MIT ESD

Massachusetts Institute of Technology
Engineering Systems Division

ESD Working Paper Series

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

Jianxi Luo
Engineering Systems Division
Massachusetts Institute of Technology
luo@mit.edu

Christopher L. Magee
Engineering Systems Division
Massachusetts Institute of Technology
cmagee@mit.edu

ESD-WP-2010-01 June 2010 esd.mit.edu/wps

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

Abstract

Hierarchies occur widely in evolving self-organizing ecological, biological, technological and social networks, but detecting and comparing hierarchies is difficult. Here we present a metric and technique to quantitatively assess the extent to which self-organizing directed networks exhibit a flow hierarchy. Flow hierarchy is a commonly observed but theoretically overlooked form of hierarchy in networks. We show that the ecological, neurobiological, economic and information processing networks are generally more hierarchical than their comparable random networks. We further discovered that hierarchy degree has increased over the course of the evolution of Linux kernels, confirming an early hypothesis by Herbert Simon on the emergence of hierarchy in evolutionary processes. Taken together, our results suggest that hierarchy is a central organizing feature of real-world evolving networks, and the measurement of hierarchy opens the way to understand the structural regimes and evolutionary patterns of self-organizing networks. Our measurement technique makes it possible to objectively compare hierarchies of different networks and of different evolutionary stages of a single network, and compare evolving patterns of different networks. It can be applied to various complex systems, which can be represented as directed networks.

Keywords: self-organizing networks | evolution pattern | flow hierarchy

Introduction

Complex systems of various kinds (social, biological, physical, technological, etc) frequently take the form of hierarchy [1,2]. On one hand, hierarchy is one of the central structural schemes that an architect may use to manage complexities. Products, organizations and other artifacts are often designed and managed hierarchically. On the other hand, hierarchies emerge and occur widely in self-organizing and evolutionary systems, such as food webs (ecological), neural networks (biological), open-source software (technological), and industrial production network (economic), etc., which have no architect. In such cases, hierarchy is viewed as a natural emergent phenomenon and the consequence of evolutionary processes [2,3].

In complex self-organizing networks, hierarchy, like the well-studied "small world" phenomenon [4] and the power law of degree sequence [5,6], is a global feature shared by various kinds of network systems (e.g. ecological, biological, social and technological) [7,8,9]. It is important to understand the hierarchy in self-organizing networks, because as emergence it may reflect important information on the functional needs of or constraints on the entities and their relationships which collectively form the network. However, detecting and comparing hierarchies is difficult in real-world networks, largely because first there are many types of hierarchy, and secondly hierarchy usually appears in impure forms in them [10,11].

Hierarchy is a generic structure, in which levels are asymmetrically ranked according to a specific type of relation. The ordering of levels, i.e. the rule of asymmetry, determines a

hierarchy. Scholars interested in complex systems [1,2,3,10] have paid attention to various types of relations existing between the elements that may determine a hierarchy and have described as many as four types of hierarchy in general [12]. By the logic construct for why an upper level is above a lower one, two types of hierarchies are useful for understanding the more specific case of network architectures: containment hierarchy and flow hierarchy.

A containment hierarchy is similar to the concepts of "nested hierarchy" [1,10,13] or "inclusion hierarchy" [12,14], in which nodes are divided into groups that are further divided into subgroups of groups and so on over multiple levels. Containment hierarchy can be represented as a pure tree or dendrogram [11,15], in which nodes that are closely connected [9,11,15,16,17], or have close equivalence measures [15,18,19], share lower common ancestors than more distantly connected or distinctly positioned nodes. A containment hierarchy can be found for both directed and undirected complex real-world networks.

Flow hierarchy is only associated with directed networks but is observed in many evolving self-organizing networks such as food webs, neural networks, information processing networks and industrial production networks. In many of these cases, the containment ordering criterion does not apply and the order of levels is essentially determined by the direction of the flows of resources essential to the network. Such flows are crucial because they provide necessary resources, for the entities to produce, reproduce, sustain (or remain in useful or necessary existence) and prosper. Via being connected by flows, the entities in such self-organizing systems co-evolve and may self-organize into a flow hierarchy. For example, in food webs, it is energy that flows. In software networks, it is information that flows as subroutines feed parent

routines. In the production network of firms, flow hierarchy arises when there is "persistent directionality in continuing flows of intermediate goods" [20] and flows of payments in a reverse direction. In production economy, firms co-evolve in networks of flows.

Imperfect Flow Hierarchy in Networks

Much of the recent interesting work [11,15,16,21] on hierarchy in complex networks has been devoted to containment hierarchies. Although flow hierarchy also frequently occurs in various kinds of systems, it has been largely ignored. This paper aims to promote awareness of flow hierarchy as an emergent property of complex self-organizing networks and as a lens to study and deepen our understanding on such networks.

The value of interpreting systems as flow hierarchies has not been fully exploited, partly because flow hierarchies usually do not appear in a pure form in complex self-organizing networks, such as food webs, neural networks, etc. Ideally, given a criterion used to link levels above and below, the links from a predefined lower level to its adjacent higher level are regarded as hierarchical. But we often observe links that skip levels, that connect between nodes on the same level, and that go in the backward direction. With all these irregularities aggregated in large complex networks, as well as the arbitrary nature of link type identification based on level assignment, flow hierarchies may become ambiguous and intractable. Figure 1 demonstrates several simple example networks which embed and exhibit flow hierarchy to varied degrees.

