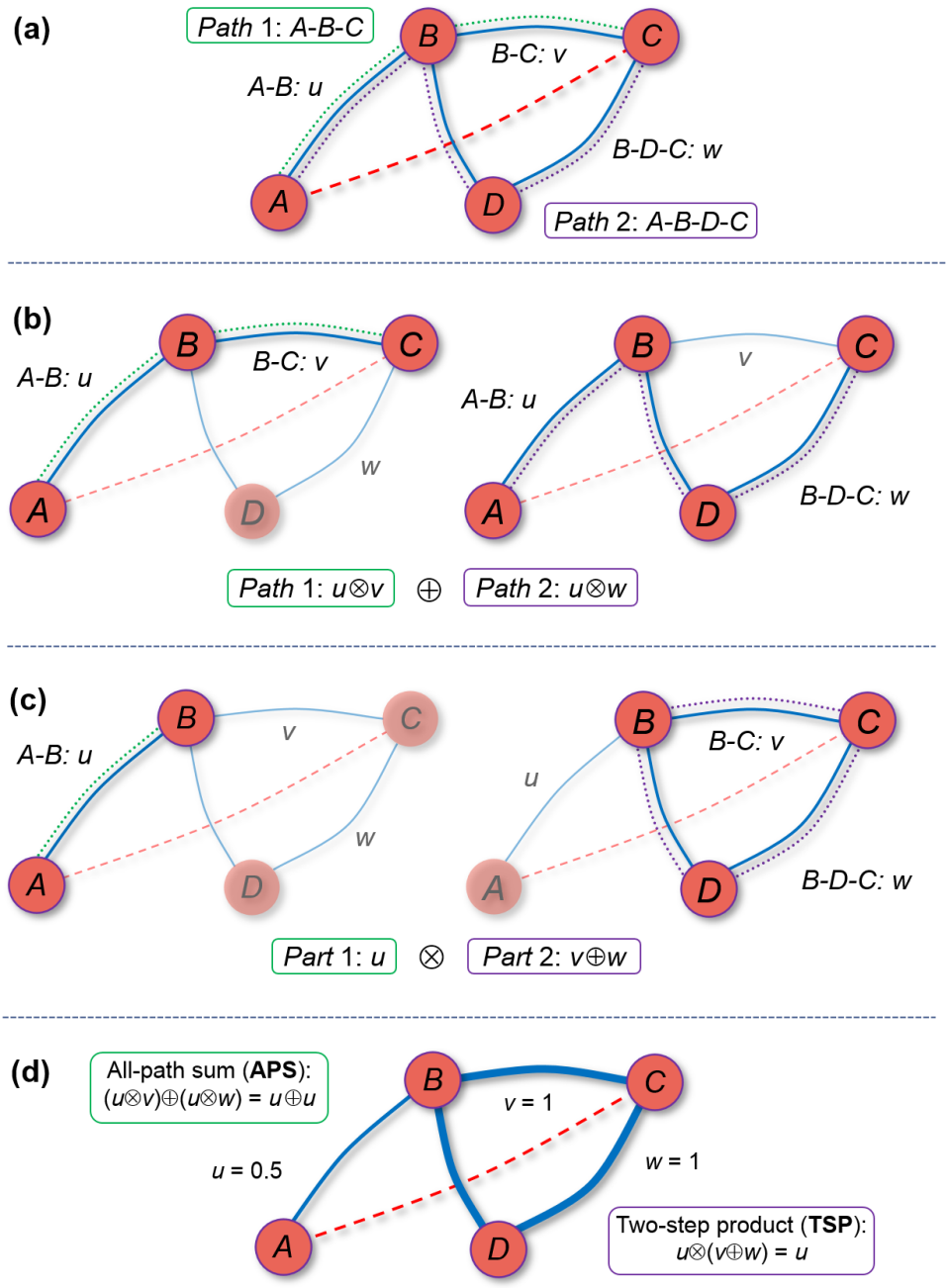


Fig. S16. Two assembly strategies to calculate total association strength. (a) A network of four nodes with two paths *A*-*B*-*C* and *A*-*B*-*D*-*C* connecting *A* and *C*. Blue solid lines: direct links, red dashed lines: indirect link between *A* and *C*, dotted lines: indirect paths (green and purple). **(b)** The All-Path Sum (APS) strategy. The total association strength between *A* and *C* is the sum

of Path 1 and Path 2: $(u \otimes v) \oplus (u \otimes w)$. **(c)** The Two-Step Product (TSP) strategy. The total association strength between *A* and *C* is the product of part 1 and part 2: $u \otimes (v \oplus w)$. **(d)** Comparison of APS and TSP strategies for Case 3 in Table S10. Association strength is $u = 0.5$ for link *A*-*B*, and $v = w = 1$ for link *B*-*C* and *B*-*C*-*D*, respectively. The APS strategy yields $u \oplus u$, while the TSP strategy yields u.

