Detecting Evolving Patterns of Self-Organizing Networks by Flow Hierarchy Measurement

Jianxi Luo* and Christopher L. Magee

Engineering Systems Division
Massachusetts Institute of Technology
77 Massachusetts Avenue, E38-450, Cambridge, Massachusetts 02139, USA

* To whom correspondence should be addressed. E-mail: luo@mit.edu. Phone: 1-617-642-1652

Supplementary Material

Appendix A: Assessments of alternative hierarchy metrics and algorithms

Besides the hierarchy metric introduced in the main paper, we also explored other possible metrics that aim to quantify the degree to which the system architecture follows a flow hierarchy. The metrics are compared and it is shown that the one proposed in the paper (main text) has advantages over the others in accuracy and ease of use.

A.1 Alternative Metric Base upon Cycle Identification

The first alternative hierarchy algorithm/metric to examine is to count the portion of nodes (instead of links) which are not included in any cycle over the total nodes. Proceeding in a way similar to the approach in the text of the main paper, we construct the node adjacency matrix first, and then raise the power of matrices to derive the node distance matrix. With the node distance matrix, we can check whether a node is involved in any cycle. One obvious disadvantage, compared to the one proposed in the paper is that, it neglects the layered hierarchy in its relative metric system. For example, in the example layered hierarchy network in Figure 2, using this algorithm, all the 6 nodes are included in at least one cycle, so the hierarchy degree is zero. However, there is obviously an existing layered hierarchy. Instead, the hierarchy metric which we propose in the main text and count links appropriately can identify the hierarchical link *d* in Figure 5 of the main text.

A.2 Alternative Metrics Based upon Level Identification

Both of the two approaches discussed above do not require ranking the nodes, but search for cyclic phenomena embedded in directed networks. Now, we examine the feasibility of other alternative ways to measure hierarchy, which are based on identifying the nonhierarchical links when a specific logic of ordering for the hierarchy is specified. The logic of ordering can be based on network structure or domain-specific characteristics.

When nodes are pre-assigned level ranks, the links from a predefined lower level to its adjacent higher level are regarded as hierarchical. Moreover, the links that skip levels and the links between nodes on the same level can also be accepted as hierarchical. However, when a link connects from a pre-indentified higher level backward to a lower one, it violates the fundamental

assumption that, in a pure flow hierarchy all flows/links follow one general direction, so it is non-hierarchical.

Nonetheless, the identification of such link types is somewhat arbitrary because it depends on the pre-assigned level ranks to nodes, which are often ambiguous. In many cases, there is no objective and definitive criterion according to which a node must be on a specific level, though experts with domain knowledge can give a level rank to a node based on their domain knowledge and subjective judgment. Such rank-assigning work based on domain knowledge is a usual practice in food web research [S1] and industrial system research [S2]. Measures based upon such assignments of ranks thus have a partially arbitrary character.

In order to avoid arbitrary ranking, we explore several practical ways of assigning level ranks to each node in a directed network, using differently the information of the network positions of nodes in a directed network. Then, we assess their feasibility and accuracy for indentifying flow hierarchies. Our ranking algorithms first identify the sinks, which have no outgoing links but only incoming links, and then use the path lengths from the other nodes to sinks as the basis of assigning a level rank. Here, path length means the number of intermediate links on a path from a node to a sink of interest (A path is a walk in which all nodes and all lines are distinct; a walk is a sequence of nodes and lines, starting and ending with nodes, in which each node is incident with the lines following and preceding it in the sequence [S3]).

In this way the sinks are used as the benchmarking boundary. Alternatively, the sources, which have no incoming links but only outgoing links, can also be used as the benchmarking boundary.

In the following section, we will only show the use of sinks as the benchmarking boundary as the analysis of sources is directly analogous. Because there is usually more than one path from nodes to a sink, and there are usually more than one sink on the top bound of the industry, five different algorithms are discussed. These algorithms are abstracted to different aspects of the relative network positions of nodes.

1) Min [Shortest]: A node's level rank is given as the shortest one among its all shortest paths to all the sinks.

$$LR_i = \min\left(\min_j\left(D_{ij}\right)\right) \quad i \in [nodes],\ j \in [sinks] \qquad \text{[S1]}$$

LR_i : the level rank of node i;

D_{ij} : the set of lengths of the paths from node i to sink j.