Figure 1A is a pure tree. Each node is assigned not only a rank, but a single link to a higher up node. In Figure 1B, some nodes have multiple inbound and outbound links. We call it a "mixed tree hierarchy". Both the pure tree and the mixed tree are strictly hierarchical because all the links regularly connect from a lower level to an adjacent higher level. In the network C in Figure 1, levels can no longer be uniquely defined. If node 2 and 5 are defined to be in the same layer, the link from node 5 to 2 can be viewed as an "in-layer link" and the link from 5 to 1 is a "regular link". However, if node 2 is pre-defined to one level higher than node 5, then the link from 5 to 1 is a "level-skipping link". Identification of level-skipping links and in-layer links relies on the pre-identification of levels. In this case, the levels are not uniquely defined [22]. But at least all the links in Figure 1C follow a general asymmetrical direction, so this network can be regarded as hierarchical. In cases A, B, and C, there is strict asymmetric ordering of relationships.

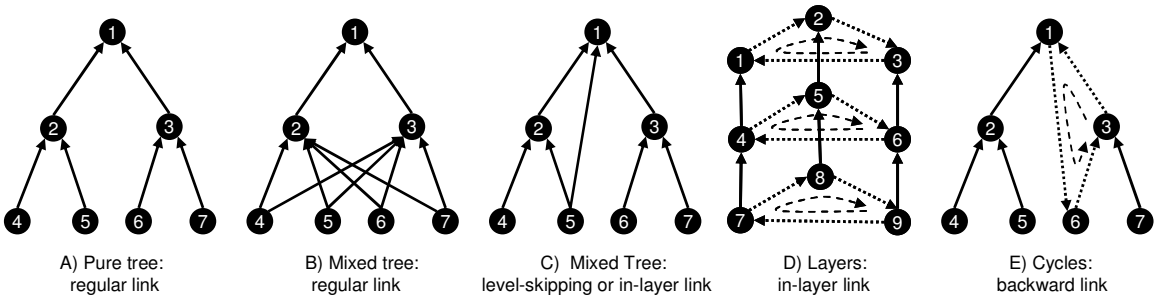


Figure 1 Example networks. The dashed links are involved in cycles.

Networks often exhibit layered structures [23], i.e. level hierarchy [12], as shown in Figure 1D. In this example, the links in cycles are symmetrical to each other and lose their global direction to some extent. However, if the nodes in the same directed cycle are presumed to be in a layer (then the links are "in-layer links"), the other links proceed in one direction from layer to layer. Thus, the network D in Figure 1 is not purely hierarchical but still has certain degree of hierarchy.

The example in Figure 1E simply shows how the emergence of a cycle may destroy the overall direction or asymmetry of a network. The examples in Figure 1 together indicate that cycles violate the directionality of a network, i.e. the asymmetry in flows, which is the fundamental principle of flow hierarchy (i.e. things move in one general direction).

The networks in Figure 1 are simple, so we can intuitively observe and sense the different degrees of hierarchy embedded in them. When given more complex and larger networks, the identification of flow hierarchy can be difficult. Figure 2 visualizes two random networks with the same numbers of nodes (100) and links (400), but vastly different degrees of hierarchy embedded. It is not surprising but important that such visualization while useful does not allow one to objectively see significant differences in hierarchy between different networks. Our technique introduced in next paragraph will reveal the large difference in hierarchy between the two networks in Figure 2.

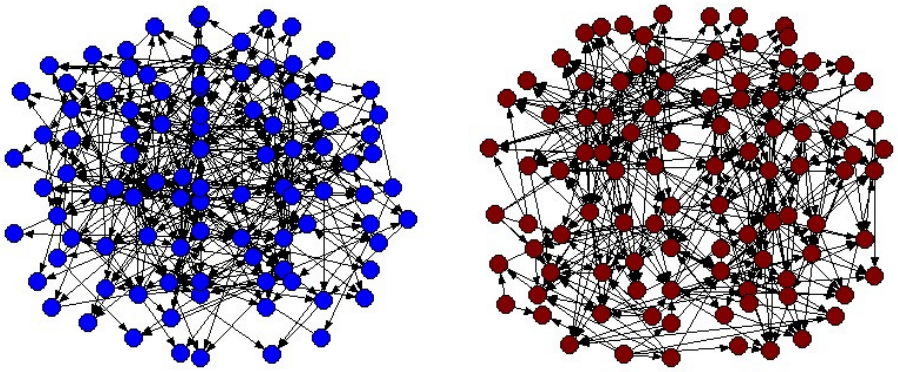


Figure 2 Random networks with the same size (*N*=100, *L*=400) but different hierarchy degrees. *N* is the number of nodes, *L* is the number of links.

The Measurement of Flow Hierarchy

Centered on the concept of flow hierarchy and its core principle – network directionality, we present a hierarchy metric that detects and measures the extent to which all the local flows follow a holistic overall "underlying direction". The hierarchy metric is calculated as the percentage of links that retain their overall direction in the network, i.e., the percentage of links that are not included in any cycle,

$$h = \frac{\sum_{i=1}^{L} e_i}{L} \quad [1]$$

where L is the number of links in the network and e_i=0 if link i is in a cycle (1 otherwise). In weighted networks, the metric can be calculated as the ratio of the weights of the links which are not included in any cycles over the total weight of all links,

$$h_w = \frac{\sum_{i=1}^{L} w_i e_i}{\sum_{i=1}^{L} w_i} \quad [2]$$

where w_i is the weight of link i. In the present paper, we will focus on unweighted networks.