865

866 **Table S1. Experimental and computational evidence for the edges of the four modules in the iDIRECT network.** For each pair
867 of transcriptional factor (TF) and the regulated gene, we listed the evidence code and evidence strength assignment from RegulonDB,
868 or nucleotide sequences and location of the binding motif that was identified.

869

Gene name	Gene name	Annotation	Evidences	RegulonDB strength	Binding motif nucleotide sequence	Loc. of the binding motif
fliA module						
fliA	cheA	Fused chemotactic sensory histidine kinase in two-component regulatory system with cheb and chey: sensory histidine kinase/signal sensing protein	['HIPP', '**GEA**'[a]]	Weak		
fliA	cheB	Chemotaxis-specific methylesterase	['HIPP', 'TIM', **'GEA'**[a]]	Strong		
fliA	cheR	Chemotaxis regulator, protein-glutamate methyltransferase	['HIPP', 'TIM', **'GEA'**[a]]	Strong		
fliA	cheW	Purine-binding chemotaxis protein	['HIPP', '**GEA**'a]	Weak		
fliA	cheY	Chemotaxis regulator transmitting signal to flagellar motor component	['HIPP', 'TIM', **'GEA'**[a]]	Strong		
fliA	cheZ	Chemotaxis regulator, protein phosphatase for chey	['HIPP', 'TIM', '**GEA**'[a]]	Strong		
fliA	flgK	Flagellar hook-associated protein K	['HIPP', **'GEA'**[a,b]]	Weak		
fliA	flgL	Flagellar hook-associated protein L	['HIPP', **'GEA'**[a,b]]	Weak		

fliA	flgN	Export chaperone for flgk and flgl	['HIPP', '**GEA**'[a]]	Weak
fliA	fliC	Flagellin	['FP', 'HIPP', '**GEA**'[a]]	Strong
fliA	fliD	Flagellar capping protein	['FP', 'HIPP', '**GEA**'[a]]	Strong
fliA	fliE	Flagellar hook-basal body protein flie	['HIPP']	Weak
fliA	fliF	Flagellar M-ring protein	['HIPP']	Weak
fliA	fliG	Flagellar motor switch protein G	['HIPP']	Weak
fliA	fliH	Flagellar biosynthesis; export of flagellar proteins?	['HIPP']	Weak
fliA	fliI	Flagellum-specific ATP synthase	['HIPP']	Weak
fliA	fliJ	Flagellar biosynthesis chaperone	['HIPP']	Weak
fliA	fliK	Flagellar hook-length control protein	['HIPP']	Weak
fliA	fliL	Flagellar basal body-associated protein flil	['HIPP', 'TIM']	Strong
fliA	fliM	Flagellar motor switch protein M	['HIPP', 'TIM']	Strong
fliA	fliN	Flagellar motor switch protein	['HIPP', 'TIM']	Strong
fliA	fliO	Flagellar biosynthesis	['HIPP', 'TIM']	Strong
fliA	fliP	Flagellar biosynthesis protein P	['HIPP', 'TIM']	Strong
fliA	fliS	Flagellar protein flis	['FP', 'HIPP', '**GEA**'[a]]	Strong
fliA	fliT	Predicted chaperone	['FP', 'HIPP', '**GEA**'[a]]	Strong
fliA	fliZ	Hypothetical protein	['HIPP', 'TIM']	Strong
fliA	flxA	Qin prophage; predicted protein	['AIPP', 'HIPP']	Weak
fliA	motA	Flagellar motor protein mota	['HIPP', '**GEA**'[a]]	Weak

fliA	motB	Flagellar motor protein motb	['HIPP', **'GEA'[a]**]	Weak		
fliA	tap	Methyl-accepting protein IV	['HIPP', 'TIM', **'GEA'[a]**]	Strong		
fliA	tar	Methyl-accepting chemotaxis protein II	['HIPP', 'TIM', **'GEA'[a]**]	Strong		
fliA	tsr	Methyl-accepting chemotaxis protein I, serine sensor receptor	['HIPP', 'TIM']	Strong		
fliA	ycgR	Protein involved in flagellar function	['AIPP', 'HIPP']	Weak		
fliA	yjcZ	Hypothetical protein	['HIPP']	Weak		
fliA	flgA	Flagellar basal body P-ring biosynthesis protein A	**[gSELEX][c]**		AAAATGGGTCGCTATTTATGCCGTTGAT	-80
fliA	flgB	Flagellar basal-body rod protein B			TACAACGTGAATTGTACCTGTCCGCAAT	-136
fliA	flgC	Flagellar basal-body rod protein C			TCGCGAACGCACCCAGTTTGCCGATAAC	-113
fliA	flgD	Flagellar basal body rod modification protein D			CAAAGGGCTACGTAAAAATGCCGAACGT	-158
fliA	flgE	Flagellar hook protein E			TAACGGTGGTACACAACTGGTTGCCCAG	-164
fliA	flgF	Flagellar component of cell-proximal portion of basal-body rod			TAAAGAACTGGTCAATATGATCGTTGCC	-126
fliA	flgG	Flagellar component of cell-distal portion of basal-body rod			TAAGGCGTTTACGCCGCATCCGGCAAGA	-70
fliA	flgH	Flagellar L-ring protein precursor H			TAAAGCGGTGTCCACCACCGATCAGATG	-105
fliA	flgI	Homolog of Salmonella P-ring of flagella basal body			CAATGGCTACATTAACGAAGCGCAAAAT	-85
fliA	flgJ	Flagellar biosynthesis protein flgj			CAAAGCGTACGTTCCAGCGCCAGCCTCA	-141
fliA	flhB	Flagellar biosynthesis protein B			TAAATCCCGCCTGTTTTGCCCCTTACTC	-93
fliA	yecR	Hypothetical protein			TAAAATAGTGCTTTCTCTTACTCTTCTG	-37