2) Max [Shortest]: A node's level rank is given as the longest one among its all shortest paths to all the sinks.

$$LR_i = \max\left(\min_j\left(D_{ij}\right)\right) \quad i \in [nodes],\ j \in [\text{sinks}] \qquad \text{[S2]}$$

3) Min [Longest]: A node's level rank is given as the shortest one among its all longest paths to all the sinks.

$$LR_i = \min\left(\max_j\left(D_{ij}\right)\right) \quad i \in [nodes],\ j \in [sinks] \qquad \text{[S3]}$$

4) Max [Longest]: A node's level rank is given as the longest one among its all longest paths to all the sinks.

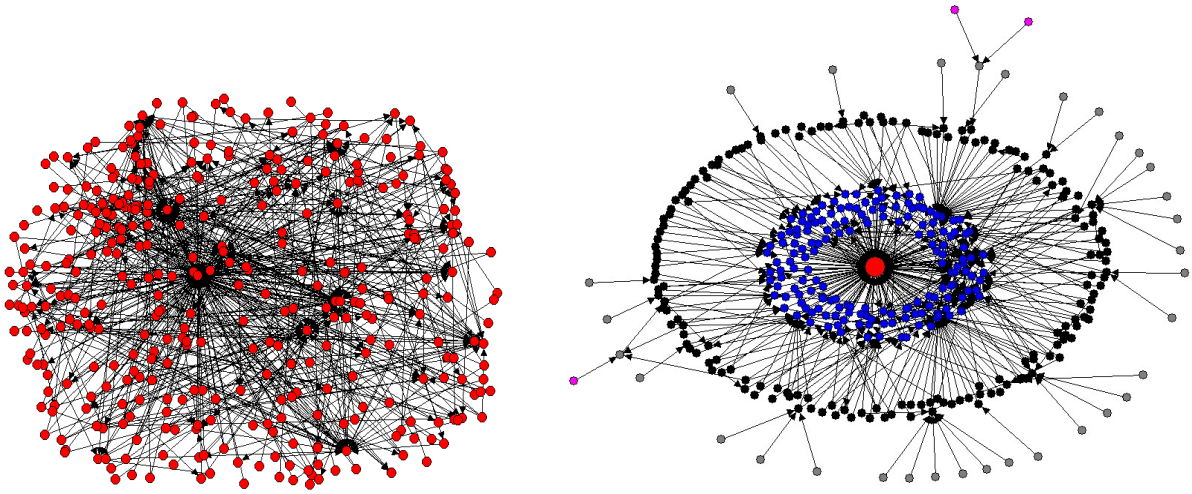
$$LR_i = \max\left(\max_j\left(D_{ij}\right)\right) \quad i \in [nodes],\ j \in [sinks] \qquad \text{[S4]}$$

5) Continuous Level Rank (Average)

$$LR_i = \underset{j}{average}\,(D_{ij}) \qquad i \in [nodes],\ j \in [\text{sinks}] \qquad [S5]$$

Note: when there is only a single sink in the network, Max [Shortest] and Min [Shortest] become the same, and Max [Longest] and Min [Longest] become the same.

The first four algorithms above tend to group the nodes into discrete levels. The fifth algorithm is different because it assigns continuous level ranks. Figure S1 shows the example of the network of Toyota Motor Company's suppliers before (left) and after (right) being grouped into levels according to the Max [Shortest] algorithm. The nodes (i.e. companies) are arranged in space (using UCINET [S4]) to illustrate the underlying flow hierarchy. This network exhibits strong hierarchy, found by the visualization based upon the arbitrary ranking/grouping result.



A) Before grouping B) After grouping

Figure S1. Networks of Toyota's suppliers (A) before grouping and (B) after grouping based upon the *Max [Shortest]* algorithm. The network contains the Japanese suppliers either directly or indirectly connected to Toyota Motor Company by the flows of transacted components and parts. The network is extracted from the data book [S5] used for the calculation of Japanese automotive production network in Table 2 of the main text. The network here includes 372 nodes (i.e., manufacturing firms) and 591 links

(i.e. supplier-customer transactional relationships). For instance, if company A sells a product to company B, there is an arrow from A to B in the network.

