BRIEFINGS IN BIOINFORMATICS. VOL 13. NO 5. 627–644
Advance Access published on 9 December 2011
doi:10.1093/bib/bbr069

Hive plots—rational approach to visualizing networks

Martin Krzywinski, Inanc Birol, Steven J.M. Jones and Marco A. Marra

Submitted: 16th June 2011; Received (in revised form): 9th November 2011

Abstract

Networks are typically visualized with force-based or spectral layouts. These algorithms lack reproducibility and perceptual uniformity because they do not use a node coordinate system. The layouts can be difficult to interpret and are unsuitable for assessing differences in networks. To address these issues, we introduce hive plots (http://www.hiveplot.com) for generating informative, quantitative and comparable network layouts. Hive plots depict network structure transparently, are simple to understand and can be easily tuned to identify patterns of interest. The method is computationally straightforward, scales well and is amenable to a plugin for existing tools.

Keywords: *networks; visualization; bioinformatics; graphs; systems biology; computation*

INTRODUCTION

Networks are routinely used in biology to capture and model the complexity and dynamics of relationships between functional units in a genome, cell, or tissue [1–3]. Networks are used in the study of gene transcription and regulation [4], protein interactions [5, 6], metabolic pathways [7, 8], genetic basis of disease [9], genome assembly [10, 11] and classifying repetitive structures therein [12]. They capture the full problem domain as a single system [13, 14], integrate experimental evidence with computational approaches [15] and characterize both global properties of the system [16] and individual components in their context, such as genes implicated in cancer [5, 17]. Network analysis [18, 19] is complemented by visualization to bridge computation and interpretation, accelerate discovery and deepen insight [20, 21].

Investigators rely on visual interfaces to networks to manage complexity, help create a mind-map of the patterns and cogently communicate their findings. Visualization applications such as BiologicalNetworks [22], Cytoscape [23, 24], Gephi [25], MatrixExplorer [26], Osprey [27] and others [20, 21] are continually challenged to cope with growing data sets and informatively render large networks [28]. To address the needs of users, who are faced with increasingly larger cognitive load in interpreting the visualizations, layout algorithms continue to be developed and refined to make network visualizations more approachable and informative [20, 29]. These include Fruchterman

Corresponding author. Martin Krzywinski, BC Cancer Research Center, BC Cancer Agency, 100-570 W 7th Ave, Vancouver, BC V5Z 4S6, Canada. Tel: +604 707 5900 ext 673262; Fax: +604 876 3561; E-mail: martink@bcgsc.ca

Martin Krzywinski is a research scientist at the BC Cancer Research Center and works in the field of data visualization. He is the creator of Circos [Circos: an information aesthetic for comparative genomics. *Genome Res* 2009;**19**(9):1639–45], a standard for visualization of genome comparisons.

Inanc Birol is the bioinformatics group leader at Canada's Michael Smith Genome Sciences Centre and adjunct professor at the School of Computing Science at Simon Fraser University. His research interests include the analysis of short read sequencing data to study genomes, epigenomes and transcriptomes of model species and humans.

Steven Jones is the head of bioinformatics and associate director of Canada's Michael Smith Genome Sciences Centre and professor of molecular biology and biochemistry at Simon Fraser University and medical genetics at University of British Columbia. He develops computational approaches to analyse DNA sequence information, especially deriving from the next-generation DNA sequencing technology, to gain insights to the mutations and other DNA re-arrangements which occur and accrue within the oncogenic process which give rise to cancer.

Marco Marra is the director of Canada's Michael Smith Genome Sciences Centre and a professor of medical genetics at the University of British Columbia. His main interest is the genetic basis of health and disease. He directs research to improve cancer control and outcomes through better understanding of how cancer develops, how it can be detected earlier and how it can be targeted with specific therapeutics.

Reingold [30], a force-directed method which generates pleasant layouts with uniform edge and vertex distribution, OpenOrd [31], a multilevel algorithm that scales to millions of nodes and effectively reveals clusters and the Sugiyama method [32, 33] ideal for discovering a network's hierarchy. Unfortunately, the effectiveness of these methods is reduced by inherent unpredictability, inconsistency and lack of perceptual uniformity. As a result, their layouts have earned the disparaging moniker 'hairballs'.

Network layouts are difficult to predict and interpret because their creation is in part, driven by an aesthetic heuristic that can influence how specific structures are rendered. The interplay between form and function can result in layouts that remain unconvincing about whether the network contains any meaningful patterns at all (Figure 2 in [34]), given that in other layouts similar patterns are acknowledged to be entirely artefactual (Figure 2B in [35]). This inherent unpredictability is confounded by the fact that different algorithms generate very different layouts of the same network (Figure 1A) and limits the usefulness of complementing one layout with another. The lack of interpretable structure in many layouts can imply variation where none exists (Figure 1B) and hide meaningful network patterns (Figure 2A and B). Most algorithms are sensitive to small changes in the network and create perceptually nonuniform layouts, where differences vary out of proportion to changes in the network. Figure 2C shows the extent to which a hierarchical layout can change upon removal of an edge or node. This lack of robustness is not limited to small networks, as exemplified in Figure 2E, which shows the impact of removing a single gene on the layout of the gene regulatory network RegulonDB [36, 37]. The magnitude of this effect will vary, but its presence makes such layouts unsuitable for comparing networks and prevents us from formulating a robust visual query language to interrogate network visualizations.

To make layouts rational, informative and reproducible, we introduce the 'hive plot' (HP), in which nodes are placed on radially oriented linear axes according to a well-defined coordinate system. HPs satisfy the five requirements for an effective and

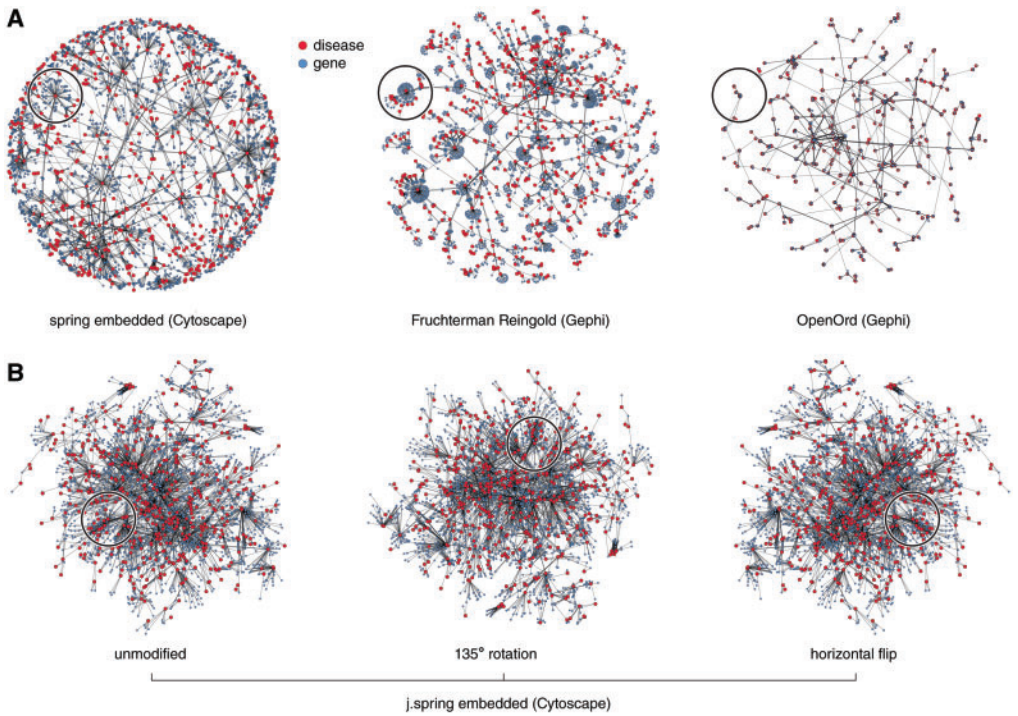

Figure 1: (**A**) Network drawing algorithms can produce strikingly different layouts when applied to large networks. Visualizations of the largest connected component (2652 nodes, 3941 edges) of the human disease network [38] generated with Cytoscape 2.8.1 [23, 24, 39] and Gephi 0.7 [27]. In each panel, the same group of nodes are highlighted to show the variety of ways in which dense hubs are rendered by each algorithm. (**B**) Affine transformations of Cytoscape's j.spring embedded algorithm result in layouts that can be easily confused with those of different networks. The same group of nodes as in panel A is highlighted.

