RESEARCH

Supplemental Material: An information-theoretic, all-scales approach to comparing networks

James P Bagrow[1*] and Erik M Bollt[2]

[*]Correspondence: james.bagrow@uvm.edu
[1]Department of Mathematics & Statistics, Vermont Complex Systems Center, University of Vermont, Burlington, VT, USA
Full list of author information is available at the end of the article

S1 Portraits and Network Portrait Divergences for weighted networks

The portrait matrix B (Eq. (2)) is most naturally defined for unweighted networks since the path lengths for unweighted networks count the number of edges traversed along the path to get from one node to another. Since the number of edges is always integer-valued, these lengths can be used to define the rows of B. For weighted networks, on the other hand, path lengths are generally computed by summing edge weights along a path and will generally be continuous rather than integer-valued.

To generalize the portrait to weighted networks requires (i) using an algorithm for finding shortest paths accounting for edge weights (here we will use Dijkstra's algorithm [1]), and (ii) defining an appropriate aggregation strategy to group shortest paths by length to form the rows of B. The algorithm for finding shortest paths defines the complexity of computing the portrait: The single-source Dijkstra's algorithm with a Fibonacci heap runs in $\mathcal{O}(M+N\log N)$ time [2] for a graph of $|V|=N$ nodes and $|E|=M$ edges. This is more costly than the single-source Breadth-First Search algorithm we use for unweighted graphs, which runs in $\mathcal{O}(M+N)$ time. Computing B requires all pairs of shortest paths, therefore the total complexity for computing a weighted portrait is $\mathcal{O}(MN+N^2\log N)$. This again is more costly than the total complexity for the unweighted portrait, $\mathcal{O}(MN+N^2)$, but this is unavoidable as finding minimum-cost paths is generically more computationally intensive than finding minimum-length paths.

The simplest choice for aggregating shortest paths by length is to introduce a binning strategy for the continuous path lengths. Let $d_0 = 0 < d_1 < \cdots < d_{b+1} = L_{\max}$ define a set of b intervals or bins, where $L_{\max}$ is the length of the longest shortest path. Then the weighted portrait B can be defined such that $B_{i,k} \equiv$ the number of nodes with k nodes at distances $d_i \leq \ell < d_{i+1}$. That is, the i-th row of the weighted portrait accounts for all shortest paths with lengths falling inside the i-th bin $[d_i, d_{i+1})$. (We also take the last bin to be inclusive on both sides, $[d_b, L_{\max}]$).

To compute B using a binning requires determining the $b+1$ bin edges. Here we consider a simple, adaptive binning based on quantiles of the shortest path distribution, but a researcher is free to adopt a different binning strategy as needed. Let $\mathcal{L}(G) = \{\ell_{ij} \mid i,j \in V \wedge \ell_{ij} < \infty\}$ be the set of all unique shortest path lengths between connected pairs of nodes in graph G. We then define our binning to be the b contiguous intervals that partition $\mathcal{L}$ into subsets of (approximately) equal size. Taking $b=100$, for example, ensures that each bin contains approximately 1% of the shortest path lengths. The number of bins b can be chosen by the researcher to