Regardless of which method is used and whether it is arbitrary, after each node is assigned a unique level rank, i.e. grouped into a specific level, we can identify if a local flow/link is from a lower level to a higher or the same level (hierarchical) or from a higher level to a lower level (non-hierarchical). More specifically, we differentiate all the links of a network into four different types (also demonstrated in the examples in Figure S2):

1) Regular: the link connects from a node on a pre-defined lower level (i) to a node on its adjacent higher level (i-1);
2) Level-Skipping: the link connects from a node on a pre-defined lower level (i) to a node on a level (j) higher than its adjacent higher level (i-1), i.e. $j < i$-1;
3) In-Layer: the link connects between nodes on the same level (i);
4) Backward: the link connects from a node on a predefined high level (i) to a node on a lower level (j), i.e. $i < j$.

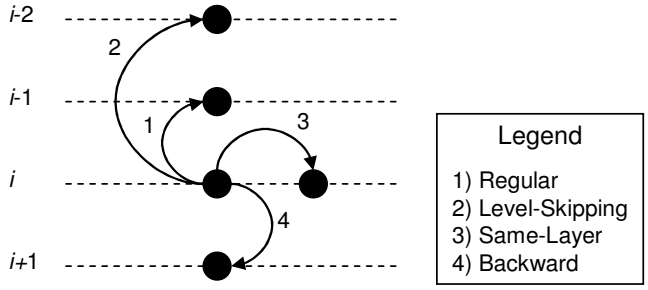


Figure S2. Examples for four types of links identified according to levels

As a matter of fact, in discussing the network examples in Figure 2 of the main text, we have noted the regular, in-layer, level-skipping, and backward links, with implicitly pre-assumed

levels. In general, the first three types are accepted as hierarchical links, although intuitively there is an order for the hierarchy degree they represent, which is:

Regular > *Level-Skipping* > *In-Layer*

The fourth type, i.e. backward link, clearly violates the fundamental assumption that, in a pure flow hierarchy all flows/links follow one general direction, so it is non-hierarchical. Now we may count the ratio of hierarchical types of links over total links as a potential hierarchy metric,

$$h = \frac{m - \sum_{i=1}^{m} e_i}{m} \tag{S6}$$

where m is the number of links in the network and e_i=1 if link i is a backward link (0 otherwise). However, because the "backward" vs. "forward" directions are relative, whether a link is backward or forward depends on the direction assumed. To make it simple, we assume that backward links are inconsistent to a system's dominant orientation, and are minor ones. Thus, at maximum only half of the links can be "backward", and the ratio calculated from formula S6 will always range between 0 and 0.5. To improve this potential metric to range between 0 and 1, we normalize it to the range of [0, 1] by multiplying 2 in equation S6 to the term which counts the backward links. Furthermore, when the same numbers of forward and backward links exist in a network, a reasonable hierarchy metric should be zero. However, in-layer links might exist so hierarchy degree is still larger than zero. To correct this and make the hierarchy degree zero when the forward and backward links are equal regardless of the in-layer links, I propose an improved formula from S7,

$$h = \frac{m - 2\times\sum_{i=1}^{m} e_i}{m} \tag{S7}$$

where m is the total number of links. e_i=1 if link i is a backward link, e_i=0.5 if link i is a in-layer link, and e_i=0 if link i is either a regular or level-skipping link.

In the Toyota network shown in Figure S1, grouping by the Max [Short] algorithm determines the ratio for each type of links: 425 links are regular links; 159 links are in-layer links; no level-skipping links; 7 links are backward links. Thus, the hierarchy degree is

$$\frac{591-2\times(159\times 0.5+7\times 1)}{591}=0.7073$$

However, such an approach may over count non-hierarchical links. Here we use a simple example network (Figure S3) of five nodes to examine the feasibility for identifying non-hierarchical links (vs. hierarchical links) based on the level ranks obtained from the five extreme algorithms introduced above. Nodes are placed on their corresponding levels given by different algorithms.

