Approximating frustration scores in complex networks via perturbed Laplacian spectra

Andrej J. Savol and Chakra S. Chennubhotla

Phys. Rev. E **92**, 062806 — Published 4 December 2015

DOI: 10.1103/PhysRevE.92.062806

Approximating frustration scores in complex networks via perturbed Laplacian spectra

Andrej J Savol and Chakra S Chennubhotla
Dept. of Computational and Systems Biology, University of Pittsburgh School of Medicine, Pittsburgh, Pennsylvania 15260, United States

Systems of many interacting components, as found in physics, biology, infrastructure, and the social sciences, are often modeled by simple networks of nodes and edges. The real-world systems frequently confront outside intervention or internal damage whose impact must be predicted or minimized, and such perturbations are then mimicked in the models by altering nodes or edges. This leads to the broad issue of how to best quantify changes in a model network after some type of perturbation. In the case of node removal there are many centrality metrics which associate a scalar quantity with the removed node, but it can be difficult to associate the quantities with some intuitive aspect of physical behavior in the network. This presents a serious hurdle to the application of network theory: real-world utility networks are rarely altered according to theoretic principles unless the kinetic impact on the network's users are fully appreciated beforehand. In pursuit of a kinetically-interpretable centrality score, we discuss the f-score, or frustration score. Each f-score quantifies whether a selected node accelerates or inhibits global mean first passage times to a second, independently-selected target node. We show that this is a natural way of revealing the dynamical importance of a node in some networks. After discussing merits of the f-score metric, we combine spectral and Laplacian matrix theory in order to quickly approximate the exact f-score values, which can otherwise be expensive to compute. Following tests on both synthetic and real medium-sized networks, we report f-score runtime improvements over exact brute force approaches in the range of 0 to 400% with low error ($< 3\%$).

I. INTRODUCTION

Systems in the physical, social, and biological sciences are composed of many interacting units which collectively give rise to complicated, global dynamics [1–5]. Yet, these emergent behaviors can also be modeled by random walks over simple network models [6]. In such models direct probability flow is permitted between nodes connected by an edge, the absence of an edge between nodes means probability can travel between them only indirectly, and nodes (V) and edges (E) collectively constitute the network $\mathcal{H}(V, E)$ as a closed-system and induce its behavior. Network models have flexibly modeled disease propagation [7], neuronal dynamics [8], router communication [9], protein folding pathways [10], utility grids [11], collaboration histories [12], and other phenomena at wide-ranging spatial and temporal scales [13, 14]. Importantly, real-world systems like these frequently confront outside intervention or internal damage whose impact must be predicted or minimized [15, 16]. Quantifying this vulnerability in the face of targeted or random attacks motivates a more general network science question that is the principal issue of this study: Which network nodes are important or central to the entire graph [17–21]? This question is open because a quantitative definition of *important* and *central* is still required [22].

To illustrate this issue, consider transition network models of protein folding, where different protein geometries are modeled by distinct nodes and observed conformational transitions are modeled by distinct edges. In such a network, a node might be *important* if it represents the folded protein conformation which is known to perform a biochemical function. Such a node is likewise *central* in the sense of providing a connectivity hub for many other possible geometries [23]. But, knowing in advance about the folded conformation node, we might then be interested in other nodes that funnel or alternately block the transition to the central node [24, 25]; these nodes are called *bottlenecks* and *traps*, respectively. An interest in these secondary nodes is natural whenever a network contains a node of more *a priori* relevance than others [26] (such a node, e.g. the folded state, is a *target node*, n_t). For these networks, our principal question has changed to: Which nodes are important *given* our pre-selected target node n_t? I.e., what happens at n_t when perturbations are made elsewhere? It is this set of perturbed nodes, denoted $n_p \in N_p$, for which we desire some individual quantification of importance in light of our inherent focus on dynamic behavior at n_t. An epidemiological analogue is to ask how the infection risk faced by a particular individual n_t changes in response to vaccination of a second individual n_p [13, 27]. A metric that encapsulates this relationship must necessarily consider three entities: target node n_t, perturbed node n_p (whose quantification of importance is desired), and an overall network topology or structure $\mathcal{H} = \mathcal{H}(V, E)$ in which both these nodes live (Fig. 1A).

Node importance more generally can be quantified by many spectral techniques and graph theoretic principles. Such centrality scores may be based on the intact network topology or, additionally, on the changes observed in network characteristics after a node or edge is altered [28–32]. Useful interpretability of these quantities in either approach depends on the formulation of the centrality measure chosen and the physical or social system modeled by the network. For example, the *subgraph centrality* and *communicability* measures provide predictions of