fliA	yhjH	Hypothetical protein	**['GEA', 'CHIP'][a,c] [TA][a]**	TAAAGTTCTGCCCTTACGCGCCGATAAT[#]	-76
fliA	yjdA	Conserved protein with nucleoside triphosphate hydrolase domain	**['HIPP']* ['GEA', 'CHIP'][a,c] [TA][a]**	TAAATAAAATAACAAAATTTGCTTTAAG	-41
fliA	ymdA	Hypothetical protein		GAAAGTATGGATAACACAACCCTCAAGG	-57
fecI module					
fecI	bfd	Bacterioferritin-associated ferredoxin		TGAAATAAGAACTATTTTCATTTATTT	-53
fecI	cirA	Ferric iron-catecholate outer membrane transporter		ACAAATCAGAGGCTGTTCCGGCTTTCT	-146
fecI	efeO	Ferrous iron transport system protein (ycdo)		GGAAATCGCCTTCGATATGAGTGCGGT	-251
fecI	entA	2,3-dihydroxybenzoate-2,3-dehydrogenase		AAGAATTACTGCCAGCACCTATCCCCG	-248
fecI	entB	Isochorismatase		TTAAATTACCGGATCGCGTGGAGTGTG	-124
fecI	entC	Isochorismate synthase		GAAAATATAAATGATAATCATTATTAA	-55
fecI	entE	2,3-dihydroxybenzoate-AMP ligase component of enterobactin synthase multienzyme complex		GAAAATCAGGTGCGTCTGTTTGCCGGA	-130
fecI	entF	Enterobactin synthase multienzyme complex component, ATP-dependent		GGCAATTCAGTCTGTGGCCGCAACAAT	-152
fecI	exbB	Membrane spanning protein in tonb-exbb-exbd complex		GCAAATAGTAATGAGAACGACTATCAA	-89
fecI	exbD	Membrane spanning protein in tonb-exbb-exbd complex		TCAAATGGGCCGCGGTAACGGCTATCT	-371
fecI	fepA	Iron-enterobactin outer membrane transporter		AGAAATATATTGATAATATTATTGATA	-202

fecI	fepB	Iron-enterobactin transporter subunit	GAAAATGAGAAGCATTATTGATGGATT	-231
fecI	fhuA	Ferrichrome outer membrane transporter	TAAAATAACATCCCATCTAAGATATTA	-180
fecI	fhuF	Ferric iron reductase involved in ferric hydroximate transport	ATAAATCCCTTGCTATCGGGTAAACCT	-74
fecI	fiu	Predicted iron outer membrane transporter	GAAAATCGCTCCAAGTGATAATGCTTA	-146
fecI	nrdE	Ribonucleotide-diphosphate reductase alpha subunit	GGAAATGCGGCGTGCCGTGGCTGTACC	-89
fecI	nrdF	Ribonucleotide-diphosphate reductase beta subunit	TGAAATTGAAGGCTGCGTCTCCTGTGC	-44
fecI	nrdH	Glutaredoxin-like protein	AAAAATCCCCCTACCCCGTCACGCTCA	-173
fecI	tonB	Membrane spanning protein in tonb-exbb-exbd complex	AAAAATGACATTTTCACTGATCCTGAT	-108
fecI	ybaN	Conserved inner membrane protein	GAAAATGATAATTGTTATGCTAAAGTA	-56
fecI	ybdB	Proofreading thioesterase in enterobactin biosynthesis	GAAAATCGCCCGTCCACAAGAGATCGC	-121
fecI	ybdZ	Hypothetical protein	AAGAATCCATTTTCTGGCGTCAGGTTG	-119
fecI	ydiE	Hypothetical protein	GATAATAAGAATCATTGTTATATCAAT	-42
fecI	yncE	Hypothetical protein	GAAAATAATGATTACCATTCCCATTTA	-107
fecI	yqjH	Ribonucleotide-diphosphate reductase beta subunit	ACAAATCGCTTGCATTTATCATGATTA	-92
rpoS module				
rpoS	nlpD	Predicted outer membrane lipoprotein	TTGCCGCAGGTCAGCGTATCGTGAACATC	-105
rpoS	yncL	Hypothetical protein	TTGCGGATTTTCTTAACCCGTACTATACA	-59
rpoS	yphA	Predicted inner membrane protein	CTGTAACCAGGATAATTAGCGAATATCTC	-103
rpoS	bfr	Bacterioferritin, iron storage and detoxification protein	TTGACTTACTCGTAAGCCGTTCTACTCTT	-61