We applied the flow hierarchy metric to the simple networks shown in Figure 1 and the larger examples shown in Figure 2. The calculated hierarchy degrees (see Table 1) capture the same understanding based upon direct observations. In particular, the metric performs well in assessing layered hierarchy but other potential metrics do not. For the example of network D in Figure 1, if we alternatively count the portion of nodes rather than links, all the nodes are involved in cycles so the alternative hierarchy metric will be zero and fail to capture the sense of layered hierarchy of this network. In general this metric is unambiguous in differentiating the

hierarchical components and non-hierarchical components [23]. It is also advantageous in its clarity and ease of computation, in comparison to other potential metrics (An assessment of alternative metrics is provided in the Supplementary Material [24]).

Table 1 Hierarchy degrees of the example networks in Figure 1 and 2

Networks	Figure 1					Figure 2	
	A	B	C	D	E	A	B
Hierarchy Degree	1	1	1	0.40	0.57	0.33	1

This metric of flow hierarchy potentially provides a way to characterize and detect different structural regimes of discrete systems with a potential direction, analogously to the different regimes of the continuous fluid flows. For example, the networks A and B in Figure 3 are strictly hierarchical (uni-directional) and similar to the "laminar flow" regime of fluid flows. In network C of Figure 3, some of the local flows (i.e. links) are involved in cycles (similar to eddies or vortexes of fluid flows). The system is no longer purely hierarchical and is in a "transitional flow" regime. In network D, all the flows are involved in cycles, so this case is analogous to the "turbulent" regime of fluid flows. Thus, as the Reynolds Number [25] characterizes different flow regimes, such as laminar, turbulent or transitional flow, the flow hierarchy metric also potentially characterizes the structural regimes of discrete network systems, such as production markets, food webs, and software.

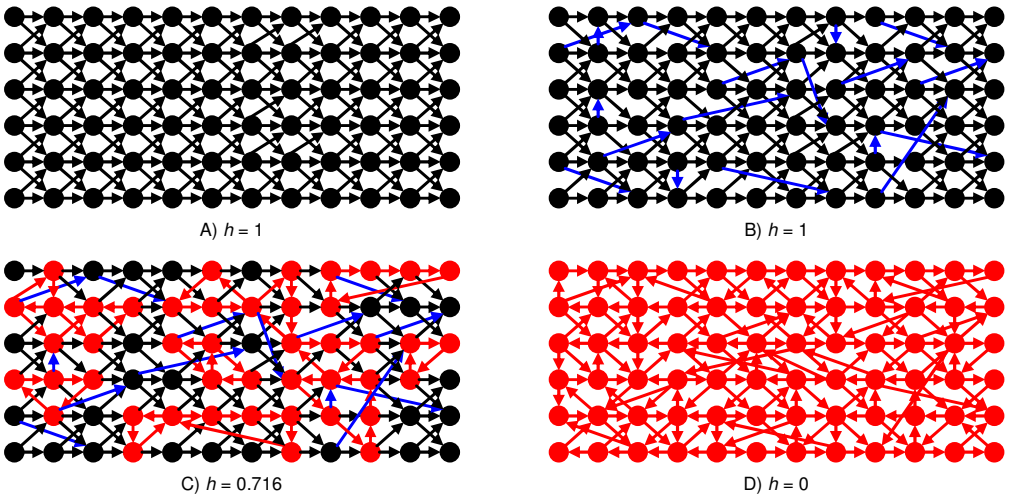


Figure 3 A network with 72 nodes and 176 directed links, oriented toward different directions in four scenarios. The links in blue color in B) and C) either skip levels or connect between nodes in the same level. Such links add complexity and difficulty in determining the levels and ranks, but do not destroy the overall network directionality, i.e. flow hierarchy. The nodes and links colored in red are involved in cycles.

To compute the flow hierarchy metric for large-scale complex networks, we use the following algorithm: First, we construct the link adjacency network and matrix for the original node adjacency network. For example, Figure 4 shows the link adjacency network transformed from and equivalent to the original node adjacency network. The 7 squares in Figure 4B correspond to the 7 links of the network of Figure 4A respectively.