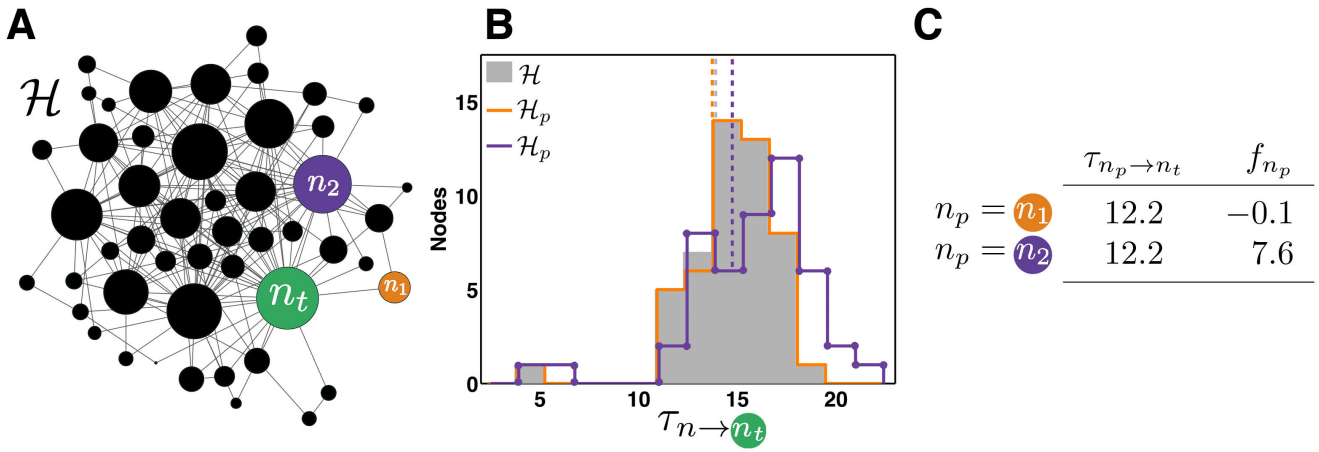

Figure 1: (Color online) F-scores quantify the strength of bottlenecks in an example complex network. (A) Example network $\mathcal{H}$ with 49 nodes; node widths indicate total degree $\mathbf{s}_n$ including self-loops. Target node n_t is shown in green. F-scores, f_{n_p}, are computed separately for two nodes, n_1 (orange) and n_2 (purple), by removing them from $\mathcal{H}$ and observing changes in MFPTs to n_t (green). (B) A histogram of mean first passage times (MFPTs), $\tau_{n\to n_t}$, where the mean first passage time is time required for a random walker from each node in network $\mathcal{H}$ to arrive at target node n_t. Solid gray histogram, intact graph $\mathcal{H}$; unmarked orange line, $\mathcal{H}_p = \mathcal{H} \setminus n_1$; dotted purple line, $\mathcal{H}_p = \mathcal{H} \setminus n_2$. Dashed vertical lines indicate the average MFPTs over all nodes, the *trapping time*. F-scores, f_{n_p}, are computed from the relative change in trapping time (Eq. 5). (C) A comparison of MFPTs and f-scores. In the intact graph $\mathcal{H}$, n_1 and n_2 have identical mean first passage times to n_t, but they impact graph dynamics differently when removed. Node n_1 minimally impacts transit times to n_t when it is removed from the graph ($f_{n_1} = -0.1$). In contrast, n_2 is a more important bottleneck between the graph and n_t, so removing it has a greater impact on MFPTs ($f_{n_2} = 7.6$), seen in the shift of the purple histogram (dotted line) to longer (slower) transit times (B).

protein lethality and diffusion for networks of protein interactions or harmonic oscillations, respectively [33, 34]. Some other interpretable metrics, such as synchronization [35], diffusion [36], and relaxation rates [28], measure global quantities and have no inherent n_t dependence. In our analogy this means these metrics only tell us about averages across all potential patients and not the particular individual, n_t, whose infection risk changes when someone else, n_p, is vaccinated. An additional consideration is that many such metrics are strongly correlated and provide duplicate information [37]. In light of these issues we therefore ask: what interpretable metric can quantify the importance of each perturbed node n_p vis-a-vis the target node n_t?

Our choice is called an f-score [25, 38], f_{n_p}, and is based on the concept of *trapping time*, the average time required by a Markov chain or random walk to arrive at the target node n_t from any other node (start node) in the network [26, 39]. Trapping time is the weighted average of mean first passage times (MFPTs, equivalent to *hitting times* [40] or *transit times*) to n_t over every node. An individual MFPT value itself, $\tau_{n\to m}$, gives the average time required for a random walk starting at node n to arrive at m [41]. As opposed to the shortest path distance, a MFPT value $\tau_{n\to m}(\mathcal{H})$ reflects the influence of all possible paths between nodes n and m in graph $\mathcal{H}$. Whereas MFPTs are necessarily a function of two specified endpoints (n and m), in this work concern is restricted to those transition paths that terminate at the user-selected target node n_t, and trapping time is then the average over all start nodes: $\bar{\tau}_{n_t} = \frac{1}{N-1}\sum_{n\neq n_t}^{N} \tau_{n\to n_t}$, where there are N nodes in the intact network $\mathcal{H}$ (Fig. 1A). We then ask how much the trapping time $\bar{\tau}_{n_t}$ changes in response to individual excision of non-target nodes n_p from the network (Fig. 1A). In agreement with intuition, bottleneck nodes when removed will increase the trapping time (random walkers must find detours to n_t) and kinetic traps when removed will decrease the trapping time (random walkers don't get 'stuck' far away from n_t) (Fig. 1B, dashed lines). The resulting quantity for excised node n_p, denoted $f(n_p, n_t, \mathcal{H})$, therefore tells us the mean relative change, or *frustration*, in all paths to n_t as a result of node n_p (Fig. 1C). Whereas *frustration* has been defined in various synchronization contexts [42, 43], here the word captures the propensity of a single node to accelerate or inhibit transition paths to n_t due to its topological context (location in the network). Formally,

$$f(n_p, n_t, \mathcal{H}) = f_{n_p} = 100*\left(\frac{1}{N-2}\sum_{n\neq n_t, n_p}^{N} \tau_{n\to n_t}(\mathcal{H}_p) - \frac{1}{N-1}\sum_{n\neq n_t}^{N} \tau_{n\to n_t}(\mathcal{H})\right) \Big/ \left(\frac{1}{N-1}\sum_{n\neq n_t}^{N} \tau_{n\to n_t}(\mathcal{H})\right), \quad (1)$$

where $\mathcal{H}_p$ is identical to $\mathcal{H}$ except node n_p has been excised, i.e. $\mathcal{H}_p = \mathcal{H} \setminus n_p$; the total number of computed MFPTs in $\mathcal{H}_p$ is $N-2$ since $\tau_{n_t \to n_t}$ is ignored. Eq. 1 includes a scaling coefficient to emphasize that f-scores convey percentages, and unless explicit dependencies are required, we often abbreviate $f(n_p, n_t, \mathcal{H})$ as f_{n_p} or f. In summary, an f-score tells us precisely how much *all* paths to n_t are inhibited ($f_{n_p} < 0$) or accelerated ($f_{n_p} > 0$) as a result of node n_p *in the intact graph* $\mathcal{H}$ (Fig. 1).