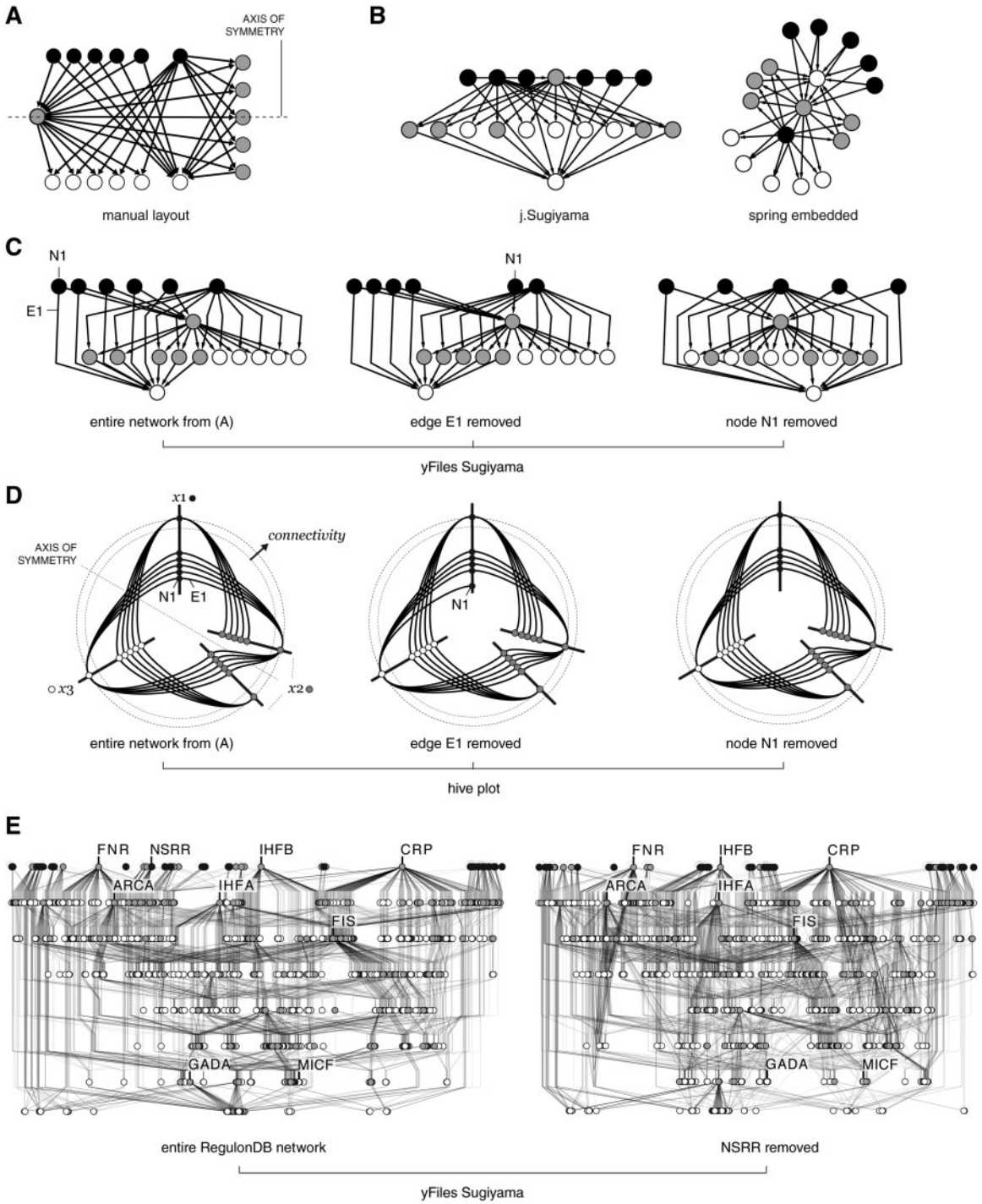

Figure 2: Conventional network drawing algorithms generate layouts that lack robustness—small changes in a network can have a disproportionately large effect on the entire layout. (**A**) Manual layout of a 15-node symmetric directed network. (**B**) Cytoscape's layouts of (A) using the j.Sugiyama [32] and force-directed algorithms obscure the simplicity and symmetry of the network. (**C**) Cytoscape's yFiles hierarchic layout [49] is very efficient and useful for showing the flow in a network, but exemplifies brittleness. The layout of the network from (A) varies considerably when edge EI and node NI are removed. (**D**) HPs of the networks in (C) are robust and globally unaffected by local changes to the network. The rules for these HP are explained in Figure 3C. (**E**) Brittleness is seen in layouts of large networks. The removal of a single top regulatory gene (NsrR, a regulatory protein, which is a member of the Rrf2 family.) from RegulonDB network [36, 37] alters layout significantly, making it impossible to determine what has changed.

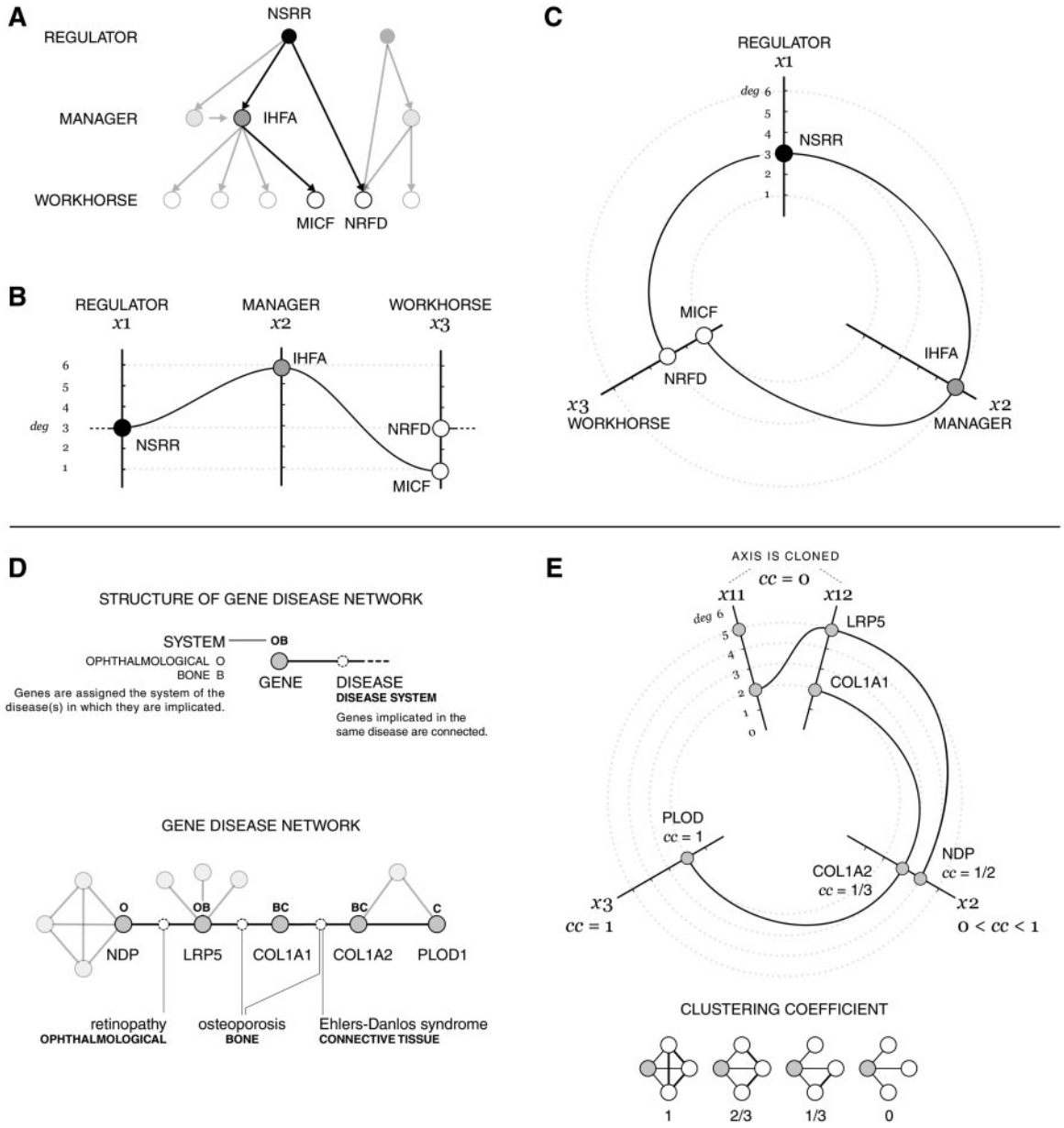

Figure 3: Process of creating HPs from directed and undirected networks. (**A**) The structure of the RegulonDB network [36, 37]. (**B**) A parallel coordinate plot of the genes highlighted in (A), assigned to axes (*x*1, *x*2, *x*3) based on their role (regulator, manager, workhorse) and positioned on the axis based on connectivity (*deg*). (**C**) HP of the genes highlighted in (A), showing the conceptual similarity between the HP and parallel coordinate plot. The circular layout of the HP permits connections between edge axes in the parallel coordinate plot (NSRR/NRFD) to be accommodated within the plot area. (**D**) The structure of the gene-disease network [38] that connects genes implicated in the same disease. Classification of the connecting diseases (e.g. ophthalmological, bone, connective tissue, etc.) partitions the network into overlapping sets of genes. (**E**) The clustering coefficient (cc) measures the extent of connections between a node's neighbors and is used to place genes on HP axis. HP of the highlighted nodes in (D) constructed using ranges of the cc, for axis assignment and connectivity for axis scale. The *x*1-axis is cloned (*x*11, *x*12) to reveal connections between cc = 0 nodes.

aesthetic layout [40]: generality (can be applied to different classes of networks), flexibility (can be adjusted to suit their purpose), transparency (can be easily explained and understood), competence (generate useful and quantitatively interpretable results) and speed (render typically much faster than traditional layouts). In addition to these, HPs have two other critical properties that distinguish them

from other layouts: reproducibility and perceptual uniformity.

The HP layout is a parallel coordinate plot [41] (typically applied to multi-genome alignments [42–44]) in which the axes are radially arranged, making the layout compact and connections easy to follow. HPs share similarities with other methods that implement a coordinate system for node layout, such as PivotGraph [45] and Semantic Substrates [46, 47], which project nodes onto multiple nonoverlapping regions whose internal layout is rectilinear and based on one or two attributes. Semantic substrates can visually capture relationships between an arbitrary number of annotations (Figure 9 in [47]) and are very effective when the layout within each panel is not overly complex, such as when ordinal or nominal attributes have a small number of different values. As the number of panels and attribute values grows, the edges can become dense and difficult to interpret (Figure 10 in [47]). In a noninteractive form, it is not possible to isolate a single node from others because its positions within each panel cannot be directly related. In comparison, HPs aim to provide a quantitative representation based on two structurally-derived attributes and apply a radial scheme to help declutter the layout. A radial approach has been used previously [48] to represent layers in a hierarchical Sugiyama layout [32]. HPs are distinguished by projecting the network onto a multidimensional coordinate system that is based directly on structural properties, such as connectivity or clustering. The coordinate system creates consistent layouts of network components that have the same structural profile and can be tuned to reveal specific patterns. This consistency makes HP appropriate for comparing networks and monitoring their evolution. Network comparison with HPs is possible because the layouts are perceptually uniform—modifying the network alters the HP only in parts directly affected by the change, making it is easy to spot the resulting difference (Figure 2D).