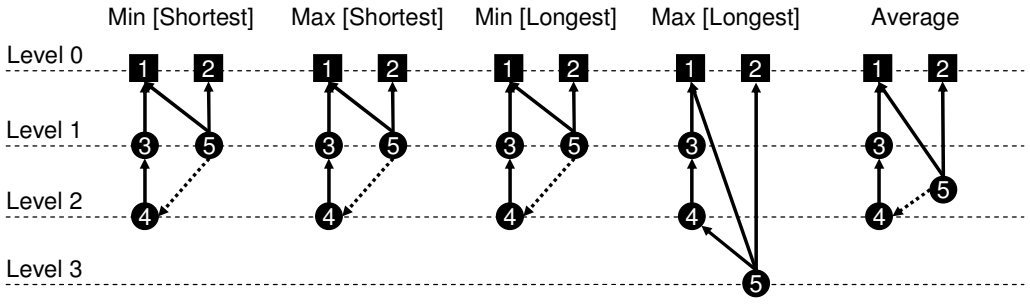


Figure S3. Identifying nonhierarchical links based on five different level ranking algorithms. Non-hierarchical backward links are dashed.

In this network, there is one source node, 5, and two sink nodes, 1 and 2. Obviously, we can observe directly that all the links in this small network do follow a holistic direction from bottom to top, so are hierarchical (h =1 using the method in our paper). However, according to the Min

[Shortest] and Max [Shortest] ranking algorithms based on counting the shortest paths, node 5 belongs to level 1, and node 4 belongs to level 2, then the link from node 5 to node 4 is a backward and nonhierarchical link. According to Min [Longest] algorithm that counts longest paths, because node 5 has its longest path to node 2 in the length of 1, it is still placed on level 1, and its link to node 4 is still a nonhierarchical one. The fifth algorithm uses the average path length to sinks as a node's level rank, then node 5 has three paths to the sinks and the average path length is 1.66. Node 4 has one path of length 2 to the sinks, so its level rank is 2. So, the link from node 5 to node 4 is again identified as a nonhierarchical one.

Only the Max [Longest] algorithm does not over count non-hierarchical links. As a matter of fact, this algorithm theoretically equates finding the layout of the dependency matrix of the directed network which minimizes the number of links above the diagonal, if we place the sinks at the left upper corner of the adjacency matrix. The other algorithms more or less ignore part of the global path information while Max [Longest] considers all the path information when it operates. In contrast, the Max [Longest] algorithm works appropriately because it has traced complete path information from the nodes to the sinks in the effort of assigning level ranks.

A.3 Hierarchy Metric based upon *Max [Longest]* Level Identification Algorithm

Therefore, we propose a second hierarchy metric based on counting the non-hierarchical links identified by the Max [Longest] level-ranking algorithm. Calculating this hierarchy metric consists of the following steps:

Step 1) Identify the sinks of the network as the benchmark. Alternatively, we can also use sources of the network as the benchmark.

Step 2) Calculate the lengths of the longest paths from each node to all the sinks, and use the longest one of these lengths as the node's level rank.

Step 3) Count the total number of the backward links. Any link, which goes from a node with higher level rank to a node with a lower level rank, is identified as a nonhierarchical link. The rest of the links are hierarchical.

Step 4) With the known information on the levels and link types, compute the hierarchical degree using formula S7.

Figure S4 lists the hierarchy degrees of several example networks based on this approach. A pure hierarchical structure, such as a tree (e.g. Figure S4A), has a hierarchy degree 1. For a pure directed cycle (e.g. Figure S4E), this approach does not give an answer because there is neither a sink nor a source node to be used as a benchmark.

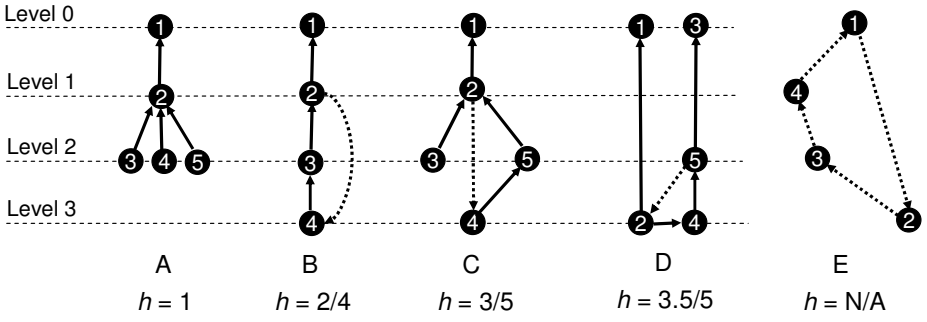


Figure S4. Hierarchy degrees of example networks based on the Max [Longest] ranking algorithm. Non-hierarchical backward links are dashed.