The intuition behind f_{n_p} values and their comparison to MFPT values can be further clarified via a node removal task: pruning a network such that trapping times at n_t are minimized (i.e. arrival rates at n_t are maximized). This is illustrated in Fig. 2 using two model networks (network $\mathcal{H}$ as introduced in Fig. 1A and a second synthetic network, $\mathcal{H}_{500}$, described in Table I). F-scores are able to make better predictions in this regard than MFPT values. This is because MFPT values do not reflect the topological context of the removed node [44, 45], and so the pruning procedure cannot determine if a given node removal will have a large impact on transit times to n_t across the remaining network. F-scores, in contrast, inherently encode the kinetic impact of each pruning candidate n_p; node degree and local connectivity are inherently reflected in each f_{n_p}'s sign and magnitude. Kinetic interpretability of this sort is key to a successful node metric [20].

In the following we first connect spectral theory with MFPTs and trapping times and then propose a protocol for approximating f-scores using matrix perturbation theory that is more efficient than direct matrix inversion methods we know of (algorithm details in appendix). Examples and tests are conducted with synthetic and real datasets, in all cases using sparse, nonregular, and undirected graphs.

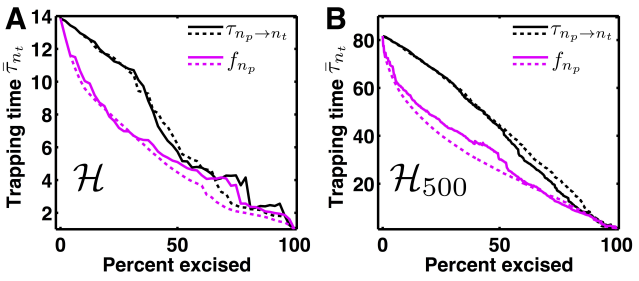

Figure 2: (Color online) MFPTs and f-scores as graph pruning criteria. Example networks $\mathcal{H}$ from Fig. 1A (A) and $\mathcal{H}_{500}$ from Table I (B) are sequentially pruned according to MFPT ($\tau_{n_p \to n_t}$, black, upper curves), or f-score (f_{n_p}, magenta, lower curves), where the trapping time (at n_t) of the resulting network is shown at each iteration. Nodes are removed in the order resulting from initial values in the full network (solid) or values recalculated at each iteration (dashed).

II. METHODS

For some chosen target node n_t in graph $\mathcal{H}$, denominator and subtrahend in Eq. 1 need be computed only once for any desired set of perturbed nodes $n_p \in N_p$. Because the topology in $\mathcal{H}$ is mostly preserved for any single node perturbation, we can therefore exploit spectral properties of $\mathcal{H}$ in order to quickly approximate the first numerator term given that we already know the second, which has no n_p dependence. We begin in this direction by introducing nomenclature relevant to mean first passage times and perturbation theory in the context of complex networks.

Let $\mathcal{H} = \mathcal{H}(V, E)$ be a weighted, undirected graph where V is the set of vertices and E is the set of edge weights. The vertices or nodes are indexed by $n, m \in \{1 \dots N\}$. Key nodes receive special symbols: n_t for the user-selected target node; $n_p \in N_p$ for the user-selected perturbed node ($N_p = \{n_1, n_2\}$ in Fig. 1A); $n_g \in G_n$ for all *neighbors* of some node n (n and n_g are directly connected by an edge); and $n_{\bar{g}} \in \bar{G}_n$ for all *foreigners* of n (n and $n_{\bar{g}}$ are not directly connected by an edge). The graph Laplacian $\mathbf{L}$, an N x N matrix, is defined as $\mathbf{L} = \mathbf{S} - \mathbf{A}$, where $\mathbf{A}$, the symmetric adjacency matrix is defined such that $\mathbf{A}_{nm} = \mathbf{A}_{mn} = a_{nm} \in E$ is the nonnegative weight of the edge connecting nodes n and m, and $\mathbf{A}_{mm}$ is the weight of self-loops for node m. Because $\mathbf{L}$ contains no information of node self-loops, which are essential for modeling many complex phenomena, our expressions often require matrix $\mathbf{S}$, whose diagonal carries node degrees, i.e., $\mathbf{S}_{mm} = \mathbf{s}_m = \sum_{n=1}^{N} \mathbf{A}_{mn}$. A column vector of these degrees is denoted as $\mathbf{s}$, and $s = \mathbf{s}^T \mathbf{1}$ is the total edge weight in the network, sometimes denoted $vol(\mathcal{H})$ [49, 50]. Perturbation of a single

Table I: Dataset summary. Six networks are compared based on node count N, edge count nnz, degree distribution exponent α, algebraic connectivity λ_2, and spectral radius λ_N. In $\mathcal{H}_{\mathsf{A}}$ edge weights denote average total daily seat capacity between busiest US commercial airports. In $\mathcal{H}_{\mathsf{YST}}$ edge weights denote confidence in functional interactions based on aggregated screening studies. In social network $\mathcal{H}_{\mathsf{UC}}$ edges denote the symmetrized number of communicated institutional electronic messages. Standard deviation of estimated degree exponent α was < 0.07 for all networks [46].