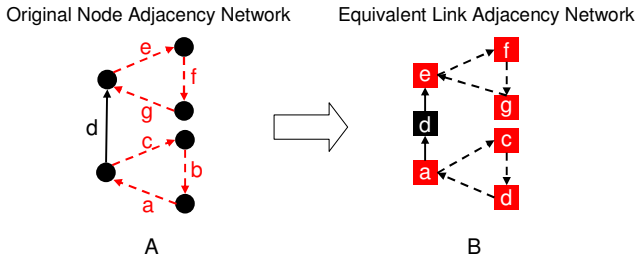


Figure 4 The link network equivalent to the original node network

We name the cell (i,j) in the link adjacency matrix x_{ij}. $x_{ij} = 1$ if and only if the end of link i is directly connected to the start of link j by a node. Otherwise, $x_{ij} = 0$. Second, we raise the link adjacency matrix's power p to find the link distance matrix M_d. We name the cell (i,j) in the link distance matrix d_{ij}. d_{ij} is the distance from link i to j, defined as the minimum number of unique nodes which a uni-directed flow has to travel through from the end of link i to the start of link j. d_{ij} is found as the value of the power, at which cell (i,j) of the power matrix M^p has a non-zero value for the first time. When p=1, the power matrix M^1 is the same as the link adjacency matrix, so that if x_{ij}=1, the distance from i to j is 1. If x_{ij} =0, and $x^{[2]}_{ij}$>0, then the distance is found as 2. And so forth. Consequently, the first power p for which the $x^{[p]}_{ij}$ element is non-zero gives the distance from i to j, i.e. the value of d_{ij} in the link distance matrix M_d. Mathematically, $d_{ij} = \min_p x^{[p]}_{ij} > 0$, for p from 1 to L, the total number of links (equal to the length of the longest possible cycle of links). We leave d_{ij} empty if the end of link i is neither directly nor indirectly connected to the start of link j. Note that alternative algorithms, such as depth-first search, can also be applied to find the link distance matrix.

Figure 5 illustrates the process to derive the link distance matrix for the example network in Figure 4. Given the final link distance matrix (at the bottom right corner of Figure 5), we are able to judge if a link is on any directed cycle by examining its main diagonal. If d_{ii} is empty, then link i is not involved in any cycle (i.e. e_i=1, 0 otherwise). In this case, only d_{dd} is empty, and this agrees with our direct observation on Figure 4 – only link d is not included in any cycle. Thus, the hierarchy degree is 1/7.

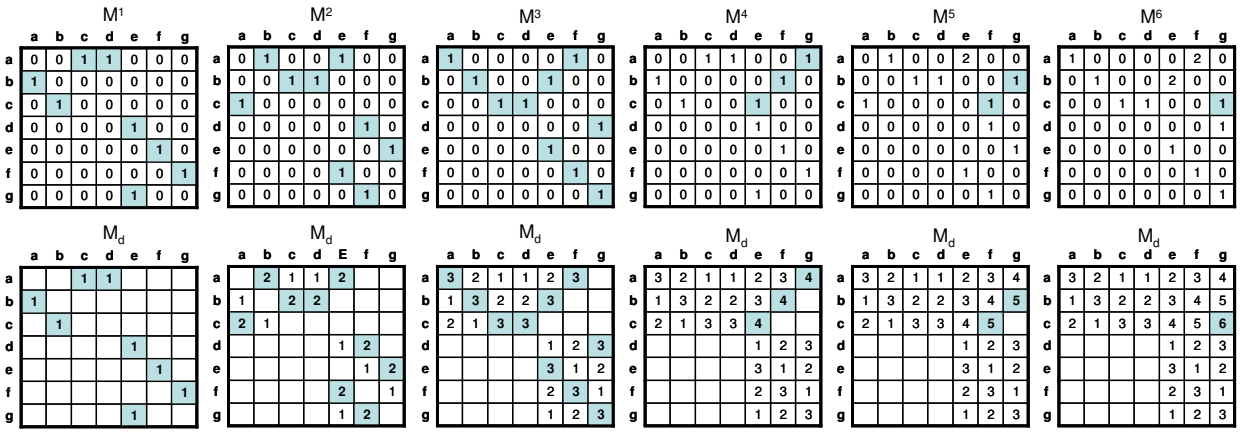


Figure 5 Deriving link distance matrix by raising power of link adjacency matrix. We pair M^p and the M_d with the state of knowledge after p steps. M^1 is the link adjacency matrix for the link network in Figure 4B and the original network in Figure 4A. The distance identified at each intermediate step is bold and its cell is shadowed. In particular, the values on the diagonal of the final M_d (after 6 steps in this case) give the length of the shortest link cycles in which each link is included.

Results and Discussions

By definition, a pure random directed network embeds no hierarchy. However, the hierarchy degree is not necessarily zero for the networks created by existing random network models. We have examined hierarchy degrees of networks generated by a simple model similar to the "Poisson random network" [9]. Networks are constructed by assigning *L* directed links to randomly chosen pairs from *N* nodes. No multiple uni-directed links between a chosen pair and no self-links are allowed.

The directed Poisson random networks alone also exhibit important properties regarding hierarchy. Figure 6A shows that network size (*N)* has little influence on hierarchy degree (*h*), especially when *N* is sufficiently large. This agrees with our intuition that hierarchy is an

architectural property independent of scale, and allows one to use random networks with a relatively small N to estimate h of those with large N but the same k (=L/N). Figure 6B shows the increase of average degree (k) significantly decreases h. When k is at its minimum 1/N, h will be exactly 1, because only one pair of nodes will be connected and one node is unambiguously above the other in this dyad flow hierarchy. When k is sufficiently high, h tends to zero because there are many cyclic pathways through which flow can travel back to its origin. A holistic direction does not exist among links in densely connected random networks. These results are of use when comparing h found from empirical networks.