HPs provide quantitative visual signatures, useful for large networks, a task that challenges conventional layouts. They can quickly reveal patterns such as hubs and clusters and identify structurally equivalent elements in the context of the full data set. HPs complement traditional layouts (e.g. force-based or semantic substrates) and can be displayed alongside for multimodal navigation of a network, particularly when functional components are delineated with constraint-based methods [50]. They can be enhanced by the same approaches that make traditional layouts easier to understand, such as reducing visual complexity by node removal and grouping [20] and including additional (e.g. matrix) representations [26]. HPs are interpretable at small sizes and can be grouped into a matrix of plots, a 'hive panel', to create a dashboard multivariate visualization of independent properties, similar to a correlation scatter plot matrix (Figure 6A in [51]). HPs may initially appear complex, but differ from other approaches in that their complexity scales well and is accessible to inspection. With familiarization, a human reader can quickly develop facility to interpret them, unlike 'hairballs', which cannot be consistently deconstructed into patterns and remain opaque to even experienced eyes.

This article introduces HPs and demonstrates their application. Using the *Escherichia coli* RegulonDB [36, 37], a directed gene regulatory network, we will show how HPs can be used to reveal structure. The human disease network (HDN) [38], an undirected network that relates genes to disease, will be used to demonstrate how to use HP to compare networks by contrasting the structure of different disease system in the network. This is conceptually equivalent to comparing two networks derived from the same data set, such as the gene and disease networks in the human disease networks (Figure 2 in [38]), or two networks derived from different experimental conditions, such as regulation in MCF-7 human breast cancer cells stimulated by epidermal growth factor or heregulin with several doses [52]. Beyond networks, HPs can communicate complex data sets that can be mapped onto a three-axis model, such as synteny between three genomes [53, 54], where the HP representation is even more effective than a circular one [55].

HIVE PLOTS

The HP is a network layout algorithm that places nodes on radially oriented linear axes with a coordinate system based on nodes' structural properties. HPs comprise three flexible components: (i) rules that govern assignment of nodes to axes (and, optionally, to collinear axis segments) and node coordinates, (ii) layout profile of axes (position, scale and orientation) and (iii) rules that control the format of edges drawn as curves between nodes. The combination of radially arranged axes and curved edges resembles the top of a beehive—hence 'hive plot'.

Table 1: Node and graph metrics useful for HP axis assignment and node coordinate rules

Structural parameter	Definition
Node parameters	
Degree (connectivity)	Number of edges incident on a node. For directed graphs, this quantity has an orientation and can be used to establish 'flow'.
Flow	For a directed graph, the difference between the number of out edges and in edges. Nodes with positive flow are sources, and those with negative flow are sinks.
Betweenness [57]	The number of shortest paths on which a node lies.
Closeness [8]	Average distance between a node and all others reachable from it.
Eccentricity	Maximum distance between a node and all others reachable from it.
Page rank [58]	A type of eigenvector centrality. For a directed graph, the popularity of a node as measured by simulating a walk across the graph along its edges, with an optional dampening factor that resets the walk to a random node. Can be applied to both directed and undirected graphs.
Clustering coefficient [59]	The extent to which the neighbors of a node are connected. Nodes with a large clustering coefficient form highly connected neighborhoods.
Topological overlap [60]	The similarity between two nodes expressed as the normalized number of shared neighbors between the nodes.
Cut vertex	Defined as 1 if the removal of a node disconnects the graph, 0 otherwise.
Network parameters	
Module [59, 61]	A set of nodes that share more edges than expected by chance. Modules reveal organization at a coarse level and are typically interpreted as functional components.
Assortativity [62]	Correlation between nodes that measures their preference to connect to others that are similar or different in some way. For example, 'neighbor connectivity' is the average degree of neighbors of a node with a given degree.
Centralization	Normalized difference between the largest and average connectivities.
Density	The average connectivity over all nodes.
Diameter	Maximum eccentricity over all nodes.
Radius	Minimum eccentricity over all nodes.

To create an HP, network structural parameters (Table 1) used by rules are initially calculated (e.g. connectivity, clustering coefficient, etc.). Parameters are selected to suit the purpose of the layout (e.g. clustering coefficient distinguishes hubs and clusters). Next, the rules are applied to each node in the network to assign it to an axis and determine its coordinate. Axis assignment rules are typically Boolean tests such as 'is the node a sink?' or 'is the node's ''clustering coefficient'' smaller than 0.5?'. It is up to the user to define rules that create a unique assignment and rules can be a function of any number of structural parameters. Node coordinates are typically derived from the absolute or rank-ordered value of a node parameter, such as connectivity. Once axis assignment and node coordinates have been computed, the HP is drawn with axes as radial linear segments, nodes as glyphs and edges as curves between corresponding node positions. The final format of the layout is flexible and formatting of components such as node glyphs and edges is addressable. Nodes with the same position on the layout may be disambiguated with labels, a repelling scheme to spread the nodes out locally, or density maps [56]. We demonstrate how these rules are operationalized to create HPs from directed and undirected networks using small example networks (Figure 3).

Figure 3A shows a typical directed network, rendered as a parallel-coordinate plot in Figure 3B and HP in Figure 3C. The assignment of nodes to the $x1$, $x2$ and $x3$ axes is based on the directionality of edges and node coordinates are a function of connectivity (total number of edges). We use the terminology from ref. [63] to refer to sources (out edges only) as 'regulators', sinks (in edges only) as 'workhorses' and the other nodes as 'managers'. The layout in Figure 3C is natural for directed networks those in which nodes can be similarly partitioned by structure, function or annotation.

In an undirected network, axis assignment can be based on a structural quantity, such as the 'clustering coefficient' (cc), which characterizes the interconnectivity of a node's neighbors. In Figure 3D, we show a representative undirected gene-disease network, in which genes implicated in the same diseases are connected. The corresponding HP in Figure 3E partitions nodes by cc value ($cc = 0$ to $x1$, $0 < cc < 1$ to $x2$, and $cc = 1$ to $x3$) and uses connectivity for node coordinates. The $x1$-axis is cloned to reveal connections between $cc = 0$ nodes (when an axis is cloned, the same position on both copies

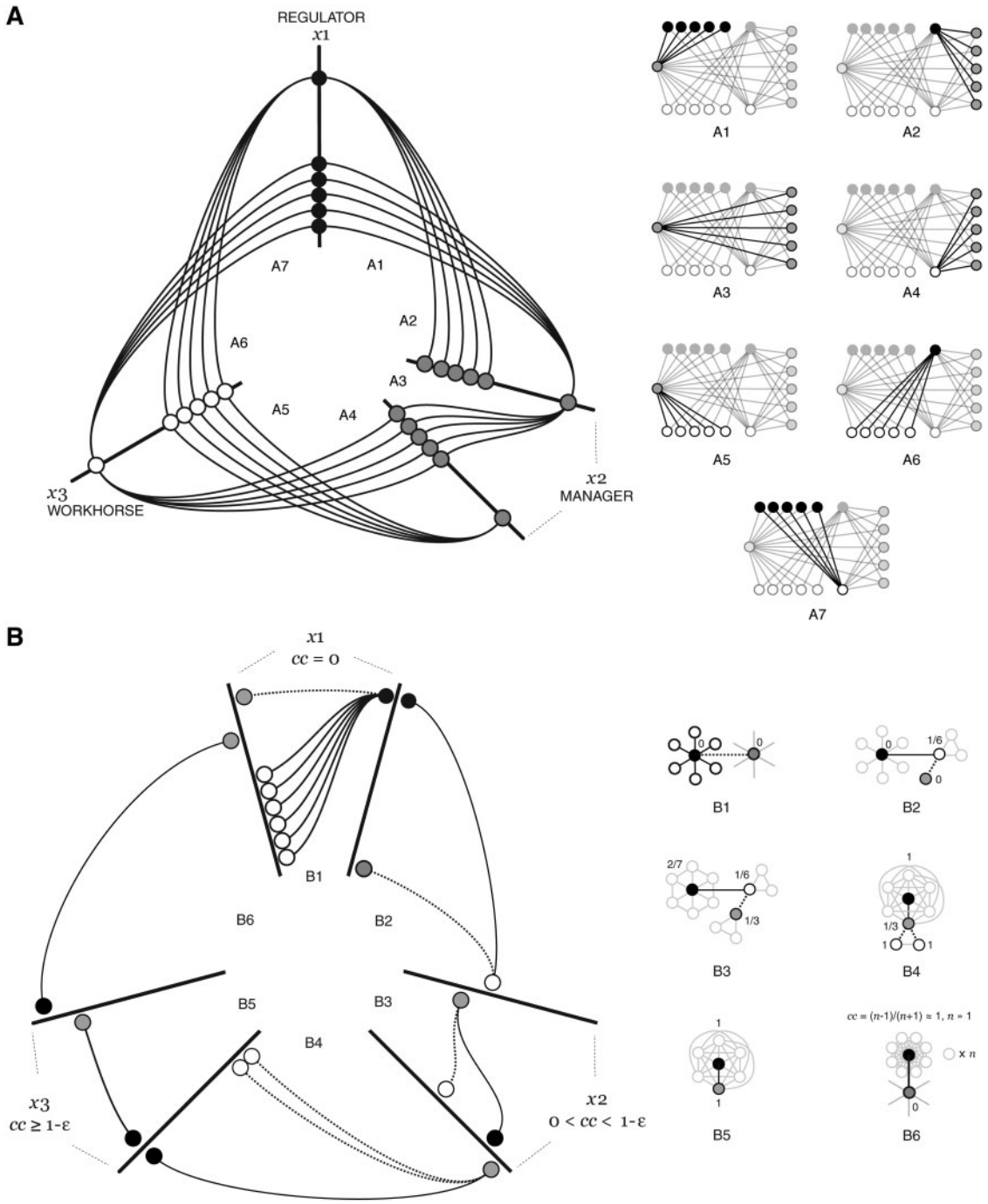

Figure 4: Location of underlying network structures on HPs. (**A**) HP of a directed network (Figure 2A) using rules from Figure 3B. Each bundle of edges (AI–A7) is identified in the manual layout. (**B**) Position of hubs and clusters in an HP that uses the clustering coefficient (Figure 3E). Hubs have cc = 0 and appear between *x*1-axis clones (BI), whereas nodes in clusters (cc = 1−ε) show up within the *x*3-axis (B5). Links in region B6 identify nodes on *x*1-axis that connect distinct clusters.

corresponds to the same node). For an undirected network, the orientation of the edges between axis clones is arbitrary, and in our implementation, we reverse the links where required for outward clockwise orientation. In a directed network, this direction can encode the orientation of the edge.