Name	Description	N	nnz	α	λ_2	λ_N
	Synthetic networks:					
$\mathcal{H}_{500}$		500	1896	2.46	5.02	1.41e+4
$\mathcal{H}_{1000}$		1000	4199	2.26	17.31	2.37e+4
$\mathcal{H}_{2000}$		2002	9725	2.13	34.46	8.20e+4
	Real networks:					
$\mathcal{H}_{\mathsf{A}}$	US airports [2]	500	5960	1.64	0.2	1.4e+05
$\mathcal{H}_{\mathsf{YST}}$	Yeast [47]	1890	9464	1.80	0.39	1.20e+03
$\mathcal{H}_{\mathsf{UC}}$	UC Irvine [48]	1893	27670	1.56	0.17	809.1

node amounts to decreasing all the node's edges, including self-transitions by some relative amount $\epsilon \in [0\ 1]$, i.e., $\mathbf{L}_{p_{n_p,n_p}} = (1-\epsilon) \times \mathbf{L}_{n_p n_p}$ with corresponding values decreased at nodes G_{n_p} so that $\sum_{m=1}^{N} \mathbf{L}_{p_{nm}} = 0\ \forall n$. Node removal occurs when $\epsilon = 1$. The matrix that encodes the ϵ-weighted decrease in self-transitions and edge weights is $\mathbf{B}$ such that $\mathbf{L}_p = \mathbf{L} + \epsilon\mathbf{B}$. A perturbation impacts the adjacency matrix analogously, $\mathbf{A}_p = \mathbf{A} - (\epsilon\mathbf{A}_{[n_p,:]} + \epsilon\mathbf{A}_{[:,n_p]})$, where the colon denotes indices $1 \dots N$. Subscript brackets denote index ranges.

A. Mean first passage times, trapping times, and f-scores

With these and a few additional definitions we can compute the pairwise MFPT matrix for all nodes in a weighted, symmetric network $\mathcal{H}$. First, the *fundamental matrix* $\mathbf{Z}$ from Markov chain literature is defined as

$$\mathbf{Z} = \left(\mathbf{I} - (\mathbf{P} - \mathbf{P}^*)\right)^{-1}, \tag{2}$$

where $\mathbf{P} = \mathbf{S}^{-1}\mathbf{A}$ is the row-stochastic transition probability matrix, $\mathbf{I}$ is the identity matrix, and $\mathbf{P}^*$ is a matrix whose columns are the stationary distribution $\vec{\alpha}$ (i.e. $\vec{\alpha}$ is the dominant eigenvector of $\mathbf{P}$). The traditional expression for computing all pairwise MFPT values then is

$$\mathbf{M}\left(\mathcal{H}\right) = \{\tau_{n\to m}\left(\mathcal{H}\right)\} = \left(\mathbf{I} - \mathbf{Z} + \mathbf{E}\mathbf{Z}_{diag}\right)\mathbf{D}, \tag{3}$$

where $\mathbf{Z}_{diag}$ is equivalent to $\mathbf{Z}$ but with vanished off-diagonals, $\mathbf{E}$ is a constant matrix of all 1's, and $\mathbf{D}$ is also diagonal and carries in its diagonal the inverse of the stationary distribution (or limiting probability): $\mathbf{D}_{nn} = \frac{1}{\alpha_n}$ [41]. Trapping times $\bar{\tau}_{n_t}$ for some target node n_t are then computed by averaging over the appropriate column of $\mathbf{M}$:

$$\bar{\tau}_{n_t} = \frac{1}{N-1} \sum_{m=1 \neq n_t}^{N} \mathbf{M}_{m,n_t}, \tag{4}$$

such that our exact f-score definition (1) becomes

$$f(n_p, n_t, \mathcal{H}) = 100 * \frac{\bar{\tau}_{n_t}\left(\mathcal{H}_p\right) - \bar{\tau}_{n_t}\left(\mathcal{H}\right)}{\bar{\tau}_{n_t}\left(\mathcal{H}\right)}. \tag{5}$$

Even though $\mathbf{A}$ is generally sparse and $\mathbf{S}$, being diagonal, is cheaply invertible, the matrix which is inverted in (2) to produce $\mathbf{Z}$ is dense. As a result, each *exact* f_{n_p} value desired requires an expensive matrix inversion, and no dynamic or topological information about $\mathcal{H}$ is recycled when iterating over user-selected $\{n_p\}$. We note, however, that the fundamental matrix for the perturbed network $\mathbf{Z}_p$ can be estimated from the intact graph's $\mathbf{Z}$ matrix using the Sherman-Morrison-Woodbury formula:

$$\mathbf{Z}_p \approx \mathbf{Z} + \mathbf{Z}\mathbf{U}\left(\mathbf{I} - \mathbf{V}\mathbf{Z}\mathbf{U}\right)^{-1}\mathbf{V}\mathbf{Z},$$

where $\mathbf{UV}$ is some low-rank approximation of $\mathbf{P}^* - \mathbf{P} + \mathbf{P}_p - \mathbf{P}_p^*$ [51]. This is worth exploring as an alternative to our Laplacian-based approach, though the rank of the perturbation will generally be equal to or larger than the number of edges at the perturbed node, potentially quite large.

One additional alternative formulation for $\bar{\tau}_{n_t}$ that flexibly allows n_t to be comprised of an arbitrary set of target nodes is presented in Ref. 24, but efficiency is an issue because matrix exponents must be evaluated multiple times for each n_p of interest. Thankfully, trapping times $\bar{\tau}_{n_t}$ can be computed without explicitly calculating individual transit times $\tau_{n\to n_t}$ and averaging over n as in (4). Specifically, a spectral formulation presented in Ref. 52 permits $\bar{\tau}_{n_t}$ to be expressed via Laplacian eigenvectors $\mathbf{u}_{1\dots N}$ and eigenvalues $\lambda_{1\dots N}$:

$$\bar{\tau}_{n_t} = \frac{N}{N-1} \sum_{k=2}^{N} \frac{1}{\lambda_k} \left(s u_{n_t k}^2 - u_{n_t k} \mathbf{s}^T \mathbf{u}_k\right), \tag{6}$$