rpoS	ygaU	Hypothetical protein		TTGACACTGCTTGGGTATATCCCCGGTT	-147
rpoS	gst	Glutathionine S-transferase		GTCACTGGAAGTCTATGGTCGCGTATTCT	-254
rpoS	yodC	Hypothetical protein		TGGCGATGATATTACCGACTGTTTTAAAT	-189
rpoS	ivy	Inhibitor of vertebrate C-lysozyme		TTGATAACAAATGCTGATATTGGAAATAT	-79
rpoS	yahK	Predicted oxidoreductase, Zn-dependent and NAD(P)-binding		TTGGCTATATTCAATGGACGCGTTTTGCC	-84
rpoS	yahO	Hypothetical protein		TCACGAACAGTCCTACGGTCAGGTAAACG	-232
rpoS	yeaQ	Conserved inner membrane protein		TTGTGCTATGCTTTTTATCAGCGACTAAC	-62
rpoS	yncB	Predicted oxidoreductase, Zn-dependent and NAD(P)-binding	**[gSELEX][c]**	TTGCAGAGGGGATGTGACGGCTGCAAACA	-91
rpoS	ydiZ	Hypothetical protein		TTGAAGAGATGGTTCGTTTTGGCGTAGCT	-217
rpoS	yehE	Hypothetical protein		TTGATCATACAGGCAATGCTTCATTATCA	-120
rpoS	yoaC	Hypothetical protein		TTGATATTAGATGCAAATTAAGGTCATAT	-71
rpoS	hdhA	7-alpha-hydroxysteroid dehydrogenase		TTGCAGCGAAATAATCCTCTCTTTATCTG	-126
rpoS	ytjA	Hypothetical protein		TTGTCGGGAGGCGCGATGTGCACCACACT	-101
rpoS	ygaM	Hypothetical protein		TTCACAACGCTTTCAGAAAAGTCCATAAA	-90
bolA module					
bolA	dsrB	Hypothetical protein		CC***GCCAG***C, CC***GCCAG***T, CT***GCCAG***A	-236, -131, -55
bolA	yoaC	Hypothetical protein		CGA**CCAG**A, GT**GCCA**TA	-354, -93
bolA	yqaE	Pmp3 family protein, a predicted membrane protein of unknown function		ATA**CCAG**C	-130
bolA	cysQ	PAPS (adenosine 3'-phosphate 5'-phosphosulfate) 3'(2'),5'-bisphosphate nucleotidase		TG***GCCAG***G	-221
bolA	ymgE	PF04226 family protein ymge		TT***GCCAG***T, CGC**CCAG**C	-337, -89

bolA	yncL	Uncharacterized protein, contains a predicted transmembrane segment	CA***GCCAG***A, TCA**CCAG**T	-184, -124
bolA	ydhL	Conserved protein	AC**GCCA**AA, TC***GCCAG***G	-273, -149
bolA	ecnB	Bacteriolytic entericidin B membrane lipoprotein	TT***GCCAG***C, CT***GCCAG***C	-206, -122
bolA	sfsA	Sugar fermentation stimulation protein A, putative DNA-binding transcriptional regulator of maltose metabolism	AC***GCCAG***C, CC***GCCAG***G	-155, -129
bolA	csrA	Carbon storage regulator	TA***GCCAG***T, AT**GCCA**TG	-276, -117
bolA	yccX	Predicted acylphosphatase	CA***GCCAG***T, AC**GCCA**TT	-374, -365

[HIPP] Human inference of promoter position; [TIM] Transcription initiation mapping; [GEA] Gene expression analysis; [FP] Foot-printing; gSELEX: gSELEX-Seq; CHIP: ChIP-PCR

Bold font and grey highlight: new evidence found in literature; bold font, italicized, and underlined: perfect binding core sequence; bold font: imperfect dining core sequence (Dressaire C et al 2015);

a: Zhao K et al 2007; b: Fitzgerald DM et al 2014; c: Shimada T et al 2017; *: gene name as crfC on RegulonDB; #: same prediction as literature b

Table S2. Topological properties of iDIRECT-processed and original networks under warming. The standard deviations of topological properties from the random networks are used for the Student *t* test of their statistical significance between iDIRECT-processed and original networks.