In Figure 4, we relate node and edge positions of HPs in Figure 3C and E to underlying network structures. Figure 4A shows an HP the small directed

network in Figure 2A, using the same rules as for the HP of *E. coli* network in Figure 3C. Each bundle of edges corresponds to relationships between nodes with different structural profiles. For example, bundle A6 represents connections between highly connected regulators and weakly connected workhorses. The corresponding visual map for an undirected network is shown in Figure 4B, which shares the format of Figure 3E. For example, connections between highly connected clusters (modules) and hubs are represented by bundle B6 in Figure 4B.

The axes in an HP have adjustable orientation (inward/outward), size (length may be normalized) and scale (linear/log/rank ordered). Typically, HPs have three axes with a uniform radial distribution (Figure 3C), which prevents edges from crossing axes. Axes can be cloned (Figure 3E) to show connections between nodes assigned to the same axis. Hive plots with three axes accommodate edges between each axis pair without crossing the other axis. In general, this condition can be achieved with more than three axes if nodes can be partitioned to axes in a way that allows for an order in which nodes are connected only to those on neighboring axes. If this is not possible, axes must be duplicated at multiple positions, or edges must be routed across or around other axes. This negatively impacts the interpretability of the figure and should be avoided.

Detecting and displaying densely connected components, or modules [64, 65], is traditionally challenging because layout algorithms typically cannot be adjusted to different module detection rules. Many algorithms natively reveal clusters of nodes (Figure 7C), but those may overlap (clusters 2 and 13). HPs can mitigate this by dividing each axis into disjoint segments to which nodes from modules are assigned. Intramodule connections would appear as concentric sets of edges, connecting the module's corresponding segment on each axis (Figure 5). This approach is similar to Semantic Substrates [46], but with the module membership as attribute. This scheme can also encode membership in strongly connected components, except that, by definition, there would be no connections between concentric regions.

APPLICATIONS OF HP

Studying network structure

Directed networks case study—gene regulatory network

To show how HPs can be used to analyze directed networks, we use the RegulonDB *E. coli* gene regulatory network [38, 39], which comprises relationships between transcription factors and genes classified by function (repressor, activator, dual) and experimental certainty (certain, uncertain, unknown). With 1584 nodes, the network is too complex for its layout to be informative and neither hierarchical nor force-based layouts (Figure 6A) help answer fundamental questions like (i) what is the relative proportion of each node type? (ii) what are the differences in connectivity patterns between each layer? and (iii) what is the role of the manager genes

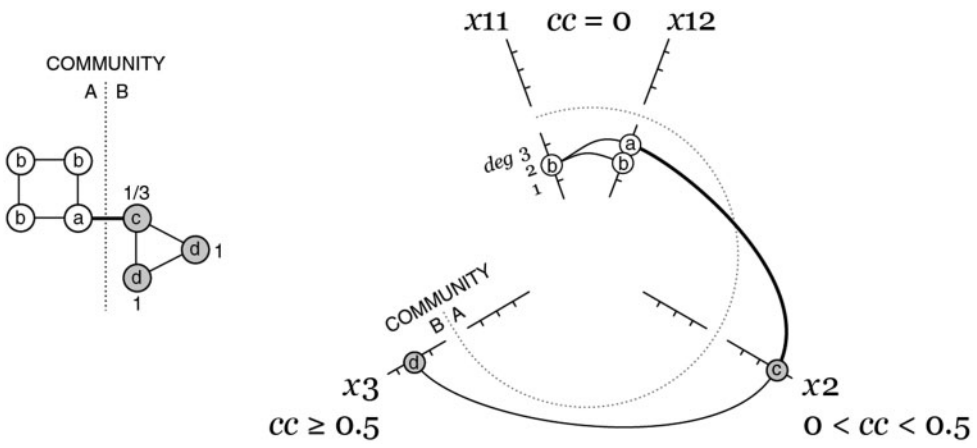

Figure 5: HPS can represent communities by assigning nodes from each community to a different axis segment. In this plot, the two interconnected clusters in a small network are shown by segmenting each axis into two. The link that connects the clusters appears between segments at different radial positions. Where not zero, *cc* are shown beside the node.

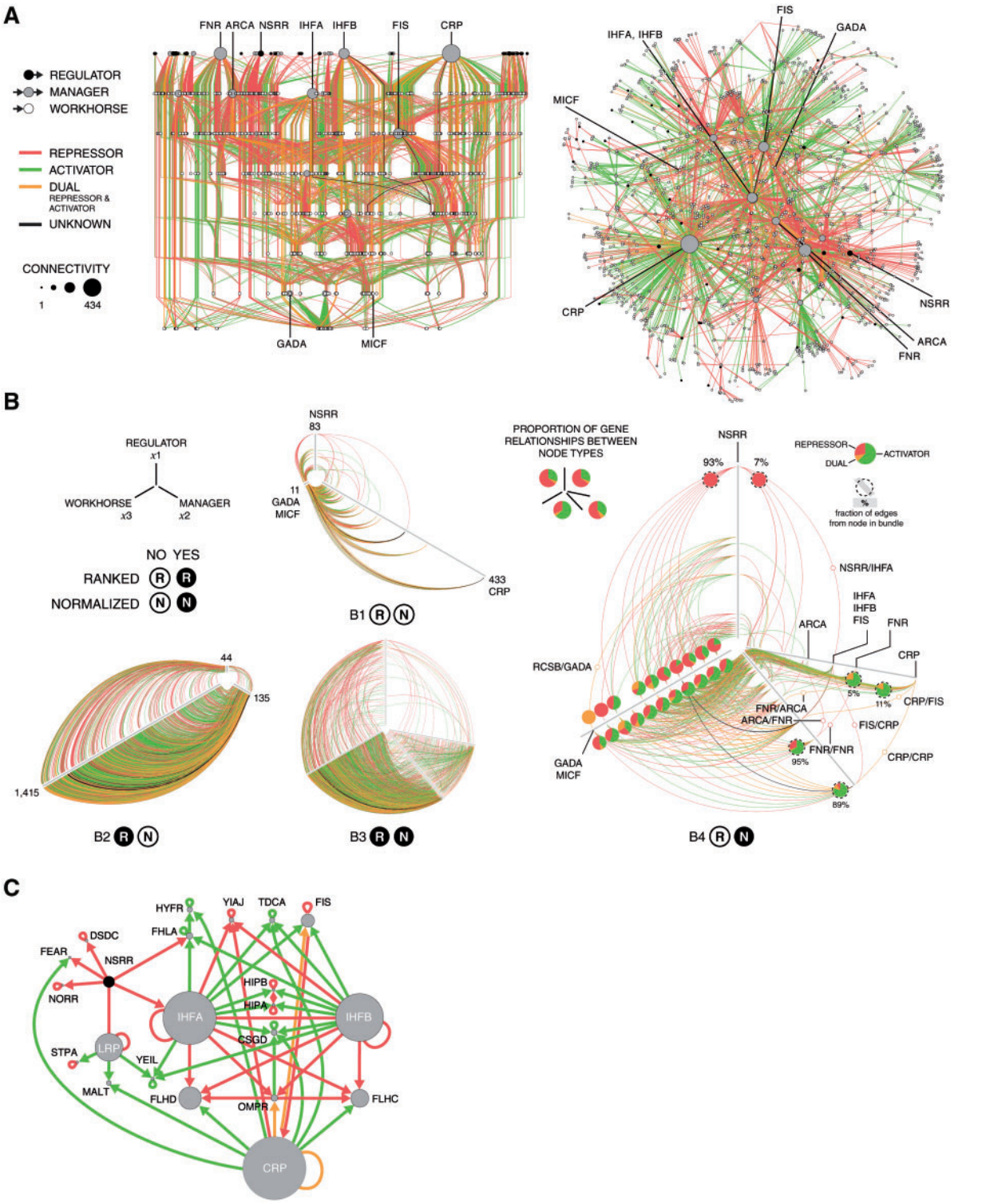

Figure 6: Application of HPs to a large directed network. (**A**) RegulonDB [37] network of 1584 regulating genes in *E. coli* generated with Cytoscape [39] using yFiles Sugyiama (left) and spring embedded layouts (right). Node size encodes connectivity and edge color encodes the function of the relationship. Genes mentioned in the text are labeled. (**B**) Four HPs of RegulonDB generated with rules from Figure 3B. The directionality of the edges between cloned axes in B4 is clockwise. Pie charts show the proportion of relationship types within an edge bundle and the fraction of edges from a node in a bundle is identified with a dotted circle. (**C**) Manual layout of the neighborhood of managers at NSRR. Node sizes reflect the number of workhorses controlled by each gene.

in the network? These layouts occlude the identity of structurally unique nodes (e.g. most connected regulator or manager) and their roles (e.g. is the most connected regulator a repressor, activator or both?).