Similar to the hierarchy metric proposed in the paper that counts links on cycles, this alternative hierarchy metric also examines how much the intermediate or local links coherently follow a holistic direction in the directed network. However, compared with the hierarchical metric in the paper, the second metric has two disadvantages in practice. First, it requires extra steps to identify the sources or sinks. In some systems where neither sources nor sinks exist mathematically, the algorithm does not apply without arbitrarily picking the benchmark nodes. The second disadvantage is that, it is computationally hard to find the longest paths between nodes in a large network. Such calculation requires exhaustive search of paths of all possible lengths. It is doable if the network size is small enough. However, when the system size becomes big, it may take "forever" to calculate the level ranks.

Therefore, among these two hierarchy metrics and algorithms, we prefer the first hierarchy metric simply because of its ease of computation, although the second metric is also meaningful in computable cases.

Appendix B: Descriptive statistics and hierarchy degrees of historical Linux kernels

Linux kernel source codes are obtained from the official online archive of Linux Kernel Organization, Inc http://www.kernel.org/.

Table S1. Descriptive statistics and calculated results for the used data points

Version	Release Date	*N*	*L*	*K*	h_{real}	h_{rand}	σ_{rand}	*z*-score	*Modularity (directed)*	*Modularity (undirected)*
0.01	17-Sep-91	35	97	2.771	0.7010	0.1332	0.0768	7.391	0.1795	0.2876
0.11	8-Dec-91	41	128	3.122	0.5547	0.0871	0.0563	8.300	0.1432	0.2779
0.12	16-Jan-92	51	168	3.294	0.4524	0.0753	0.0459	8.208	0.2036	0.2781
0.95	8-Mar-92	50	176	3.520	0.5057	0.0581	0.0403	11.113	0.1345	0.2503
0.96a	22-May-92	60	218	3.633	0.5596	0.0528	0.0360	14.079	0.1598	0.2773