where the first eigenpair is excluded because $\lambda_1 = 0$. A related treatment with adjacency matrix spectra is also possible [39]. Eq. 6 invokes all non-dominant eigenpairs, where an eigenpair is defined as the associated quantities $\{\mathbf{u}_k, \lambda_k\}$ such that $\mathbf{L}\mathbf{u}_k = \lambda_k \mathbf{u}_k$. Eigenpairs are indexed by eigenindices $j, k \in \{1 \dots N\}$ and sorted: $\lambda_1 = 0 \leq \lambda_1 \leq \lambda_2 \dots \leq \lambda_N$. The dominant eigenvector $\mathbf{u}_1 = \mathbf{1}/N$. Eigenvectors together form the columns of a matrix $\mathbf{U} \in \mathbb{R}^{N\times N}$, where $\mathbf{U}_k$ or $\mathbf{u}_k$ indicates the kth column and $\mathbf{U}_{ij}$ or u_{ij} indicates the ith element of the jth column of $\mathbf{U}$.

Across many disciplines, these Laplacian eigenvectors ($\mathbf{U}$) are used to map the topology encoded in $\mathbf{L}$ to an alternate or lower-dimensionality basis, often to facilitate coarse-graining [53, 54] or clustering [50, 55], and many dynamic measures have naturally been formulated from them [56]. For example, one may ask which link or node removals maximally or minimally impact the *algebraic connectivity* λ_2 or the *eigenratio* λ_2/λ_N [57], both being summary measures of dynamic synchronization [5, 58, 59]. One may also examine an individual row of the eigenvector matrix, i.e. $\mathbf{U}_{[n_p,1:N]}$, whose elements convey the dynamical importance of node n_p within each eigenfrequency [22]. Critically, most such interpretations of $\mathbf{U}$ and λ relate to global behavior over the entire graph.

Part of the appeal of synchronization- and eigenratio-based centrality measures is that only dominant and/or extreme eigenpairs are required, meaning these centrality values even for very large graphs are feasible with sparse eigensolvers. Formally, Eq. 6 requires the entire spectrum and cannot take advantage of these numerical methods. However, Eq. 6 favorably permits us to consider each eigenpair separately, and so we associate a symbol $\bar{\tau}_{n_t}^k$ with the trapping time contribution of each distinct eigenpair k: $\bar{\tau}_{n_t}^k = \frac{N}{N-1}\left(s u_{n_t k}^2 - u_{n_t k}\mathbf{s}^T\mathbf{u}_k\right)$ such that total trapping time is their sum: $\bar{\tau}_{n_t} = \sum_{k=2}^{N} \bar{\tau}_{n_t}^k$. The

central concept is that the spectra of $\mathbf{L}$ and $\mathbf{L}_p$ are closely related and therefore many $\bar{\tau}^k_{n_t}$ values will be unchanged upon network perturbation. That is, given trapping time contributions $\bar{\tau}^k_{n_p}$ $\forall k \neq 1$ for the intact graph $\mathcal{H}$, we can selectively estimate only those eigenpairs in $\mathcal{H}_p$ (and thus only those $\bar{\tau}^k_{n_p}$ values) that non-negligibly impact a node's associated f-score (the other variables in Eq. 6, s and $\mathbf{s}$, are known observables of $\mathcal{H}_p$). In summary, instead of an exact f_{n_p} we compute an estimate $\tilde{f}_{n_p}$ by (1) identifying free eigenindices k_F that substantially alter total trapping time $\sum_{k=2}^{N} \bar{\tau}^k_{n_t}$, and then (2) efficiently estimating quantities $\mathbf{u}_k$ and $\tilde{\lambda}_k$ necessary for Eq. 6.

B. Estimating λ_p

In the case of networks with very controlled or regular structure, convenient analytic expressions for the perturbed eigenvalues λ_p are known; brute force eigendecomposition is not required [26, 52]. With complex networks, however, alternatives other than dense eigensolvers include perturbation theory or eigenvalue bounds from interlacing formulas. In the latter, one can bound the maximum shift of the eigenvalues $|\lambda - \lambda_p|$ given the local topology of the perturbed node n_p [60–62], but in our experience these bounds are not adequately tight and, besides, eigenvalue perturbation is more accurate and almost as fast. Regardless, it is the estimation of the eigenvectors $\tilde{\mathbf{U}}$ that represents the largest computational expense.

For notational clarity, tildes are assigned to approximate/estimated quantities of the perturbed spectrum, subscript or superscript p's indicate exact quantities or indices, and, when necessary, subscript 0's indicate unperturbed variables. A matrix of estimated Laplacian eigenvectors is therefore denoted $\tilde{\mathbf{U}}$, while dense eigendecomposition would yield $\mathbf{U}_p$ given $\mathbf{L}_p$.

Using classical first order perturbation theory, for some eigenpair k:

$$\tilde{\lambda}_k - \lambda_k = \frac{\mathbf{u}_k^T \epsilon \mathbf{B} \mathbf{u}_k}{\mathbf{u}_k^T \mathbf{u}_k}, \tag{7}$$

where $\mathbf{L}_p = \mathbf{L} + \epsilon \mathbf{B}$ is the Laplacian of $\mathcal{H}_p$ [63]. However, in the case that the perturbation impacts a single node n_p, meaning all connected edges (and self-loops) are proportionally decreased by ϵ, the expression can be simplified (subscript k implied after first line):