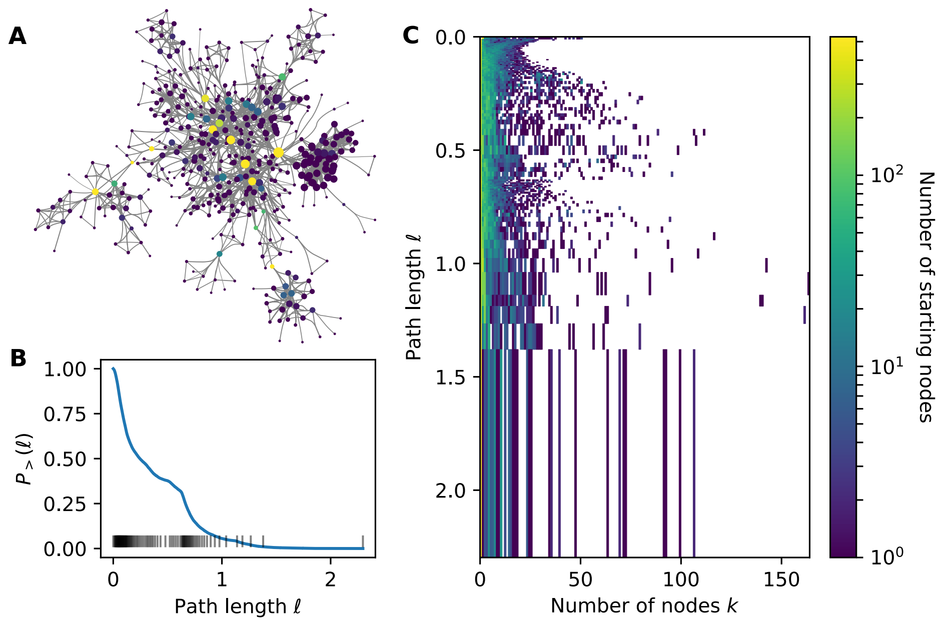

Figure S1 A weighted network and its portrait. (**A**) The network of developers contributing to IBM projects on GitHub. This is the giant connected component of the union of all graphs shown in Fig. 5A. The weight on edge (i, j) represents the number of source files edited in common by developers i and j. Node size is proportional to degree; node color is proportional to betweenness centrality. (**B**) The cumulative distribution of shortest path lengths ℓ computed using Dijkstra's algorithm with reciprocal edge weights. (**C**) The weighted network portrait. The vertical marks in panel B denote the path length binning used in C.

suit her needs, or automatically using any of a number of histogram binning rules such as Freedman-Diaconis [3] or Sturges' Rule [4].

Figure S1 shows the portrait for a weighted network, in this case taken from the IBM developer collaboration network. Edge (i, j) in this network has associated non-negative edge weight w_{ij} = the number of files edited in common by developers i and j. The network is the union of the networks shown in Fig. 5A; we draw the giant connected component of this network in Fig. S1A. For this network, we consider shortest paths found using Dijkstra's algorithm with reciprocal edge weights, i.e., the "length" of a path $(i = i_0, i_1, i_2, \ldots, i_{n+1} = j)$ is $\ell_{ij} = \sum_{t=0}^{n} w_{i_t,i_{t+1}}^{-1}$, as larger edge weights define more closely related developers. However, this choice is not necessary in general. The cumulative distribution of shortest path lengths, which we computed on all components of the network, is shown in Fig. S1B. Lastly, Fig. S1C shows the portrait B for this network. For illustration, we draw the vertical positions of the rows in this matrix using the bin edges. These bin edges are highlighted on the cumulative distribution shown in Fig. S1B.

With a new definition for B now in place for weighted networks, the Network Portrait Divergence can be computed exactly as before (Definition 3.1). However, to compare portraits for two graphs G and G', it is important for the path length binning to be the same for both. We do this here by computing b bins as quantiles of $\mathcal{L} = \mathcal{L}(G) \cup \mathcal{L}(G')$ and then compute $B(G)$ and $B(G')$ as before. This ensures the rows of B and B' are compatible in the distributions used within Definition 3.1.

Author details
[1]Department of Mathematics & Statistics, Vermont Complex Systems Center, University of Vermont, Burlington, VT, USA. [2]Department of Mathematics, Clarkson University, Potsdam, NY, USA.

References

1. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische mathematik **1**(1), 269–271 (1959)
2. Fredman, M.L., Tarjan, R.E.: Fibonacci heaps and their uses in improved network optimization algorithms. Journal of the ACM (JACM) **34**(3), 596–615 (1987)
3. Freedman, D., Diaconis, P.: On the histogram as a density estimator: L 2 theory. Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete **57**(4), 453–476 (1981)
4. Sturges, H.A.: The choice of a class interval. Journal of the American Statistical Association **21**(153), 65–66 (1926)