0.96b	22-Jun-92	62	227	3.661	0.5551	0.0472	0.0315	16.125	0.0644	0.2778
0.96c	5-Jul-92	69	258	3.739	0.5891	0.0457	0.0299	18.169	0.1138	0.2966
0.97	1-Aug-92	77	299	3.883	0.5886	0.0388	0.0259	21.186	0.0931	0.2879
0.99.2	1-Jan-93	150	575	3.833	0.7200	0.0423	0.0243	27.893	0.1258	0.3071
0.99.5	9-Feb-93	149	573	3.846	0.726	0.0422	0.0236	28.962	0.1135	0.3103
0.99.7	13-Mar-93	170	699	4.112	0.7668	0.0322	0.0215	34.199	0.1207	0.3130
0.99.9	24-Apr-93	171	702	4.105	0.7764	0.0316	0.0200	37.243	0.1002	0.3254
0.99.10	7-Jun-93	196	850	4.337	0.7553	0.0247	0.0185	39.413	0.1289	0.3184
0.99.11	18-Jul-93	196	859	4.383	0.7579	0.0243	0.0173	42.422	0.1223	0.3258
0.99.12	15-Aug-93	201	888	4.418	0.7635	0.0218	0.0167	44.524	0.1212	0.3324
0.99.13	20-Sep-93	203	904	4.453	0.7788	0.0216	0.0172	43.941	0.1254	0.3084
0.99.15	3-Feb-94	234	1084	4.632	0.8044	0.0174	0.0146	53.773	0.1274	0.328
1.0	13-Mar-94	235	1100	4.681	0.7673	0.0162	0.0139	53.913	0.1232	0.3662
1.1.0	6-Apr-94	234	1084	4.632	0.7703	0.0171	0.0142	53.006	0.1256	0.3209
1.1.13	23-May-94	242	1092	4.512	0.7711	0.0205	0.0163	45.992	0.1022	0.3003
1.1.23	27-Jun-94	252	1189	4.718	0.7771	0.0165	0.0144	52.638	0.1154	0.3531
1.1.29	14-Jul-94	254	1214	4.780	0.7727	0.0151	0.0134	56.689	0.0949	0.312
1.1.45	15-Aug-94	275	1293	4.702	0.7873	0.0162	0.0134	57.440	0.0864	0.3309
1.1.52	6-Oct-94	277	1315	4.747	0.7916	0.0163	0.0142	54.439	0.0967	0.3527
1.1.63	14-Nov-94	275	1293	4.702	0.7873	0.0157	0.0139	55.672	0.0864	0.3309
1.1.70	2-Dec-94	287	1385	4.826	0.8065	0.0141	0.0133	59.396	0.0549	0.3838
1.1.76	2-Jan-95	296	1546	5.223	0.8273	0.0096	0.0104	78.481	0.0326	0.321
1.1.89	5-Feb-95	333	1709	5.132	0.8660	0.0106	0.0114	74.773	0.0298	0.3576
1.2.0	7-Mar-95	334	1738	5.204	0.8452	0.0101	0.0110	76.072	0.047	0.3576
1.2.3	2-Apr-95	334	1739	5.207	0.8436	0.0101	0.0110	75.942	0.0465	0.3571
1.2.8	3-May-95	334	1740	5.210	0.8437	0.0094	0.0108	77.329	0.043	0.3559
1.3.0	12-Jun-95	344	1898	5.517	0.8583	0.0073	0.0094	90.217	0.0376	0.3401
1.3.7	6-Jul-95	382	2108	5.518	0.8667	0.0074	0.0091	94.892	0.0496	0.3276
1.3.15	2-Aug-95	384	2140	5.573	0.8673	0.0063	0.0086	100.121	0.046	0.3288
1.3.22	1-Sep-95	384	2170	5.651	0.8659	0.0056	0.0081	106.810	0.0443	0.3279
1.3.31	4-Oct-95	390	2248	5.764	0.8674	0.0050	0.0075	114.472	0.0477	0.3295
1.3.38	7-Nov-95	406	2301	5.667	0.8609	0.0057	0.0085	100.725	0.0526	0.3242
1.3.46	11-Dec-95	438	2585	5.902	0.8286	0.0045	0.0070	117.154	0.0449	0.3324
1.3.53	2-Jan-96	457	2647	5.792	0.8342	0.0048	0.0073	113.011	0.0202	0.3031
1.3.60	7-Feb-96	482	2776	5.759	0.835	0.0054	0.0079	104.383	0.049	0.3289
1.3.70	1-Mar-96	514	3010	5.856	0.8432	0.0048	0.0075	111.167	0.0216	0.319
1.3.82	2-Apr-96	554	3355	6.056	0.8393	0.0040	0.0066	126.854	0.0261	0.319
1.3.98	4-May-96	646	3952	6.118	0.8335	0.0034	0.0063	132.816	0.0289	0.3145
2.0	9-Jun-96	661	4055	6.135	0.8486	0.0037	0.0067	125.871	0.0269	0.3259
2.0.5	10-Jul-96	661	4070	6.157	0.8511	0.0036	0.0067	126.504	0.0261	0.3172
2.0.13	16-Aug-96	663	4084	6.160	0.8511	0.0032	0.0058	146.264	0.0253	0.3157
2.1	30-Sep-96	668	4100	6.138	0.8515	0.0036	0.0066	128.125	0.0261	0.3088
2.1.6	29-Oct-96	663	3955	5.965	0.8516	0.0043	0.0072	117.095	0.038	0.3033
2.1.13	23-Nov-96	704	4258	6.048	0.8422	0.0039	0.0065	128.103	0.0368	0.3389
2.1.16	18-Dec-96	743	4579	6.163	0.8574	0.0033	0.0063	136.632	0.0204	0.2939
2.1.20	2-Jan-97	757	4709	6.221	0.8609	0.0031	0.0060	142.333	0.0166	0.2835