		Warming		
		iDIRECT	**Original**	**p**
Empirical	Total nodes	432	489	---
	Total links	1139	1572	---
	Average connectivity	5.273	6.429	---
	Connectance	0.01223	0.01318	---
	R2 of power law	0.886	0.913	---
	Average vulnerability	0.003272	0.002754	7.9×10^{-143}
	Average clustering coefficient (avgCC)	0.265	0.321	2.9×10^{-133}
	Average path distance (GD)	6.268	5.809	7.4×10^{-174}
	Module #	33	24	1.9×10^{-73}
	Relative modularity	0.9219	0.8092	7.8×10^{-237}
Random networks	Average clustering coefficient (avgCC)	0.049±0.006[a]	0.071±0.006	6.7×10^{-63}
	Average path distance (GD)	3.483±0.032	3.284±0.027	6.4×10^{-107}
	Modularity	0.397±0.005	0.338±0.005	1.4×10^{-157}

[a] mean value of topological properties followed by standard deviations from 100 simulations.

Table S3. Topological properties of iDIRECT-processed and original networks under control. The standard deviations of topological properties from the random networks are used for the Student *t* test of their statistical significance between iDIRECT-processed and original networks.

		Control		
		iDIRECT	**Original**	**p**
Empirical	Total nodes	250	284	---
	Total links	399	504	---
	Average connectivity	3.192	3.549	---
	Connectance	0.01282	0.01254	---
	R2 of power law	0.937	0.926	---
	Average vulnerability	0.004334	0.004163	5.4×10^{-10}
	Average clustering coefficient (avgCC)	0.263	0.298	4.9×10^{-87}
	Average path distance (GD)	5.145	4.935	1.2×10^{-57}
	Module #	29	23	2.1×10^{-46}
	Relative modularity	0.3565	0.1993	3.2×10^{-195}
Random networks	Average clustering coefficient (avgCC)	0.024±0.007[a]	0.028±0.007	1.7×10^{-5}
	Average path distance (GD)	4.172±0.070	3.921±0.055	4.6×10^{-69}
	Modularity	0.574±0.008	0.529±0.006	7.7×10^{-103}

[a] mean value of topological properties followed by standard deviations from 100 simulations.

Table S4. Taxonomic information for keystone taxa under warming and control, before and after applying iDIRECT. The last four columns indicated whether the OTU appears in one of the original/iDIRECT-processed networks under warming/control or not. Among the newly identified keystone species, OTU_15 belongs to the genus *Sphingomonas*, which is metabolically versatile and can utilize a wide range of naturally occurring compounds (18); OTU_38 belongs to the genus *Nitrospira* and is capable of aerobic hydrogen oxidation (19) and nitrite oxidation (20); OTU_134 belongs to the genus *Pedomicrobium* and has the ability to adhere strongly to surfaces and form biofilm (21); OTU_443 belongs to the genus *Gemmatimonadetes* and is very common in soil with an adaptation to low soil moisture (22).

								iDIRECT		Original	
ID	**Domain**	**Phylum**	**Class**	**Order**	**Family**	**Genus**	**Role**	**Warming**	**Control**	**Warming**	**Control**
OTU_15	Bacteria	Proteobacteria	Alphaproteobacteria	Sphingomonadales	Sphingomonadaceae	Sphingomonas	Net Hub	Yes			
OTU_14	Bacteria	Acidobacteria	Acidobacteria_Gp1	Unclassified	Unclassified	Gp1	Mod Hub	Yes	Yes	Yes	
OTU_21	Bacteria	Acidobacteria	Acidobacteria_Gp1	Unclassified	Unclassified	Gp1	Mod Hub		Yes		Yes
OTU_35	Bacteria	Proteobacteria	Alphaproteobacteria	Sphingomonadales	Sphingomonadaceae	Sphingomonas	Mod Hub		Yes		Yes
OTU_102	Bacteria	Acidobacteria	Acidobacteria_Gp1	Unclassified	Unclassified	Gp1	Mod Hub		Yes	Yes	Yes
OTU_117	Bacteria	Proteobacteria	Alphaproteobacteria	Rhizobiales	Hyphomicrobiaceae	Devosia	Mod Hub		Yes		Yes
OTU_131	Bacteria	Actinobacteria	Actinobacteria	Solirubrobacterales	Solirubrobacteraceae	Solirubrobacter	Mod Hub	Yes	Yes	Yes	
OTU_141	Bacteria	Proteobacteria	Alphaproteobacteria	Sphingomonadales	Sphingomonadaceae	Sphingomonas	Mod Hub	Yes	Yes	Yes	Yes
OTU_176	Bacteria	Verrucomicrobia	Subdivision3	Unclassified	Unclassified	Subdivision3	Mod Hub		Yes		
OTU_4	Bacteria	Proteobacteria	Alphaproteobacteria	Sphingomonadales	Sphingomonadaceae	Sphingosinicella	Mod Hub	Yes			
OTU_31	Bacteria	Acidobacteria	Acidobacteria_Gp6	Unclassified	Unclassified	Gp6	Mod Hub	Yes		Yes	
OTU_38	Bacteria	Nitrospira	Nitrospira	Nitrospirales	Nitrospiraceae	Nitrospira	Mod Hub	Yes			
OTU_53	Bacteria	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified	Mod Hub	Yes		Yes	
OTU_81	Bacteria	Actinobacteria	Actinobacteria	Solirubrobacterales	Solirubrobacteraceae	Solirubrobacter	Mod Hub	Yes		Yes	
OTU_115	Bacteria	Proteobacteria	Deltaproteobacteria	Unclassified	Unclassified	Unclassified	Mod Hub	Yes		Yes	
OTU_134	Bacteria	Proteobacteria	Alphaproteobacteria	Rhizobiales	Hyphomicrobiaceae	Pedomicrobium	Mod Hub	Yes			
OTU_443	Bacteria	Gemmatimonadetes	Gemmatimonadetes	Gemmatimonadales	Gemmatimonadaceae	Gemmatimonas	Mod Hub	Yes			
OTU_761	Bacteria	Proteobacteria	Deltaproteobacteria	Unclassified	Unclassified	Unclassified	Mod Hub	Yes			
OTU_799	Bacteria	Proteobacteria	Alphaproteobacteria	Rhodospirillales	Rhodospirillaceae	Unclassified	Mod Hub	Yes		Yes	
OTU_892	Bacteria	Proteobacteria	Alphaproteobacteria	Rhodospirillales	Rhodospirillaceae	Skermanella	Mod Hub	Yes		Yes	
OTU_940	Bacteria	Acidobacteria	Acidobacteria_Gp1	Unclassified	Unclassified	Gp1	Mod Hub	Yes			
OTU_1513	Bacteria	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified	Mod Hub	Yes		Yes	
OTU_60	Bacteria	Actinobacteria	Actinobacteria	Actinomycetales	Unclassified	Unclassified	Mod Hub			Yes	