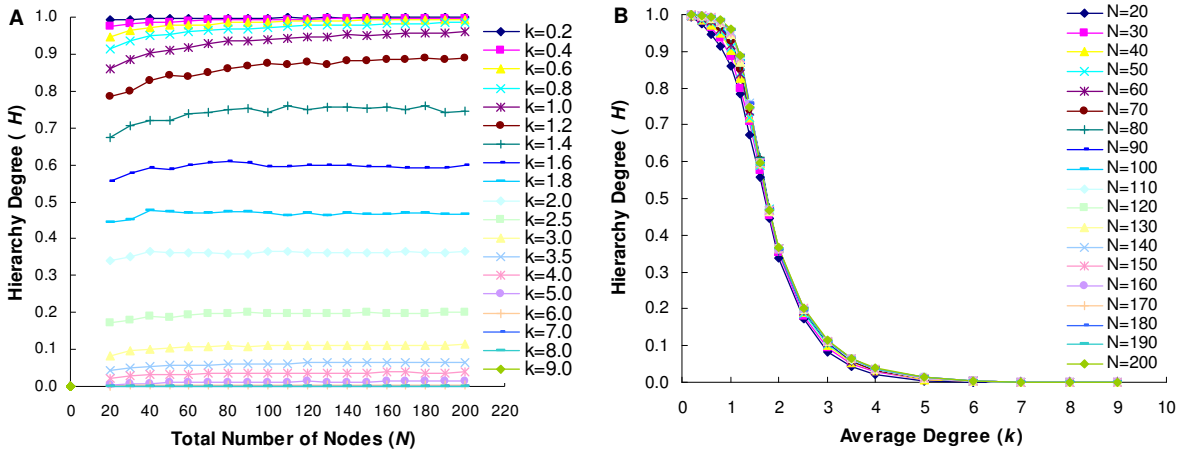


Figure 6 Hierarchy degrees of randomly-generated directed networks. The value at each data point is the average of hierarchy degrees of 1,000 randomly-generated networks given the same N and k. Data points are connected by straight lines.

We calculated the hierarchy metrics of a diverse set of empirical evolving self-organizing networks: the Bridge Brook Lake food web [26,27] and the Northeastern US Shelf food web [28,29], Japanese supplier-producer networks in automotive and electronics production sectors [30, 31, 32], two biological information-processing networks including the synaptic connections between neurons in the nematode worm Caenorhabditis elegans [33] and developmental transcription network of Drosophila Melanogaster [33], the call networks of the kernels of two

operation system software, Linux [34] and Apple computer's Mac OS X (Darwin) [35]. We used the algorithm detailed above to compute the hierarchy degrees of these large-scale empirical networks.

Hierarchy degrees of these empirical networks are compared to those of comparable Poisson random networks with the same numbers of nodes (N) and links (L), with a focus on z-score,

$$z = \frac{h_{real} - \hat{h}_{rand}}{\sigma_{rand}} \qquad [3]$$

where h_{real} is the hierarchy degree of the empirical network, and $\hat{h}_{rand}$ and σ_{rand} are the average and standard deviation of the hierarchy degrees of an ensemble of randomly-generated networks with the same N and L (or k) of the empirical network. The larger the value of the *z-score*, the more the structure of the network deviates from randomness[1].

Table 2. Hierarchy degrees of empirical networks and comparable random networks

Network	Type	N	L	k	h_{real}	h_{rand}	σ_{rand}	z-score
Bridge Brook Lake	Food Web	25	104	4.160	0.9809	0.0213	0.0338	28.39
NE US Shelf	Food Web	79	1378	17.443	0.8273	0	0	infinite
Japanese Automobile Sector	Production	679	2437	3.589	0.9988	0.0601	0.0114	82.34
Japanese Electronics Sector	Production	227	648	2.855	0.5957	0.1338	0.0310	14.90
C. elegans	Biological	280	2170	7.750	0.1171	0.0009	0.0018	64.56
D. melanogaster	Biological	107	301	2.813	0.3289	0.1308	0.0444	4.46
Darwin XNU-123.5	Software	646	4351	6.735	0.4872	0.0024	0.0021	230.86
Linux Kernel 1.1.70	Software	287	1385	4.826	0.8065	0.0159	0.0082	96.41

Each empirical network is compared to an ensemble of 1,000 randomly-generated networks with the same N and L (or k). We extracted the software call networks using the architecture analysis software Understand C++. In the call networks, a link from

[1] The results in Figure 6B shows that the hierarchy degree is not necessarily zero for random networks. This indicates a potential kind of "noise" hidden in hierarchy degree interfering with indication of hierarchy due real-world patterns. The *z-score* metric aids in assuring that the observed hierarchy goes beyond such "noise".

source code B to source code A exists if any function in A calls and relies on any function located in B. In the industrial production networks, a link from firm B to firm A exists if firm A procures any products from firm B.

Domain-specific knowledge is needed to understand the detected difference in hierarchy degree of empirical networks of the same type. For the specific example of the two production networks, the automotive sector is significantly more hierarchical than the electronics sector (Table 2). This difference in hierarchy degree may imply and result from some important differences in the strategies and behaviors of individual firms and differences in the technological environments in the two production sectors[2].

However, in the results (Table 2), there is no clear evidence to show that system types (e.g. biological vs. production) differentiate networks in terms of flow hierarchy. In general, from a network science perspective, the results show all of these typical empirical networks exhibit stronger hierarchical architectures than comparable random networks with the same sizes and average degrees, indicating the emergence of hierarchy as a significant feature of real-world evolving self-organizing networks.

This indicates that ***evolution*** (e.g. the extent to which the system has evolved) might be one fundamental determinant of the hierarchical degrees of self-organizing networks. Simon [1] first hypothesized that hierarchy emerges inevitably through a wide variety of evolutionary processes because hierarchical structures are stable [1,36]. However, quantitative evidence on hierarchy, either containment or flow hierarchy, over the course of system evolution has not been reported

[2] Reference [31] attempts to explain why different industrial sectors may exhibit different degrees of hierarchy in the specific industrial and economic context.

previously, largely due to the lack of an appropriate measure. The hierarchy metric and technique in the present paper allows the exploration of this fundamental question linking hierarchy and system evolution.