We show how HPs can be used to address such questions in Figure 6B, in which layouts vary only by axis scaling and normalization. The absolute scale, unnormalized HP [Figure 6(B1)] shows the maximum connectivity of the node types, demonstrating that managers are the most connected nodes (CRP, $deg = 433$), then regulators (NSRR, $deg = 83$) and then workhorses (GADA, MICF, $deg = 11$). When connectivity rank is used as node coordinate [Figure 6(B2)], the axis length becomes proportional to the number of nodes, revealing a preponderance of workhorses (1415) to managers (135) and regulators (44). The overall distribution of edges within the layers of the network can be observed in the normalized rank-ordered HP in Figure 6(B3), which demonstrates that connections between highly connected managers and workhorses have a larger proportion of dual-role edges than from sparsely connected managers.

The most revealing HP is Figure 6(B4), where node coordinates are based on absolute connectivity but axis length is normalized. This HP neatly accommodates annotation, such as pie charts to indicate the proportion of each type of edge within bundles of edges. We see that the most connected regulator (NSRR) is a strict repressor and has 93% of its connections to workhorses and only 7% to managers. The proportion of regulator/manager and regulator/workhorse relationships appears to be very similar (30% activators, 65% repressors and 5% dual), indicating that purpose of the top layer in the network is to provide negative feedback. NSRR connects to both busy and sparse workhorses, but preferentially to only sparse managers with the exception of IHFA (third most connected manager). The two busiest managers, FNR and CRP, are shielded from the top layer of the network (there are no connections from regulators) and independent (they are not connected to each other) and exert a primarily activating influence.

This HP also reveals that highly connected managers predominately have a repressing effect on themselves. CRP is self-connected in dual capacity, as shown by the orange line between the outermost ends of the cloned manager axis, and is affected by only one other gene (FIS), repressively. FNR, the second most connected manager, is repressed by itself and ARCA. These top manager/manager relationships are distinguished from the generally activating relationships between the top managers (FNR and CRP) and workhorses or sparsely connected managers, as indicated by the pie charts in the bundles from these genes. Moreover, relatively few top manager connections are to other managers (5% FNR/manager, 11% CRP/manager) with most of the connections made to workhorses (95% FNR/workhorse, 89% CRP/workhorse).

To emphasize that the challenges of conventional layouts apply to even small networks, we show a manual layout of the relationship of NSRR, the busiest regulator, with its neighboring managers in Figure 6C. This layout was laboriously created to clarify the structure of this region and reveal the symmetry of its connections, but fundamental patterns remain slow to glean. For example, to notice that NSRR does not connect to the most connected manager (CRP), one first has to find NSRR (or CRP) in the layout and then scan all of the outgoing (or incoming) edges to detect the connection. In the HP, this observation is made instantly because NSRR and CRP, being the most connected nodes of their type, are always found at the ends of their respective axes.

Undirected networks case study—cancer gene network

To show how HPs can be applied to undirected networks, we use the human disease network (HDN) [38], which associates 1284 unique disease identifiers with 3823 gene symbols through 6275 associations maintained in the Morbid Map of OMIM [10]. It is analogous to protein–protein interaction networks, but with genes in place of proteins and diseases in place of interaction evidence. The network connects an average of 4.9 genes per disease (most connected disease is deafness, with 113 gene symbols) and 1.6 diseases per gene symbol (most connected gene is TP53, implicated in 11 diseases). Each disease is further categorized into one of 23 systems, such as cancer, neurological and muscular. For this case study, we use HPs to visualize relationships between genes by constructing the disease gene network (DGN) [38]. The DGN is derived from the HDN by connecting genes associated with the same one or more diseases (Figure 3D) and resolving synonymous gene symbols. It comprises 1934 genes

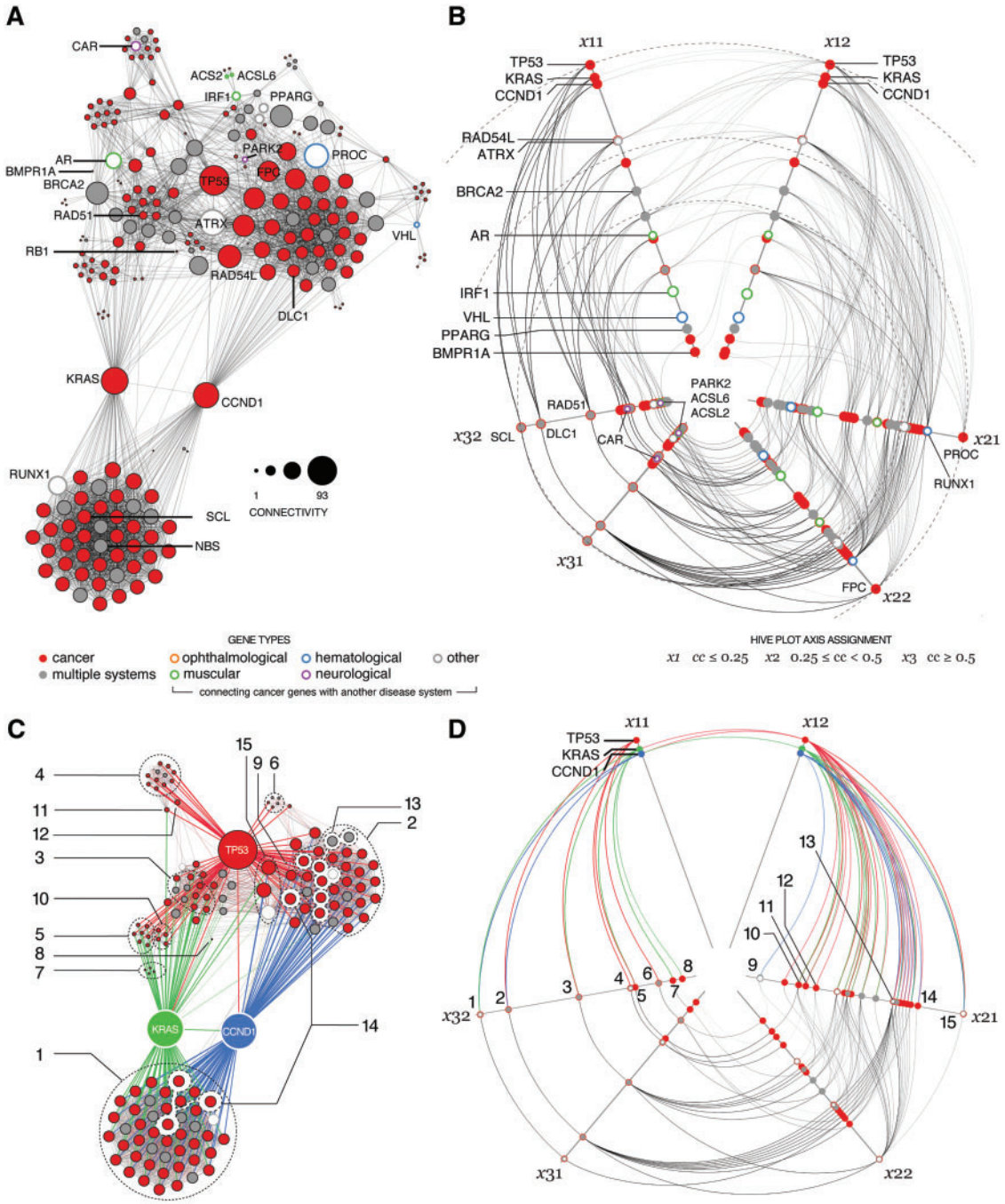

Figure 7: Application of HPs to a large undirected network. (**A**) Force atlas layout generated with Gephi [25] of the cancer component of the gene-disease network showing 258 genes and 3057 edges. Highly connected genes are labeled. Genes that connect the cancer subsystem to other systems are represented by hollow glyphs, colored by the target system. (**B**) HP of the network, with overlapping nodes drawn in decreasing size by color frequency to maintain visibility. Clustering coefficient ranges for the axes are *x*1: $cc < 0.25$, *x*2: $0.25 \leq cc < 0.5$, *x*3: $cc \geq 0.5$, chosen to effectively separate the nodes. (**C**) Force atlas layout of genes TP53, KRAS and CCNDI, their immediate neighbors in the network and any genes connecting the neighbors. (**D**) Corresponding HP, with axis length normalized. Corresponding clusters, defined as nodes with the same connectivity and clustering coefficient, are identified with an index in (C) and (D).

and 13 235 connections. To show how HP can be used to compare networks, we will study individual components of the DGN composed of genes associated by diseases categorized under the same system (e.g. cancer).

Figure 7A shows the cancer DGN with the corresponding HP shown in Figure 7B. The HPs of the DGN are created by assigning nodes to axes based on the clustering coefficient and using connectivity for their coordinates (Figure 3D). The locations of structures in this plot are shown in Figure 4B: 'hubs' ($x1$-axis), 'highly connected components' ($x2$-axis) and 'clusters' ($x3$-axis). The traditional layout in Figure 7A suggests that CCND1 and KRAS play similar roles in the network, connecting a large cluster containing RUNX1 and SCL to the rest of the network, details are difficult to determine because density of nodes in the layout is high. The connectivity pattern of TP53, a central node and several nodes connecting the cancer component to other systems (AR and IRF1 to muscular system, PROC and VHL to hematological system and CAR and PARK2 to neurological system) is also difficult to assess.