2.1.25	2-Feb-97	775	4893	6.314	0.8580	0.0030	0.0057	151.306	0.019	0.2918
2.1.30	26-Mar-97	823	5444	6.615	0.8720	0.0023	0.0052	166.595	0.0252	0.2815
2.1.36	23-Apr-97	880	5833	6.628	0.8838	0.0019	0.0047	188.563	0.0268	0.2588
2.1.40	22-May-97	871	5776	6.631	0.8776	0.0021	0.0051	173.238	0.0276	0.2456
2.1.43	16-Jun-97	883	5807	6.576	0.8776	0.0024	0.0054	160.998	0.0279	0.2644
2.1.45	17-Jul-97	945	6177	6.537	0.8844	0.0024	0.0053	166.812	0.0321	0.2604
2.1.50	14-Aug-97	972	6376	6.560	0.8912	0.0022	0.0049	182.892	0.0318	0.2648
2.1.56	20-Sep-97	1014	6615	6.524	0.8698	0.0021	0.0049	175.799	0.0334	0.2656
2.1.60	25-Oct-97	1044	6672	6.391	0.8698	0.0029	0.0061	142.437	0.0304	0.269
2.1.65	18-Nov-97	1053	6776	6.435	0.8669	0.0025	0.0054	161.265	0.0304	0.2587
2.1.75	22-Dec-97	1152	7343	6.374	0.8501	0.0025	0.0053	160.780	0.0366	0.2915
2.1.80	21-Jan-98	1279	7990	6.247	0.8831	0.0034	0.0062	141.106	0.0491	0.2933
2.1.88	21-Feb-98	1316	8205	6.235	0.8851	0.0031	0.0060	145.903	0.0458	0.2413
2.1.90	18-Mar-98	1321	8227	6.228	0.8922	0.0036	0.0066	134.715	0.0468	0.2441
2.1.97	18-Apr-98	1389	8725	6.281	0.8889	0.0029	0.0058	153.434	0.0783	0.2535
2.1.103	21-May-98	1441	9131	6.337	0.8938	0.0029	0.0059	151.729	0.0743	0.2841
2.1.105	7-Jun-98	1470	9323	6.342	0.8985	0.0028	0.0055	162.692	0.0705	0.2114
2.1.109	17-Jul-98	1476	9383	6.357	0.8995	0.0029	0.0058	154.547	0.0738	0.1974
2.1.116	19-Aug-98	1502	9528	6.344	0.8969	0.0029	0.0058	154.768	0.0732	0.2873
2.1.122	16-Sep-98	1516	9661	6.373	0.8970	0.0028	0.0059	151.750	0.0718	0.1937
2.1.126	23-Oct-98	1550	9905	6.390	0.8900	0.0026	0.0057	156.533	0.0832	0.2796
2.1.129	19-Nov-98	1559	9978	6.400	0.8931	0.0026	0.0055	162.222	0.0837	0.2833
2.1.132	22-Dec-98	1615	10413	6.448	0.8939	0.0024	0.0053	167.164	0.0655	0.282
2.2	26-Jan-99	1663	10811	6.501	0.9053	0.0023	0.0052	173.143	0.057	0.2635
2.2.2	22-Feb-99	1663	10826	6.510	0.9049	0.0021	0.0050	181.488	0.0562	0.2841
2.2.4	23-Mar-99	1661	11040	6.647	0.9027	0.0022	0.0053	170.552	0.0591	0.2248
2.2.6	16-Apr-99	1663	11091	6.669	0.9042	0.0021	0.0051	178.270	0.0589	0.2108
2.3	11-May-99	1695	11315	6.676	0.9088	0.0019	0.0050	182.177	0.0581	0.2333

1,000 randomly-generated comparable networks are used to calculate h_{rand} and *z-score* for each data point. Because hierarchy degrees of random networks do not vary significantly with the increases of N when $k>2$ and $N>80$ (see Fig.3 in the main text), to reduce computation efforts we use randomly-generated networks with a constant N (=100) and corresponding k to predict h_{rand} and *z-scores* for most of the data points with large N, except the earliest 8 ones with less than 100 nodes. For the earliest 8 data points, the random networks have the same N and L of their corresponding actual networks.

Supplementary References

S1. Dunne, J.; Williams, R.; Martinez, N.; Woods, R.; Erwin, D. Compilation and network analyses of Cambrian food webs. PLoS Biology 2008, 6, 693-708.

S2. Dalziel, M. A systems-based approach to industry classification. Research Policy 2007, 36, 1559-1574

S3. Wasserman, S.; Faust, K. Social Network Analysis; Cambridge University Press: Cambridge, 1994.

S4. UCINET network analysis and visualization package is available at http://www.analytictech.com/.

S5. Dodwell Marketing Consultants. The Structure of Japanese Auto Parts Industry; Tokyo, 1993。