We calculated the hierarchy metrics of the call networks of various historical versions of the Linux kernel from its origin, version 0.01 to 2.3.0. The Linux kernel is an open source system developed by self-organized contributors around the world. As indicated in Figure 7A, the hierarchy degree and corresponding z-score (Figure 7B) of the Linux kernel have been generally increasing over its life cycle. The first version (0.01) was built and released by a single person. After that, many people contributed subroutines to the project, and thus hierarchy degree declined for a little while. During the most of its life as an open-source system, the hierarchy degree has increased as the self-organizing system grows, stabilizes, and matures, as Simon [1] argued. The observation of a general increase of k as the system evolves (Figure 7C) affirms the hierarchical tendency of this system since increases in k alone would work to decrease the hierarchical metric.

Network decompositions may reveal certain underlying architectures and interesting methods to detect modularity have been developed recently [11,16,19,21,37-39]. We also calculated the optimal modularity of the Linux kernel networks, using Newman's eigenvector-based algorithms for both undirected [38] and directed networks [39], and found unclear trends during the same period of time, if not slightly decreasing. No theoretical or observational indication has been found about how modularity of self-organizing networks should change in evolutionary

processes. Compared to the flow hierarchy metric, the usefulness of modularity in terms of tracking the evolving patterns of self-organizing networks may be limited.

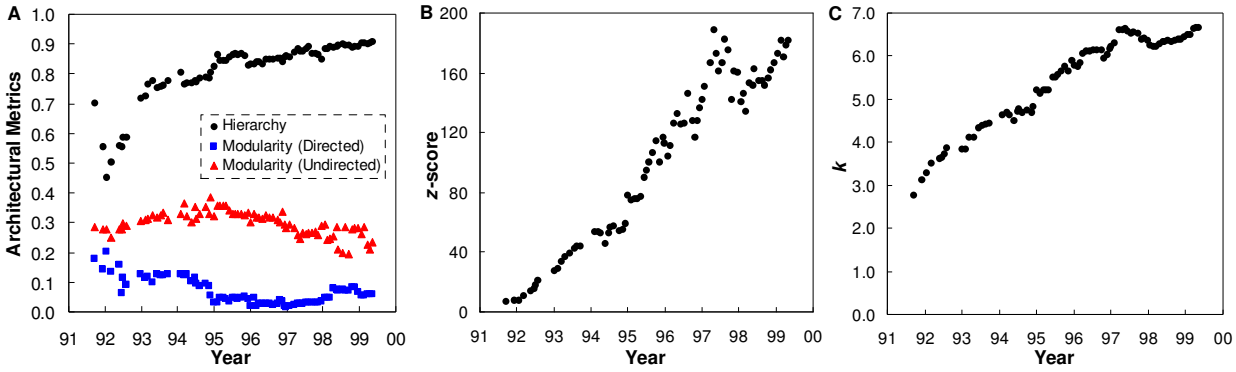


Figure 7 Longitudinal evolution of Linux kernel. (A) hierarchy degree and modularity. (B) *z*-score. (C) average degree. At least one data point is included for each month when there are multiple releases in a month. For some months, no data point is included because there were no versions either released or available in the online archive. See Supplementary Material for more details of the data and results.

Conclusion

In general, this paper explores a commonly observed but theoretically overlooked form of hierarchy in networks -- flow hierarchy. Our measurement technique makes it possible to objectively compare hierarchies of different networks, detect the structural regime or evolutionary stages of a single network, and compare the evolving patterns of different networks. Our analysis shows that the ecological, neurobiological, economic and information processing networks are generally more hierarchical than their comparable random networks. We further discovered that hierarchy degree has increased over the course of the evolution of Linux kernels, confirming an early hypothesis by Herbert Simon on the emergence of hierarchy in evolutionary processes. Taken together, the results may suggest that flow hierarchy is a central organizing feature of real-world evolving networks.

Our major purpose of this article is not to argue flow hierarchy must increase or decrease in the evolutionary course of a complex system, but to stimulate more research to explore the value of flow hierarchy as lens to understand the architecture and dynamics of complex network systems. This paper is not intended as the final word on flow hierarchy, but a beginning of further and boarder research on it. Important questions, such as what flow hierarchy means to the functional performance of a network and how flow hierarchy emerges from the behaviors and interactions of individual network nodes, have not been answered.

We anticipate application of the hierarchy metric and measurement technique to more systems, such as ecological, biological, brain and neural, social and technological systems, in order to help understand better their domain-specific architectures and evolutionary patterns. Like the contribution of Reynolds Number for the development of the overall field of fluid mechanics, the flow hierarchy metric may also potentially provide great value for designing and managing complex network systems, but further research is needed.

Acknowledgements

We thank Daniel Whitney, Luis A. N. Amaral, and Joel Moses for insightful and stimulating discussions, Jennifer Dunne for providing the food web data, Ron Milo for providing the biological network data, and Kevin Groke for technical support to extract the call networks of software packages. We also thank MIT Portugal Program Engineering Systems Fundamentals Project for providing funding for this research.