The HP in Figure 7B identifies TP53, KRAS and CCND1 as the major hubs in the network, indicating that TP53 is the most connected. The fact that TP53 is connected to both KRAS and CCND1 is clear from the connections within the $x1$-axis region, but not possible to determine from the layout in Figure 7A. IRF1 and VHL which connect the cancer to muscular and hematological systems, respectively, appear to have about 1/2 of the connectivity of the largest cluster and CAR. The latter connects to the neurological system and is identified within a cluster with connectivity similar to VHL.

The HP reveals that RUNX1, which appears on the $x2$-axis, is not part of the largest cluster, a fact obscured in the conventional layout. The number of clusters with distinct connectivity can be discerned by counting the number of node groups on the $x3$-axis. The cluster that contains CAR, which connects to the neurological system, is the largest that connects cancer to another system. PARK2, ACS2 and ACSL6 are also parts of clusters, but smaller ones. The HP makes possible visual identification of how genes that are involved in another system are integrated into the cancer component.

Reduction is frequently used to assist in visualizing large networks, where nodes with similar structure are combined (e.g. clusters) or removed (e.g. leaves) to reduce complexity [22]. HPs automate this strategy as demonstrated in Figures 7C and 6D, which limit the cancer subsystem to genes immediately adjacent to TP53, KRAS and CCND1, or interconnecting their neighbors. Comparing Figure 7A and C, we see that layouts of large networks occlude patterns, which may emerge when the layout is limited to fewer nodes. In the HP in Figure 7D, structurally distinct sets of nodes are independent and can be easily identified and enumerated because their location is based on cluster size and extent of internal connectivity. The membership of clusters can be easily read out when labels are employed in the HP (suppressed in this example for clarity). In contrast, unambiguously identifying clusters in Figure 7C is not possible. For example, there are nodes within the layout area of the largest cluster (no.1) that do not belong to the cluster (identified by group no.14). Similarly, nodes within the apparent boundary of cluster no.2 belong elsewhere (no.13) and the boundary cluster no.3 cannot be easily discerned.

Comparing network structure

Case study—functional systems with disease gene network

HPs can be directly compared to demonstrate differences between networks because they have a fixed coordinate system. We show plots for the largest connected component in the ophthalmological, muscular, hematological and neurological systems in Figure 8, created by calculating network properties for each subsystem in isolation from the rest of the network. The plots reveal the internal structure of each subsystem, without influence from edges that embed the system in the rest of the network. Both the conventional layouts and HPs provide information about the community structure of each system, but quantitative information about connectivity is only available through the HPs.

While the density of the conventional layouts makes resolving clusters and identifying hubs within each network difficult, in the HPs the process is as simple as looking at the start of the $x1$-axis (leaves), end of the $x1$-axis (hubs) and $x3$-axis (clusters). The HPs immediately reveal the absence of leaves or hubs in the ophthalmological and muscular

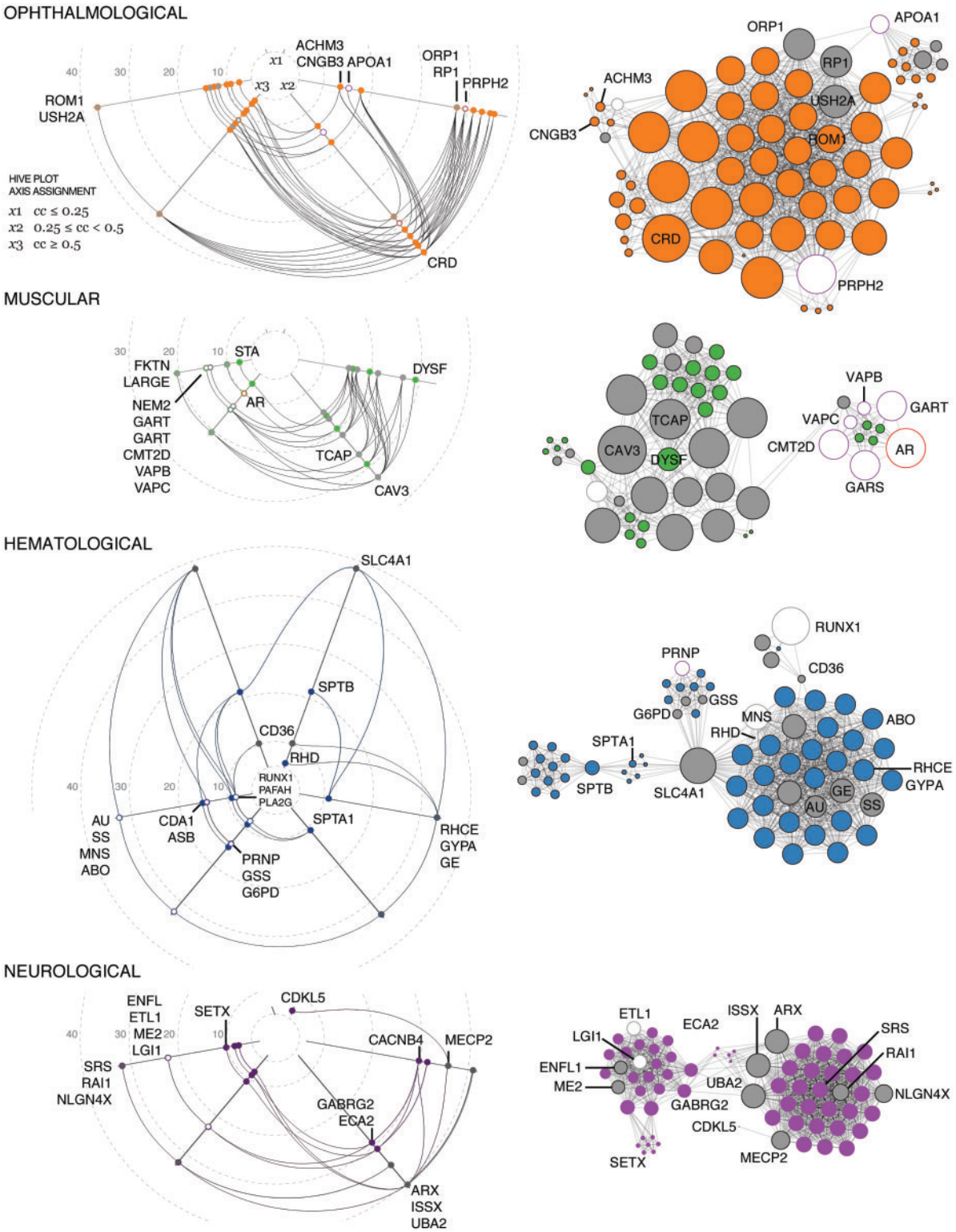

Figure 8: Comparison of largest connected components of ophthalmological, muscular, hematological and neurological subsystems in the DGN, drawn with the same layout rules as Figures 7A and 6B. Node size encodes the connectivity of the gene in the entire DGN. HP grid shows connectivity in steps of 10 edges.