OTU_112	Bacteria	Actinobacteria	Actinobacteria	Actinomycetales	Micromonosporaceae	Micromonospora	Mod Hub			Yes
OTU_121	Bacteria	Proteobacteria	Alphaproteobacteria	Rhizobiales	Methylobacteriaceae	Microvirga	Mod Hub			Yes
OTU_3634	Bacteria	Proteobacteria	Alphaproteobacteria	Rhizobiales	Xanthobacteraceae	Pseudolabrys	Mod Hub			Yes
OTU_7456	Bacteria	Acidobacteria	Acidobacteria_Gp6	Unclassified	Unclassified	Gp6	Mod Hub			
OTU_119	Bacteria	Firmicutes	Bacilli	Bacillales	Paenibacillaceae 2	Oxalophagus	Connector	Yes		
OTU_132	Bacteria	Acidobacteria	Acidobacteria_Gp3	Unclassified	Unclassified	Gp3	Connector	Yes		
OTU_1382	Bacteria	Proteobacteria	Alphaproteobacteria	Rhizobiales	Unclassified	Unclassified	Connector	Yes		
OTU_21522	Bacteria	Actinobacteria	Actinobacteria	Solirubrobacterales	Solirubrobacteraceae	Solirubrobacter	Connector	Yes		
OTU_22728	Bacteria	Proteobacteria	Betaproteobacteria	Unclassified	Unclassified	Unclassified	Connector	Yes		
OTU_11276	Bacteria	Acidobacteria	Acidobacteria_Gp2	Unclassified	Unclassified	Gp2	Connector			Yes

908
909

Table S5. Topological properties of the iDIRECT-processed networks under warming and control. The standard deviations of topological properties from the random networks are used for the Student *t* test of their statistical significance between iDIRECT-processed networks.

		iDIRECT		
		Warming	**Control**	**p**
Empirical	Total nodes	432	250	---
	Total links	1139	399	---
	Average connectivity	5.273	3.192	---
	Connectance	0.01223	0.01282	---
	R2 of power law	0.886	0.937	---
	Average vulnerability	0.003272	0.004334	1.7×10^{-75}
	Average clustering coefficient (avgCC)	0.265	0.263	0.03383
	Average path distance (GD)	6.268	5.145	1.2×10^{-153}
	Module #	33	29	2.7×10^{-29}
	Relative modularity	0.9219	0.3565	2.6×10^{-40}
Random networks	Average clustering coefficient (avgCC)	0.049±0.006[a]	0.024±0.007	9.4×10^{-68}
	Average path distance (GD)	3.483±0.032	4.172±0.070	1.6×10^{-124}
	Modularity	0.397±0.005	0.574±0.008	2.2×10^{-200}

[a] mean value of topological properties followed by standard deviations from 100 simulations.

Table S6. Module preservation between warming and control networks. All module pairs with p-value < 0.01 from the Fisher's exact test are listed in an ascending order per p-value.