$$\begin{aligned}
\frac{\Delta\lambda_k}{\epsilon} &= \frac{\tilde{\lambda}_k - \lambda_k}{\epsilon} = \mathbf{u}_k^T \mathbf{B} \mathbf{u}_k \\
&= \sum_{n \in G_{n_p}} u_n \left(\mathbf{u}^T \mathbf{B}_n\right) + u_{n_p} \mathbf{u}^T \mathbf{B}_{n_p} \\
&= \sum_{n \in G_{n_p}} \mathbf{B}_{nn} u_n^2 + u_{n_p} \left(\mathbf{u}^T \mathbf{B}_{n_p} - u_{n_p} \mathbf{B}_{n_p n_p}\right) \\
&\qquad + u_{n_p} \left(\mathbf{u}^T \mathbf{B}_{n_p}\right) \\
&= \left(-\mathbf{u}^T \mathsf{diag}(\mathbf{B}_{n_p}) \mathbf{u} + u_{n_p}^2 \mathbf{B}_{n_p n_p}\right) \\
&\qquad + u_{n_p n_p}^2 \left(-\lambda - \mathbf{B}_{n_p n_p}\right) + u_{n_p n_p}^2 \left(-\lambda\right) \\
&= \mathbf{u}^T \mathsf{diag}(\mathbf{L}_{n_p}) \mathbf{u} + u_{n_p}^2 \left(-\mathbf{L}_{n_p n_p} - \lambda + \mathbf{L}_{n_p n_p} - \lambda\right) \\
&= \left(\mathbf{u}.^2\right)^T \mathbf{L}_{n_p} - 2\lambda u_{n_p}^2 \\
\Rightarrow \tilde{\lambda}_k - \lambda_k &= \epsilon * \left(\left(\mathbf{u}_k.^2\right)^T \mathbf{L}_{n_p} - 2\lambda_k u_{n_p k}^2\right)
\end{aligned} \tag{8}$$

where the notation $\left(.^2\right)$ signifies the element-wise exponent, $\mathsf{diag}(\mathbf{x})$ is a zero matrix with $\mathbf{x}$ along its diagonal, $\mathbf{B}_{n_p}$ is the n_pth column vector of $\mathbf{B}$, $\mathbf{L}_{n_p}$ denotes the n_pth column of the *intact* Laplacian, and a matrix with two subscripts denotes a single element, as in $\mathbf{B}_{n_p n_p}$.

C. Estimating $\mathbf{U}_p$

Likewise, we can also update the eigenvectors using standard perturbation approaches [64, 65]:

$$\tilde{\mathbf{u}}_k = \mathbf{u}_k + \sum_{j=1 \neq k}^{N} \frac{\mathbf{u}_j^T \left(\mathbf{L}_p - \mathbf{I}\tilde{\lambda}_k\right) \mathbf{u}_k}{\tilde{\lambda}_k - \tilde{\lambda}_j} \mathbf{u}_j. \tag{9}$$

This update step has complexity $\mathcal{O}(n^2)$, and updating N eigenvectors of the spectrum costs $\mathcal{O}(n^3)$. Naively implemented, this would constitute a profligate linear estimate to the eigenbasis when exact, direct eigensolvers have the same approximate cost, sparse solvers being cheaper still. In practice, however, the perturbations here require only the subset k_F of the spectrum to be updated for accurate estimates, and the corrections themselves are small and vanish rapidly. As we will show, the set of selected eigenpairs are often non-extreme and non-adjacent, and most efficient eigensolvers are not traditionally amenable to updating simultaneously non-contiguous eigenpairs [66]. It is for this reason that we choose to iteratively update $\tilde{\mathbf{U}}$ using the method least efficient in traditional implementation but well-suited to the specific perturbation structure $\mathbf{B}$ and stopping criterion $|\Delta\tilde{f}_{n_p}| < f^*$.

D. A heuristic for k_F

As mentioned, we accelerate Eq. 9 by limiting the summation to selected eigenindices k_F. We identify this set of indices by observing that when a local perturbation is made in a network, some Laplacian eigenpairs

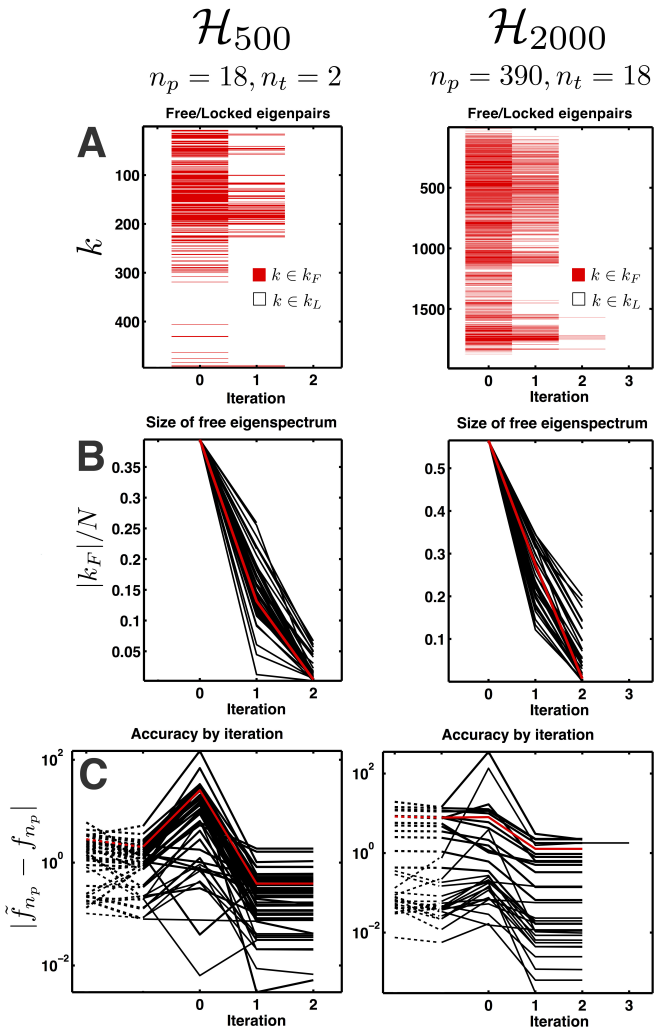

Figure 3: (Color online) The number of free eigenindices $|k_F|$ decreases each iteration. (A) Free eigenindices per iteration are shown for representative perturbed n_p and target n_t nodes in $\mathcal{H}_{500}$ (left) and $\mathcal{H}_{2000}$ (right). (B) Convergence of k_F shown for large set of test target nodes N_p. Convergence for target node n_p from row (A) shown in red (print version, gray). Vertical axis gives proportion of total spectrum. (C) Absolute accuracy of $\tilde{f}$ at each iteration. Dashed lines show accuracy change with only the eigenvalue update $\tilde{\lambda}$ (Eq. 8), which is performed only once and only before the first eigenvector update which occurs at Iteration 0 (see Appendix pseudocode line 11). Red (gray) curves as in (B). Algorithm terminates when $\tilde{f}$ changes by less than f^*.