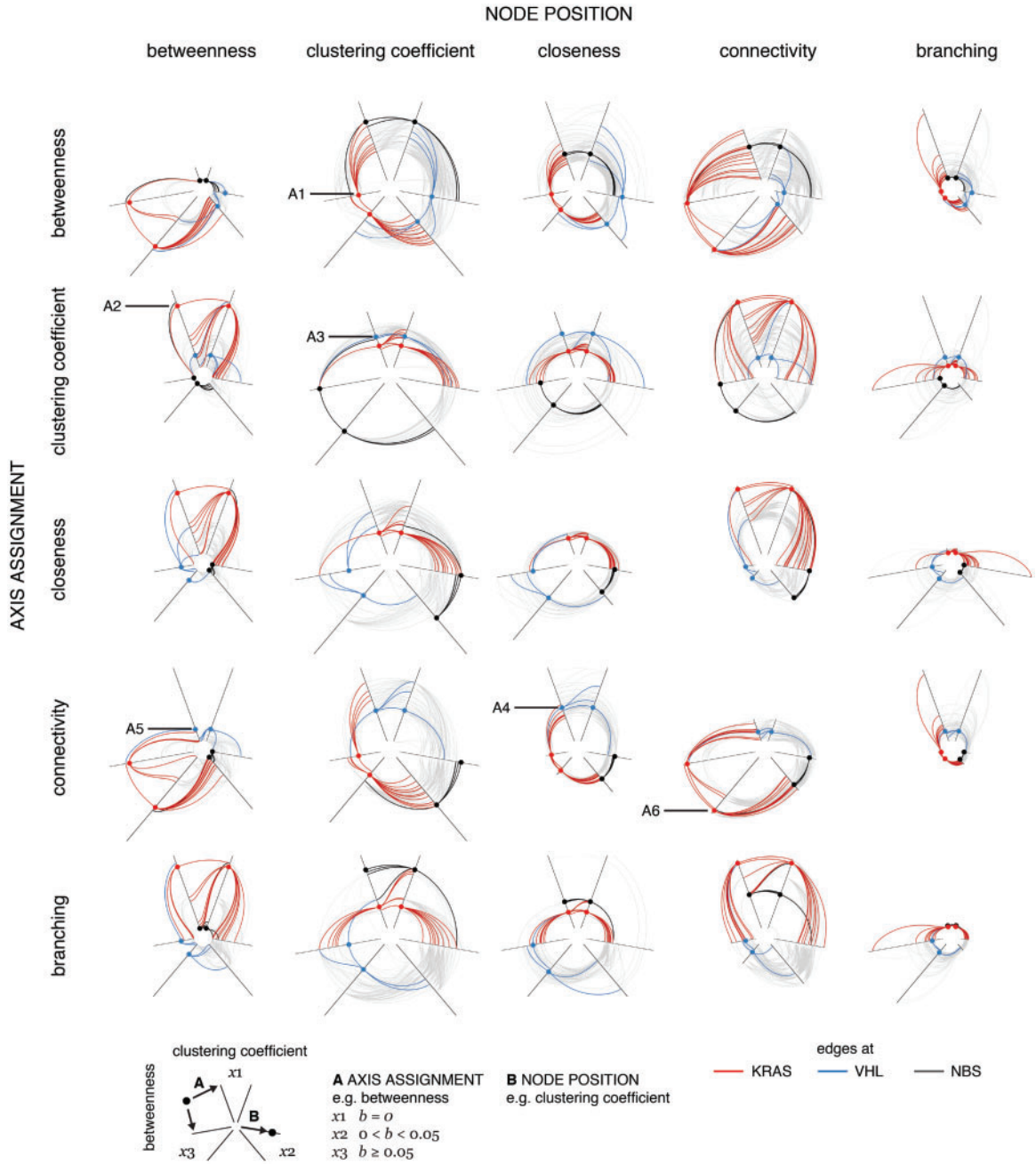

Figure 9: Demonstration of correlation between pairs of structural components in a network using a hive panel. The cancer DGN from Figure 7A is shown as a matrix of HPs. Links to KRAS, VHL and NBS are highlighted to illustrate how a panel can be used to distinguish nodes of this kind (KRAS mediates connections between two clusters, VHL is a sparsely connected gene at the edge of the network and NBS is within the largest fully connected cluster). Each HP uses a unique combination of parameters for axis assignment (shown in rows) and node placement (shown in columns) from the set: betweenness (*b*), clustering coefficient (cc), closeness (*c*, remapped to $c-1$), connectivity (*deg*) and branching (*nn*/*n*, ratio of next-neighbors to neighbors). For example, the plot in row betweenness (*b*) and column clustering coefficient (cc) uses *b* for axis assignment and cc for node placement. Parameter cutoffs for *b*, cc, *c*, *deg* and *nn*/*n* used to determine node axis assignment to *x*1, *x*2 and *x*3-axes for are *b*:(0, 0–0.05, >0.05), cc:(0–0.7, 0.7–1, 1), *c*:(<2, 2–2.5, >2.5), *deg*:(<30, 30–60, >60) and *nn*/*n*:(<2, 2–5, >5). These cutoffs were selected heuristically to efficiently populate all axes of the graph.

subsystem (no nodes on $x1$-axis). The neurological subsystem has a single leaf, CDKL5, which is difficult to find in the conventional layout. The hematological system is seen to have one highly connected hub (SLC4A1) and two moderately connected ones (SPTB and CD36), with the connectivity ratios clearly evident (about 5:2:1). Furthermore, the fact that SLC4A1 and SPTB are connected is clear in the HPs, but hidden by the cluster of nodes between them in the conventional layout. Near absence of nodes on $x2$-axis of the hematological HP emphasizes that it is composed almost entirely of hubs and clusters, and HPs show that RHCE and GYPA are not fully interconnected to the cluster in which they are placed by the conventional layout.

The quantitative axes of the HP allow us to determine that the connectivity of SLC4A1 is nearly 50, 20% larger than the next most connected nodes in the systems, which are the ROM1/USH2A cluster in the ophthalmological system. The largest cluster of all four networks is in the ophthalmological subsystem with a connectivity of 37. We can also quickly see that the SRS/RA1 neurological and AU/SS hematological clusters are structurally equivalent— they are the same size and both connect to the most connected node on the $x2$-axis in their respective plots.

Hive panels—network dashboard

HPs that use the same rules can be used to compare networks, whereas those that use a variety rules can provide different perspectives on one network. Figure 9 shows a 25-plot hive panel of the cancer DGN which uses 'betweenness' (b, fraction of a graph's shortest paths on which a node lies), 'clustering coefficient' (cc), 'closeness' (c, average distance from a node to all nodes, using c-1 as the quantity for axis position), 'connectivity' (deg) and 'branching' (nn/n, ratio of next neighbors to neighbors) to reveal different aspects of the network. In each panel, we highlight connections to KRAS, VHL and NBS, which are genes with different network characteristics.

In the b/cc plot (row 'betweenness', column 'clustering coefficient') KRAS [Figure 9(A1)] is seen on the $x3$-axis because it has a high 'betweenness' (on the shortest path between many node pairs), but closer to the center of the HP, because it has a low cc (it is a hub). Nodes in this region of this plot divide a network into individual communities. In the cc/b plot KRAS [Figure 9(A2)] is seen to be on the $x1$-axis (low cc) and far along the b axis (high b), in symmetry to the b/cc plot. VHL is seen to be weakly connected [Figure 9(A4), low connectivity] and at the edge of the network [Figure 9(A4), high c]. This is emphasized by VHL's position on the cc/b plot [Figure 9(A5)], in which it is seen to have a low value of b, indicating it is on few shortest paths.

Plots on the diagonal, a hive panel, are used to stratify the nodes based on a single parameter. For example, among the nodes with low cc in the cc/cc, we see that VHL has a slightly higher cc than KRAS [0.26 versus 0.18, Figure 9(A3)]. Similarly, of all the highly connected nodes ($x3$-axis in deg/deg plot), KRAS is at the end of the axis [Figure 9(A6)].

Some of the quantities in this hive panel are related (connectivity, betweenness and closeness are all measures of centrality), and it is up to the user to decide which parameters capture the essential characteristics of the network and are relevant to communicate. The panel presentation is particularly suitable for an interactive software interface, in which a user can search for network components, or select them directly and see the elements highlighted in each panel to compare them with the rest of the network across many parameters.

Performance

Computational time required to generate an HP layout is largely determined by the speed of algorithms used to calculate network properties used by axis assignment and node placement rules. Some properties are quick to calculate, such as degree centrality [$O(n^2)$, in adjacency matrix representation], while others are more computationally intensive, such as clustering coefficient [$O(n^3)$], [66] and betweenness [$O(n^3)$], [57], where n is the number of nodes in the network. The Perl implementation of HPs, which uses the Graph module to parse and analyze networks, takes about 20 s (Macbook Pro i7, 2.66 GHz) to compute connectivity and cluster coefficient statistics for a 1000 node network with 5000 edges. Once these statistics are calculated, plots of the type seen in Figure 7B can be generated in less than a second.

CONCLUSIONS

To date, network visualization has lacked a quantitative, visually parsable and scalable approach to

visually assess and compare large networks. HPs provide a solution to bridge the gap between assessing and comparing structures of large networks and studying fine structure of small components of interest. As a layout algorithm, hive plots can be tuned to answer specific questions of interest to the experimenter. Finally, they are computationally simpler than traditional layout algorithms and therefore, amenable to responsive interactive interfaces through a plugin.

Implementation

HPs in this article were generated with a Perl implementation, which can be downloaded from http://www.hiveplot.com. An R implementation of 2D and 3D hive plots has been created by Bryan Hanson and is available at http://academic.depauw.edu/~hanson/HiveR/HiveR.html.

Key Points

- Hive plots introduce a rational method of network visualization in which nodes are placed on radial axes, with coordinates determined solely by structural properties of the network.
- Hive plots are flexible: node placement rules can be tuned to increase sensitivity to specific patterns.
- Hive plots are predictable: for a given set of rules, the same structures always have the same layout.
- Hive plots are comparable: differences in layouts is proportional to differences in corresponding networks.
- Hive plots are transparent and practical: rational rules make the layout easy to understand, interpret and implement.

Acknowledgements

M.K. would like to acknowledge helpful discussions with Steven Hallam, Benjamin Good, Olena Morozova for access to experimental data sets and Brian Turner for an online prototype, http://wodaklab.org/hivegraph/graph/index. Extended acknowledgement to Bryan Hanson for releasing the R package in time for this publication (http://academic.depauw.edu/~hanson/HiveR/HiveR.html).

FUNDING

The authors gratefully acknowledge funding support from the BC Cancer Foundation and BC Cancer Agency.