References

1. Simon, H. The architecture of complexity. Proc. American Phil. Soc 1962, 106, 467-482.
2. Holland, J. Emergence: From Chaos to Order; Addison-Wesley: Reading, 1998.
3. Anderson, P.W. More is different: Broken symmetry and the hierarchical nature of science. Science 1972, 177, 393-396.
4. Watts, D.J.; Strogatz, S. Collective dynamics of 'small-world' networks. Nature 1998, 393, 440-442
5. Price D.J.de Solla. Networks of scientific papers. Science 1965, 149, 510-515.
6. Barabasi A-L; Albert, R. Emergence of scaling in random networks. Science 1999, 286, 509-512.
7. Albert, R.; Barabasi, A-L. Statistical mechanics of complex networks. Rev. Mod. Phys. 2002, 74, 47-97.
8. Strogatz, S.H. Exploring complex networks. Nature 2001, 410, 268-276.
9. Newman, M.E.J. The structure and function of complex networks. SIAM Rev. 2003, 45, 167-256.
10. Ahl, V.; Allen, T. F. H. Hierarchy Theory: A Vision, Vocabulary and Epistemology; Columbia University Press: New York, 1996.
11. Clauset, A.; Moore, C.; Newman, M. E. J. Hierarchical structure and the prediction of missing links in networks. Nature 2008, 453, 98-101.
12. Lane, D. Hierarchy, complexity, society, in Pumain, Denise (ed.), Hierarchy in Natural and Social Sciences, 81-119; New York: Springer-Verlag, 2006.
13. Alexander, C. Notes on the Synthesis of Form; Harvard University Press: Cambridge, MA, 1964.
14. Wilson, D., 'Forms of hierarchy: a selected bibliography,' In L. L. Whyte, A. G. Wilson and D. Wilson (eds.), Hierarchical Structures, 287-314; American Elsevier: New York, 1969.
15. Wasserman, S.; Faust, K. Social Network Analysis; Cambridge University Press: Cambridge, 1994.
16. Sales-Pardo, M.; Guimera, R.; Moreira, A.A.; Amaral, L.A.N. Extracting the hierarchical organization of complex systems. Proc. Natl. Acad. Sci. U.S.A. 2007, 104, 15224-15229.
17. Ravasz, E., Barabasi, A-L. Hierarchical organization in complex networks. Phys. Rev. E 2003, 67, 026112
18. Hsieh, M. H.; Magee, C. L. An algorithm and metric for network decomposition from similarity matrices: Application to positional analysis. Social Networks 2008, 30, 146-158.

19. Hsieh, M.-H.; Magee, C.L.. A new method for finding hierarchical subgroups from networks. Social Networks 2010, doi:10.1016/j.socnet.2010.03.005
20. White, H. C. Businesses mobilize production through markets: parametric modeling of path-dependent outcomes in oriented network flows. Complexity 2002, 8(1), 87-95.
21. Girvan, M.; Newman, M. E. J. Community structure in social and biological networks. Proc. Natl. Acad. Sci. USA. 2002, 99, 7821-7826.
22. Nakano, T.; White, D. R. Network structures in industrial pricing: the effect of emergent roles in Tokyo supplier-chain hierarchies. Structure and Dynamics 2007, 2(3).
23. Moses, J. Three design methodologies, their associated organizational structures and their relationship to various fields. Proceedings of Engineering Systems Symposium, MIT, 2004. Available at http://esd.mit.edu/symposium/pdfs/papers/moses.pdf
24. Supplemental Material can be found at the end of document.
25. Reynolds, O. An experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and of the law of resistance in parallel channels. Philosophical Transactions of the Royal Society of London 1883, 174, 935-982.
26. Havens, K. Scale and structure in natural food webs. Science 1992, 257, 1107-1109.
27. Dunne, J.A.; Williams, R.J.; Martinez, N.D. Network structure and biodiversity loss in food webs: robustness increases with connectance. Ecology Letters 2002, 5, 558–567.
28. Link, J. Does food web theory work for marine ecosystems. Mar. Ecol. Prog. Ser. 2002, 230, 1-9.
29. Dunne, J.A.; Williams, R.J.; Martinez, N.D. Network structure and robustness of marine food webs. Mar. Ecol. Prog. Ser. 2004, 230, 291-302.
30. Dodwell Marketing Consultants. The Structure of Japanese Auto Parts Industry; Tokyo, 1993
31. Dodwell Marketing Consultants. The Structure of Japanese Electronics Industry; Tokyo, 1993
32. Luo, J. Hierarchy in industry architecture: Transaction strategy under technological constraints. Doctoral Thesis 2010, Massachusetts Institute of Technology, Cambridge, MA.
33. Milo, R.; Itzkovitz, S.; Kashtan, N.; Levitt, R.; Shen-Orr, S.; Ayzenshtat, I.; Sheffer, M.; Alon, U. Superfamilies of Evolved and Designed Networks. Science 2004, 303, 1538-1542.
34. Linux Kernel Organization. Inc. http://www.kernel.org/.

35. Apple, Inc. http://www.opensource.apple.com/darwinsource/10.0/.
36. Agre, P. Hierarchy and history in Simon's 'Architecture of complexity. Journal of the Learning Sciences 2003, 12, 413-426.
37. Guimera, R.; Sales-Pardo, M.; Amaral, L.A.N. Module identification in bipartite and directed networks. Phys. Rev. E 2007, 76, 1-8.
38. Newman, M. E. J. Modularity and community structure in networks. Proc. Natl. Acad. Sci. U.S.A. 2006, 103, 8577-8582.
39. Leicht, E. A.; Newman, M.E.J. Community structure in directed networks. Phys. Rev. Lett. 2008, 100, 118703.