	Warming		Control		Shared node #	Fisher's exact test p-value
	Module	Size	Module	Size		
Pair 1	W4	20	C4	22	19	1.50×10^{-30}
Pair 2	W1	53	C1	60	29	7.51×10^{-17}
Pair 3	W2	16	C2	32	12	5.34×10^{-13}
Pair 4	W8	8	C1	60	7	1.49×10^{-6}
Pair 5	W5	93	C5	13	9	4.99×10^{-5}
Pair 6	W3	89	C3	46	18	1.30×10^{-4}
Pair 7	W3	89	C6	6	5	0.00070
Pair 8	W5	93	C7	6	5	0.00086
Pair 9	W9	7	C3	46	4	0.00151

Table S7. Commonly used copulas and their corresponding binary operators ⊗ and ⊕. $C(u,v)$ is an Archimedean copula under consideration, $\psi(t)$ is the corresponding generator function, and ⊗ and ⊕ are resulting binary operators for sequential and parallel paths, respectively.

$C(u,v)$	$\psi(t)$	$u \otimes v$	$u \oplus v$	Notes
uv	$-\ln(t)$	uv	$u+v-uv$	Independent copula
$\max\{u+v-1,\, 0\}$	$1-t$	$\max\{u+v-1,\, 0\}$	$\min\{u+v,\, 1\}$	Lower Fréchet-Hoeffding bound
$\min\{u,\, v\}$	---	$\min\{u,\, v\}$	$\max\{u,\, v\}$	upper Fréchet-Hoeffding bound
$\frac{uv}{1-(1-u)(1-v)}$	$\frac{1-t}{t}$	$\frac{uv}{1-(1-u)(1-v)}$	$\frac{u+v-2uv}{1-uv}$	Eqs. (C2)

Table S8. Important families of bivariate Archimedean copulas. The table shows the copulas $C_\theta(u,v)$ and their generator functions $\psi_\theta(t)$, both are parametrized by θ, as well as the range of parameter θ. For convenience, derived functions $\psi_\theta(1-t)$ and $1-\psi_\theta^{-1}(t)$ are also included in the table, which are essential in the implementation of the binary operator $\oplus$.

Family name	Bivariate copula $C_\theta(u,v)$	Range of θ	Generator $\psi_\theta(t)$	$\psi_\theta(1-t)$	$1-\psi_\theta^{-1}(t)$
Ali-Mikhail-Haq	$\frac{uv}{1-\theta(1-u)(1-v)}$	$[-1,\ 1]$	$\ln\frac{1-\theta(1-t)}{t}$	$\ln\frac{1-\theta t}{1-t}$	$\frac{e^t-1}{e^t-\theta}$
Clayton	$\left(\max\{u^{-\theta}+v^{-\theta}-1,\ 0\}\right)^{-1/\theta}$	$[-1,\ +\infty)$	$\frac{t^{-\theta}-1}{\theta}$	$\frac{(1-t)^{-\theta}-1}{\theta}$	$1-(\max\{1+\theta t,\ 0\})^{1/\theta}$
Frank	$-\frac{1}{\theta}\ln\left(1+\frac{(e^{-\theta u}-1)(e^{-\theta v}-1)}{e^{-\theta}-1}\right)$	$(-\infty,\ +\infty)$	$\ln\frac{e^{-\theta}-1}{e^{-\theta t}-1}$	$\ln\frac{1-e^{\theta}}{e^{\theta t}-e^{\theta}}$	$\frac{1}{\theta}\ln\left(e^{\theta}+\frac{1-e^{\theta}}{e^t}\right)$
Gumbel	$\exp\left(-\left((-\ln u)^\theta+(-\ln v)^\theta\right)^{1/\theta}\right)$	$[-1,\ +\infty)$	$(-\ln t)^\theta$	$\left(-\ln(1-t)\right)^\theta$	$1-\exp\left(-t^{1/\theta}\right)$
Joe	$1-\left((1-u)^\theta+(1-v)^\theta-(1-u)^\theta(1-v)^\theta\right)^{1/\theta}$	$[-1,\ +\infty)$	$-\ln\left(1-(1-t)^\theta\right)$	$-\ln\left(1-t^\theta\right)$	$\left(1-e^{-t}\right)^{1/\theta}$

Table S9. Comparison of the different realizations of ⊕ for parallel paths. All possible combinations of 0.1, 0.5, and 0.9 were used as the inputs. The red bold numbers show that the ordinary addition *u* + *v* violates the natural range [0,1] of association data.