References

1. Micheli-Tzanakou E, Wuchty S, Ravasz E, *et al*. The architecture of biological networks. In: Deisboeck TS, Kresh JY, (eds). *Complex Systems Science in Biomedicine*. US: Springer, 2006;165–81.
2. Newman MEJ. *Networks: An introduction*. US: Oxford University Press, 2010.
3. Pavlopoulos G, Secrier M, Moschopoulos C, *et al*. Using graph theory to analyze biological networks. *BioData Mining* 2011;**4**(1):10.
4. Schlitt T, Brazma A. Current approaches to gene regulatory network modelling. *BMC Bioinformatics* 2007;**8(Suppl 6)**: S9.
5. Kar G, Gursoy A, Keskin O. Human cancer protein-protein interaction network: a structural perspective. *PLoS Comput Biol* 2009;**5**(12):e1000601.
6. Stelzl U, Worm U, Lalowski M, *et al*. A human protein-protein interaction network: a resource for annotating the proteome. *Cell* 2005;**122**(6):957–68.
7. Kanehisa M, Goto S, Kawashima S, *et al*. The kegg resource for deciphering the genome. *Nucleic Acids Res* 2004; **32**(Database issue):D277–80.
8. Ma HW, Zeng AP. The connectivity structure, giant strong component and centrality of metabolic networks. *Bioinformatics* 2003;**19**(11):1423–30.
9. Amberger J, Bocchini CA, Scott AF, *et al*. Mckusick's online mendelian inheritance in man (omim). *Nucleic Acids Res* 2009;**37**(Database issue):D793–6.
10. Simpson JT, Wong K, Jackman SD, *et al*. Abyss: A parallel assembler for short read sequence data. *Genome Res* 2009; **19**(6):1117–23.
11. Pevzner PA, Tang H, Waterman MS. An eulerian path approach to DNA fragment assembly. *Proc Natl Acad Sci USA* 2001;**98**(17):9748–53.
12. Pevzner PA, Tang H, Tesler G. De novo repeat classification and fragment assembly. *Genome Res* 2004; **14**(9):1786–96.
13. Kitano H. Systems biology: a brief overview. *Science* 2002; **295**(5560):1662–4.
14. Aderem A. Systems biology: its practice and challenges. *Cell* 2005;**121**(4):511–3.
15. Kitano H. Computational systems biology. *Nature* 2002; **420**(6912):206–10.
16. Goodacre R, Vaidyanathan S, Dunn WB, *et al*. Metabolomics by numbers: acquiring and understanding global metabolite data. *Trends Biotechnol* 2004;**22**(5):245–52.
17. Sun J, Zhao Z. A comparative study of cancer proteins in the human protein-protein interaction network. *BMC Genomics* 2010;**11(Suppl 3)**:S5.
18. Mason O, Verwoerd M. Graph theory and networks in biology. *IET Syst Biol* 2007;**1**(2):89–119.
19. Aittokallio T, Schwikowski B. Graph-based methods for analysing networks in cell biology. *Brief Bioinform* 2006; **7**(3):243–55.
20. Gehlenborg N, O'Donoghue SI, Baliga NS, *et al*. Visualization of omics data for systems biology. *Nat Methods* 2010;**7(Suppl 3)**:S56–68.
21. Suderman M, Hallett M. Tools for visually exploring biological networks. *Bioinformatics* 2007;**23**(20):2651–9.
22. Baitaluk M, Sedova M, Ray A, *et al*. Biologicalnetworks: visualization and analysis tool for systems biology. *Nucleic Acids Res* 2006;**34**(Web Server issue):W466–71.
23. Smoot ME, Ono K, Ruscheinski J, *et al*. Cytoscape 2.8: New features for data integration and network visualization. *Bioinformatics* 2011;**27**(3):431–2.

24. Shannon P, Markiel A, Ozier O, *et al*. Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome Res* 2003;**13**(11): 2498–504.
25. Bastian M, Heymann S, Jacomy M. Gephi: an open source software for exploring and manipulating networks. In: *International AAAI Conference on Weblogs and Social Media: 2009*. San Jose, CA: AAAI Publications, 2009;361–2.
26. Henry N, Fekete JD. Matrixexplorer: a dual-representation system to explore social networks. *IEEE Trans Visu Comput Graph* 2006;**12**(5):677–84.
27. Breitkreutz BJ, Stark C, Tyers M. Osprey: a network visualization system. *Genome Biol* 2003;**4**(3):R22.
28. Jensen LJ, Kuhn M, Stark M, *et al*. String 8 - a global view on proteins and their functional interactions in 630 organisms. *Nucleic Acids Res* 2009;**37(Suppl 1)**: D412–6.
29. Kaufmann M, Wagner D, (eds). *Drawing Graphs: Methods and Models*. Berlin Heidelberg: Springer, 2001.
30. Fruchterman TMJ, Reingold EM. Graph drawing by force-directed placement. *Software Pract Exper* 1991;**21**(11): 1129–64.
31. Martin S, Brown WM, Klavans R, *et al*. Openord: an open-source toolbox for large graph layout. In: *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference, San Francisco, CA, 2011*. SPIE (Bellingham, Washington) and IS&T (Springfield, Virginia), 786806–11.
32. Sugiyama K, Tagawa S, Toda M. Methods for visual understandings of hierarchical system structures. *IEEE Trans Syst Man Cybernetics* 1981;**smc-11**(2):109–25.
33. Eiglsperger M, Siebenhaller M, Kaufmann M. An efficient implementation of sugiyama's algorithm for layered graph drawing. *J Graph Algorithms Appl* 2005;**9**(3):305–25.
34. Evans JA, Foster JG. Metaknowledge. *Science* 2011; **331**(6018):721–5.
35. Rual JF, Venkatesan K, Hao T, *et al*. Towards a proteome-scale map of the human protein-protein interaction network. *Nature* 2005;**437**(7062):1173–8.
36. Gama-Castro S, Salgado H, Peralta-Gil M, *et al*. Regulondb version 7.0: Transcriptional regulation of escherichia coli k-12 integrated within genetic sensory response units (gensor units). *Nucleic Acids Res* 2011;**39**:D98–105.
37. Huerta AM, Salgado H, Thieffry D, *et al*. Regulondb: a database on transcriptional regulation in escherichia coli. *Nucleic Acids Res* 1998;**26**(1):55–9.
38. Goh KI, Cusick ME, Valle D, *et al*. The human disease network. *Proc Natl Acad Sci USA* 2007;**104**(21):8685–90.
39. Kohl M, Wiese S, Warscheid B. Cytoscape: software for visualization and analysis of biological networks. *Methods Mol Biol* 2011;**696**:291–303.
40. Coleman MK, Parker DS. Aesthetics-based graph layout for human consumption. *Softw Pract Exp* 1996;**26**(12): 1415–38.
41. Wegman EJ. Hyperdimensional data analysis using parallel coordinates. *J Am Stat Assoc* 1990;**85**(411):664–75.
42. Amadou C, Pascal G, Mangenot S, *et al*. Genome sequence of the beta-rhizobium cupriavidus taiwanensis and comparative genomics of rhizobia. *Genome Res* 2008;**18**(9): 1472–83.
43. Fukui T, Atomi H, Kanai T, *et al*. Complete genome sequence of the hyperthermophilic archaeon thermococcus kodakaraensis kod1 and comparison with pyrococcus genomes. *Genome Res* 2005;**15**(3):352–63.
44. Thomson NR, Clayton DJ, Windhorst D, *et al*. Comparative genome analysis of salmonella enteritidis pt4 and salmonella gallinarum 287/91 provides insights into evolutionary and host adaptation pathways. *Genome Res* 2008;**18**(10):1624–37.
45. Wattenberg M. Visual exploration of multivariate graphs. In: *Proceedings of the SIGCHI conference on Human Factors in computing systems: 2006*. New York, NY: ACM Press, 2006, 811–819.
46. Aris A, Shneiderman B. Designing semantic substrates for visual network exploration. *Information Visualization* 2007; **6**(4):281–300.
47. Lieberman MD, Taheri S, Guo H, *et al*. Visual exploration across biomedical databases. *IEEE/ACM Trans Comput Biol Bioinform* 2011;**8**(2):536–50.
48. Bachmaier C, Brandenburg F, Brunner W, *et al*. Coordinate assignment for cyclic level graphs. In: Ngo H, (ed). *Computing and Combinatorics*, Vol. 5609. Berlin/Heidelberg: Springer, 2009, 66–75.
49. Yfiles. http://www.yworks.com/en/products_yfiles_about.html (11 August 2011, date last accessed).
50. Schreiber F, Dwyer T, Marriott K, *et al*. A generic algorithm for layout of biological networks. *BMC Bioinformatics* 2009; **10**:375.
51. Kita Y, Takahashi T, Uozumi N, *et al*. Pathway-oriented profiling of lipid mediators in macrophages. *Biochem Biophys Res Commun* 2005;**330**(3):898–906.
52. Shimamura T, Imoto S, Yamaguchi R, *et al*. Inferring dynamic gene networks under varying conditions for transcriptomic network comparison. *Bioinformatics* 2010;**26**(8): 1064–72.
53. Mandakova T, Joly S, Krzywinski M, *et al*. Fast diploidization in close mesopolyploid relatives of arabidopsis. *Plant Cell* 2010;**22**(7):2277–90.
54. Castellarin M, Warren RL, Freeman JD, *et al*. Fusobacterium nucleatum infection is prevalent in human colorectal carcinoma. *Genome Res* 2011. doi/10.1101/gr.126516.111, Advance Access publication 18 October 2011.
55. Krzywinski M, Schein J, Birol I, *et al*. Circos: an information aesthetic for comparative genomics. *Genome Res* 2009;**19**(9): 1639–45.
56. Eilers PH, Goeman JJ. Enhancing scatterplots with smoothed densities. *Bioinformatics* 2004;**20**(5):623–8.
57. Brandes U. A faster algorithm for betweenness centrality. *J Math Sociol* 2001;**25**:163–77.
58. Ding Y, Yan E, Frazho A, *et al*. Pagerank for ranking authors in co-citation networks. *J Am Soc Inf Sci Technol* 2009;**60**(11): 2229–43.
59. Dong J, Horvath S. Understanding network concepts in modules. *BMC Syst Biol* 2007;**1**:24.
60. Li A, Horvath S. Network neighborhood analysis with the multi-node topological overlap measure. *Bioinformatics* 2007;**23**(2):222–31.
61. Pereira-Leal JB, Enright AJ, Ouzounis CA. Detection of functional modules from protein interaction networks. *Proteins Struct Funct Bioinform* 2004;**54**(1):49–57.
62. Almaas E. Biological impacts and context of network theory. *J Exp Biol* 2007;**210**(Pt 9):1548–58.

63. Yan KK, Fang G, Bhardwaj N, *et al*. Comparing genomes to computer operating systems in terms of the topology and evolution of their regulatory control networks. *Proc Natl Acad Sci USA* 2010;**107**(20):9186–91.

64. Newman ME. Modularity and community structure in networks. *Proc Natl Acad Sci USA* 2006;**103**(23): 8577–82.

65. Lancichinetti A, Kivela M, Saramaki J, *et al*. Characterizing the community structure of complex networks. *PloS One* 2010;**5**(8):e11976.

66. Schank T, Wagner D. Approximating clustering coefficient and transitivity. *J Graph Algor Appl* 2005;**9**.