are impacted more than others. Efficient computation of the perturbed spectrum should ignore unimpacted eigenpairs, and we can discriminate between eigenpairs further by considering only those whose contributions to trapping time at n_t change substantially upon the perturbation, that is $|\Delta\tau^k_{n_t}| > \tau^*_0$. In order to effectively classify eigenpairs into a *free* class, k_F and a *locked* class, k_L, we need a heuristic for $|\tilde{\Delta}\tau^k_{n_t}|$ that avoids direct eigendecomposition. Our choice is

$$|\tilde{\Delta}\bar{\tau}^k_{n_t}| = \tilde{\bar{\tau}}^k_{n_t}(\mathcal{H}_p) - \bar{\tau}^k_{n_t}(\mathcal{H}) \tag{10}$$

where

$$\tilde{\bar{\tau}}^k_{n_t} = \frac{1}{\tilde{\lambda}_k}\left(\frac{N}{N-1}\right)\left(s_p\tilde{u}^2_{n_t k} - \mathbf{s}_p^T\tilde{\mathbf{u}}_k\tilde{u}_{n_t,k}\right). \tag{11}$$

Vector $\tilde{\mathbf{u}}_k$ is a column of $\tilde{\mathbf{U}}$, itself equal to $\mathbf{U}$ with the exception of rows corresponding to the perturbed node n_p and its neighbors G_{n_p}. Specifically,

$$\tilde{\mathbf{U}}_{[n_{pg},:]} = \mathbf{U}_{[n_{pg},:]} - 2\left(\mathbf{L}_{p[n_{pg},:]} * \mathbf{U} - \mathbf{U}_{[n_{pg},:]} * \mathbf{I}\vec{\tilde{\lambda}}\right) \tag{12}$$

where $n_{pg} = \{n_p \cup G_{n_p}\}$, $\vec{\tilde{\lambda}}$ is a vector of currently estimated eigenvalues, and the colon denotes indices $1 \ldots N$. Changes in the elements of the approximation vectors $\tilde{\mathbf{U}}$ correspond to the gradient of the Rayleigh quotient [67] evaluated only at n_p and G_{n_p} since the gradient at all other nodes will be negligible. Tildes over returned values emphasize that (11) and (12) are not exact but still provide a convenient heuristic for selecting the initial free eigenindices:

$$k_F = \underset{k}{\mathsf{find}}\left(|\tilde{\Delta}\bar{\tau}^k_{n_t}| > \bar{\tau}^*_{\mathsf{iter}}\right). \tag{13}$$

Intuitively, Eq. 12 tells us about the impact of the perturbation given (i) the network $\mathcal{H}$ and (ii) the perturbed node n_p, whereas Eq. 11 tells us about the impact of the perturbation given all three involved entities: graph $\mathcal{H}$, node n_p, *and* target node n_t. Together, the expressions reveal which k eigenindices give rise to large predicted $|\Delta\bar{\tau}^k|$ values. We only employ this routine at $\mathsf{iter} = 0$, before vectors $\mathbf{U}_{k_F}$ have been updated with linear estimate Eq. 9. Subsequently, provided with $\tilde{\mathbf{U}}_{\mathsf{iter}>0}$, we can utilize the observed changes in trapping time contributions $|\bar{\tau}^k_{n_t}|$ to select k_F for the next iteration (Fig. 3).

E. Algorithm thresholds

There are two user-selected parameters that control the trade-off between speed and accuracy within the procedure. The first, $\bar{\tau}^*_{\mathsf{iter}}$, controls whether a given eigenvector $\mathbf{U}_{k\in k_F}$ remains *free* and in k_F after an iterative update or gets *locked* and moved into the set k_L. Presently, $\bar{\tau}^*_{\mathsf{iter}}$ is set so that k_F after each iteration includes those eigenvectors that contribute 99.5% percent of the total change in $\bar{\tau}$. Iteration histories of $|k_F|$ with this threshold are shown for two synthesized networks in Fig. 3.

The second user parameter, f^*, determines when the algorithm terminates. Once $\tilde{f}_{n_p}$ proportionally changes less than f^* per iteration, the algorithm terminates. A threshold of $f^* = 0.01$ in our experience produces good accuracy correlations.

F. Methods Summary

Our protocol works by perturbing node n_p by a small amount $\epsilon \sim 10e{-}4$ and iteratively correcting eigenvectors $\mathbf{U}$ from the intact graph $\mathcal{H}$ to approximate the basis of the altered graph, $\mathcal{H}_p$. However, we choose to update only vectors that make significant ($> \bar{\tau}^*$) contribution to the trapping time, $\bar{\tau}_{n_t}$, given the user-chosen target node n_t. That is, we choose to permit small non-orthogonalities in the updated spectrum as long as the estimated frustration score $\tilde{f}_{n_p}$ stabilizes. Specifically, at each iteration the set of vectors that gets updated is denoted $k_F \subset \{2 \dots N\}$, and this set is non-increasing with each iteration. Those eigenvectors that are already converged are called *locked* and denoted k_L such that $k_L \cap k_F = \emptyset$. (Moreover, when $\mathsf{iter} = 0$, most eigenvector elements do not change, so we can restrict the update to elements corresponding to n_F, that is, *free* elements row-wise of the current eigenvectors $\mathbf{U}$. In subsequent iterations, when $\mathsf{iter} > 0$, $n_F = \{1 \dots N\}$. See appendix pseudocode lines 14 and 23). Boxed pseudocode is given in the appendix: **Fast f-score estimation**. All computations were performed with Matlab [68]. Network visualizations were produced with Gephi [69].