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.

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

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.

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

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

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

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

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

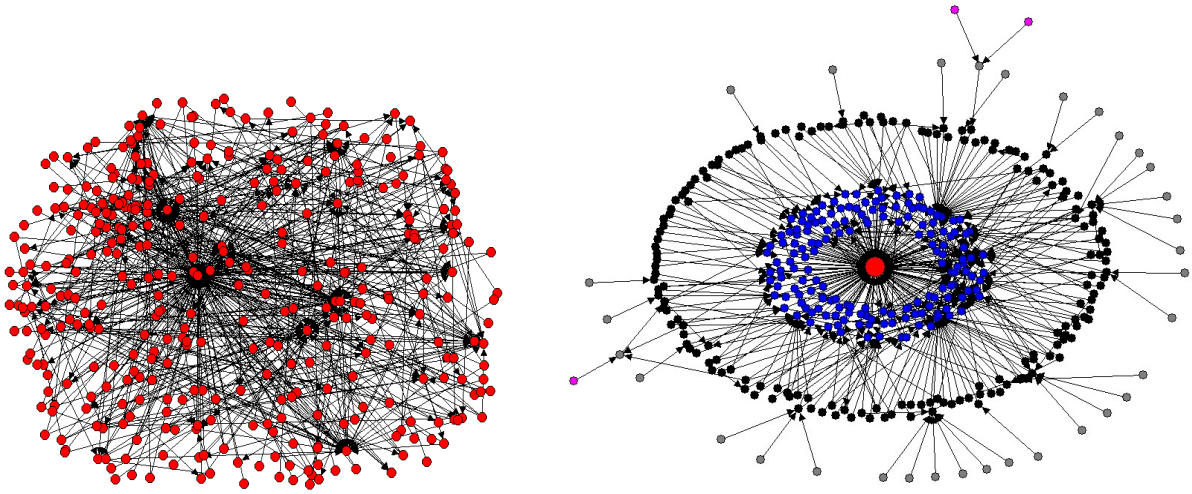
5) Continuous Level Rank (Average)

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

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. $i < j$.

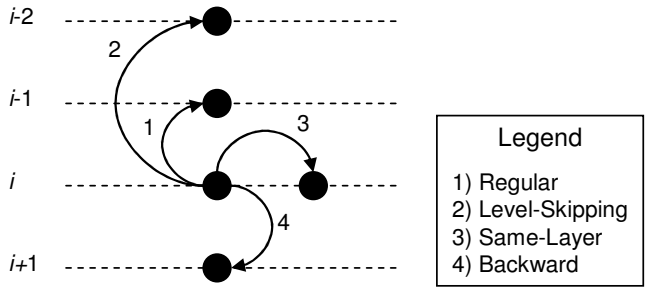


Figure S2. Examples for four types of links identified according to levels

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:

$$Regular > Level\text{-}Skipping > In\text{-}Layer$$

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.

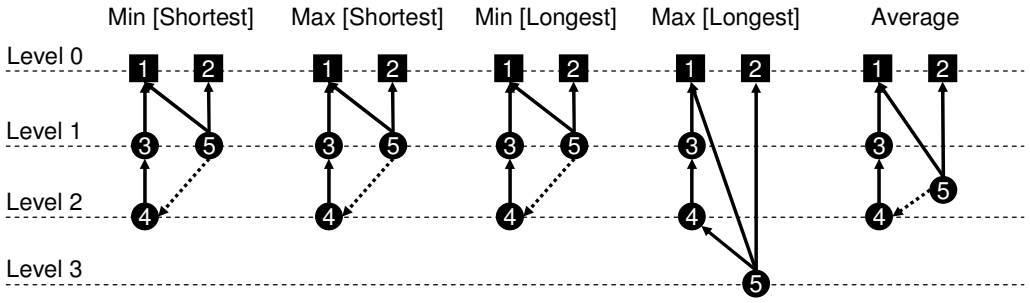


Figure S3. Identifying nonhierarchical links based on five different level ranking algorithms. Non-hierarchical backward links are dashed.

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.

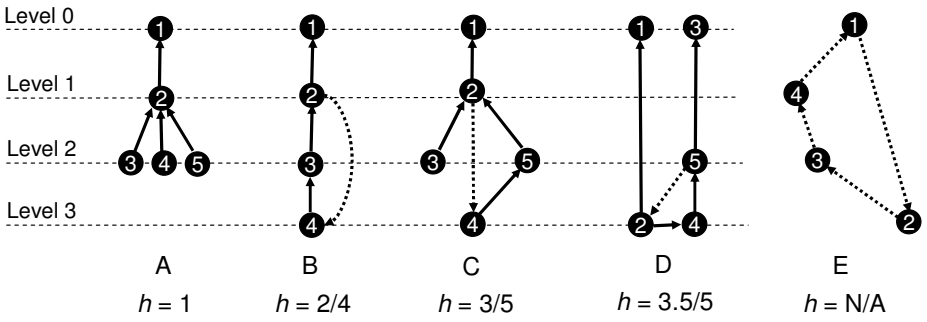


Figure S4. Hierarchy degrees of example networks based on the Max [Longest] ranking algorithm. Non-hierarchical backward links are dashed.

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。