$u \otimes v$	$u+v-uv$	$\max\{u,\ v\}$	$\frac{u+v-2uv}{1-uv}$	$u+v$
0.1⊕0.1	0.1900	0.1000	0.1818	0.2000
0.1⊕0.5	0.5500	0.5000	0.5263	0.6000
0.1⊕0.9	0.9100	0.9000	0.9011	1.0000
0.5⊕0.5	0.7500	0.5000	0.6667	1.0000
0.5⊕0.9	0.9500	0.9000	0.9091	**1.4000**
0.9⊕0.9	0.9900	0.9000	0.9474	**1.8000**

Table S10. Comparison of two assembly strategies. $u \otimes v = uv$ for sequential paths, $u \oplus v = (u+v-2uv)/(1-uv)$ for parallel paths, and three different combinations of u, v, and w are used. $(u \otimes v) \oplus (u \otimes w)$ is from the APS strategy, $u \otimes (v \oplus w)$ is from the TSP strategy, and Δ is the difference. Red bold number shows that Δ is non-zero when $u = 0.5$ and $v = w = 1$. See Fig. S16d for visualization of Case 3.

	u	v	w	$(u \otimes v) \oplus (u \otimes w)$	$u \otimes (v \oplus w)$	Δ
Case 1	0.0	0.5	0.5	0.0000	0.0000	0.0000
Case 2	1.0	0.5	0.5	0.6667	0.6667	0.0000
Case 3	0.5	1.0	1.0	0.6667	0.5000	**0.1667**

SI References

1. S. Feizi, D. Marbach, M. Médard, & M. Kellis, Network deconvolution as a general method to distinguish direct dependencies in networks. *Nature biotechnology.* **31**(8), 726 (2013).
2. B. Barzel & A.-L. Barabási, Network link prediction by global silencing of indirect correlations. *Nature biotechnology.* **31**(8), 720 (2013).
3. Z. D. Kurtz*, et al.*, Sparse and compositionally robust inference of microbial ecological networks. *PLoS computational biology.* **11**(5), e1004226 (2015).
4. P. Bastiaens*, et al.*, Silence on the relevant literature and errors in implementation. *Nature biotechnology.* **33**(4), 336 (2015).
5. B. N. Kholodenko*, et al.*, Untangling the wires: a strategy to trace functional interactions in signaling and gene networks. *Proceedings of the National Academy of Sciences.* **99**(20), 12841-12846 (2002).
6. A. A. Margolin*, et al.*, ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics.* **7**(1), S7 (2006).
7. B. Alipanahi & B. J. Frey, Network cleanup. *Nature biotechnology.* **31**(8), 714 (2013).
8. J. Stoer & R. Bulirsch, "*Introduction to numerical analysis*". (Springer-Verlag, New York, 2002); trans Gautschi W, Witzgall C, & Bartels R 3 Ed.
9. L. El Ghaoui, Inversion error, condition number, and approximate inverses of uncertain matrices. *Linear algebra and its applications.* **343**, 171-193 (2002).

10. D. Marbach, *et al.*, Revealing strengths and weaknesses of methods for gene network inference. *Proceedings of the National Academy of Sciences.* **107**(14), 6286-6291 (2010).
11. R. De Smet & K. Marchal, Advantages and limitations of current network inference methods. *Nature Reviews Microbiology.* **8**(10), 717 (2010).
12. A. Tarantola, "*Inverse problem theory and methods for model parameter estimation*". (SIAM, 2005).
13. G. H. Tucci & K. Wang, New methods for handling singular sample covariance matrices. Preprint at https://arxiv.org/abs/1111.0235. 2011).
14. R. B. Nelsen, "*An Introduction to Copulas*". (Springer-Verlag, New York, 2006) 2 Ed.
15. R. B. Nelsen, Dependence and order in families of Archimedean copulas. *Journal of Multivariate Analysis.* **60**(1), 111-122 (1997).
16. M. M. Ali, N. Mikhail, & M. S. Haq, A class of bivariate distributions including the bivariate logistic. *Journal of multivariate analysis.* **8**(3), 405-412 (1978).
17. D. G. Clayton, A model for association in bivariate life tables and its application in epidemiological studies of familial tendency in chronic disease incidence. *Biometrika.* **65**(1), 141-151 (1978).
18. D. L. Balkwill, J. K. Fredrickson, & M. F. Romine, Sphingomonas and related genera. *Pacific Northwest National Lab.(PNNL), Richland, WA (United States),* (2003).
19. H. Koch, *et al.*, Growth of nitrite-oxidizing bacteria by aerobic hydrogen oxidation. *Science.* **345**(6200), 1052-1054 (2014).
20. H. Koch, *et al.*, Expanded metabolic versatility of ubiquitous nitrite-oxidizing bacteria from the genus Nitrospira. *Proceedings of the National Academy of Sciences.* **112**(36), 11371-11376 (2015).

21. L. Sly, M. Hodgkinson, & V. Arunpairojana, Effect of water velocity on the early development of manganese-depositing biofilm in a drinking-water distribution system. *FEMS Microbiol. Ecol.* **4**(3-4), 175-186 (1988).

22. J. M. DeBruyn, L. T. Nixon, M. N. Fawaz, A. M. Johnson, & M. Radosevich, Global biogeography and quantitative seasonal dynamics of Gemmatimonadetes in soil. *Applied and environmental microbiology.* **77**(17), 6295-6300 (2011).