III. NUMERICAL RESULTS

We tested our algorithm on six small to medium networks, both synthesized and naturally occurring (Table I). Symmetric synthesized networks $\mathcal{H}_{500}$, $\mathcal{H}_{1000}$, and $\mathcal{H}_{2000}$ were first generated with Complex Networks [70] and then self and non-self weights were assigned randomly but symmetrically to existing edges. Visualizations for $\mathcal{H}_{1000}$ and $\mathcal{H}_{\mathsf{A}}$ are provided in Fig. 4. To illustrate the relationship between (i) the free eigenspectrum k_F and (ii) f-score predictions as the algorithm progresses for the synthetic networks, we randomly chose a n_t in each synthetic network and charted algorithm execution for multiple representative nodes $\{n_p\}$ (Fig. 3). Specifically, convergence properties for one example node n_p are shown in red while other selected n_p are shown with black curves (Fig. 3B and C).

Convergence for a single representative n_p is illustrated in Fig. 5. Qualitatively, convergence behavior was consistent among all tested networks. We observed that the size of the free eigenspectrum $|k_F|$ decreases quasi-linearly each iteration (Fig. 3B) given a selection threshold $\tau^* = 0.995$, and that $\tilde{f}$ convergence is attained within three iterations for $\mathcal{H}_{500}$ and four iterations for $\mathcal{H}_{2000}$ (Fig. 3C). The free eigenpairs were distributed throughout the spectra, consistent with our claim that changes in trapping time cannot be fully recovered by extreme eigenpairs alone (Fig. 5C). Some pairs remain free through several iterations, but only free eigenpairs can remain free and once locked an eigenpair will not be updated further.

Even though $|k_F|$ apparently decreases, it is not the case that estimated f-scores likewise converge monotonically toward the true f_{n_p}, and in fact they often get worse during the first iteration, $\mathsf{iter} = 0$ (Figs. 3C and 5B). That is, a single iteration of eigenvector update (Eq. 9) often produces worse f predictions than scores estimated with only approximated eigenvalues (Fig. 3C, dashed lines). This illustrates that transit/trapping times are many-to-one indirect functions of the spectrum; the objective formally being minimized in Eq. 9 (and pseudocode line 16) is not $\tilde{f}$ but the gradient of the Rayleigh quotient (at nodes n_F). Consequently, as free eigenpairs adjust to the graph structure in $\mathcal{H}_p$ our estimates $\tilde{f}$ can temporarily suffer. However, as k_F diminishes and trapping time contributions ($\bar{\tau}^k$) stabilize the predicted f-score $\tilde{f}$ generally approaches the true value (Fig. 3C). A final prediction error $|f - \tilde{f}_{\mathsf{iter}>0}|$ worse than starting prediction error $|f - \tilde{f}_{\mathsf{iter}=0}|$ suggests either a failed k_F selection heuristic (pseudocode lines 4-8) or overly permissive convergence thresholds f^* and τ^*.

When altering a physical network such that n_t trapping times are impacted, f-score accuracy rather than eigenvector convergence is the more relevant statistic. While f-scores are often close to zero for nodes distant from n_t, nodes that are first and second degree neighbors of n_t often have appreciable f_{n_p} values, up to 10% for the networks tested (Fig. 4). Figure 6 compares predicted and exact f_{n_p} values for neighbor nodes and randomly-selected non-neighbor nodes of $n_t = 498 \in \mathcal{H}_{500}$. In the upper panels, direct neighbors of n_t are designated with diamonds while foreigners are filled circles. F-scores predicted using the full procedure are denoted

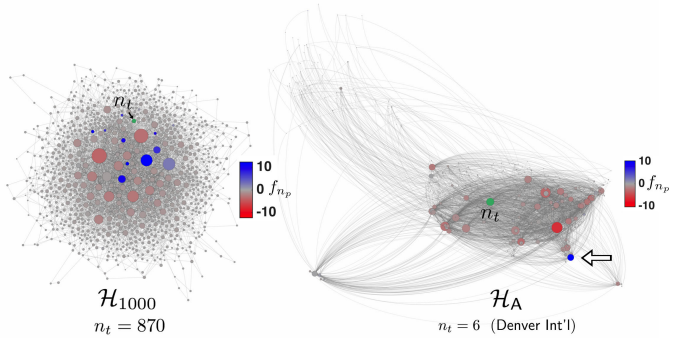

Figure 4: (Color online) F-scores for $\mathcal{H}_{1000}$ and $\mathcal{H}_{\mathsf{A}}$. A representative target node (n_t, green) for each network was selected and f-scores for all other nodes were computed and shown by colorscale. Node widths reflect total edge weight including self-loops for each node, and the spatial arrangement results from the Gephi Force Atlas algorithm [69] (left), or geographical location (right). Edge weights are not depicted. (Right) Most major airports are densely connected throughout the network and by their presence retard average transit times of a random walk to n_t, Denver International Airport. One major airport, Miami's (white arrow), however, has a substantial positive f-score, meaning average MFPTs to Denver would in fact drop by 10.3% if MIA were removed from the network (c.f. Ref. 71). F-score ranges were -3.8 to 12.3 ($\mathcal{H}_{1000}$) and -8.0 to 10.3 ($\mathcal{H}_{\mathsf{A}}$).

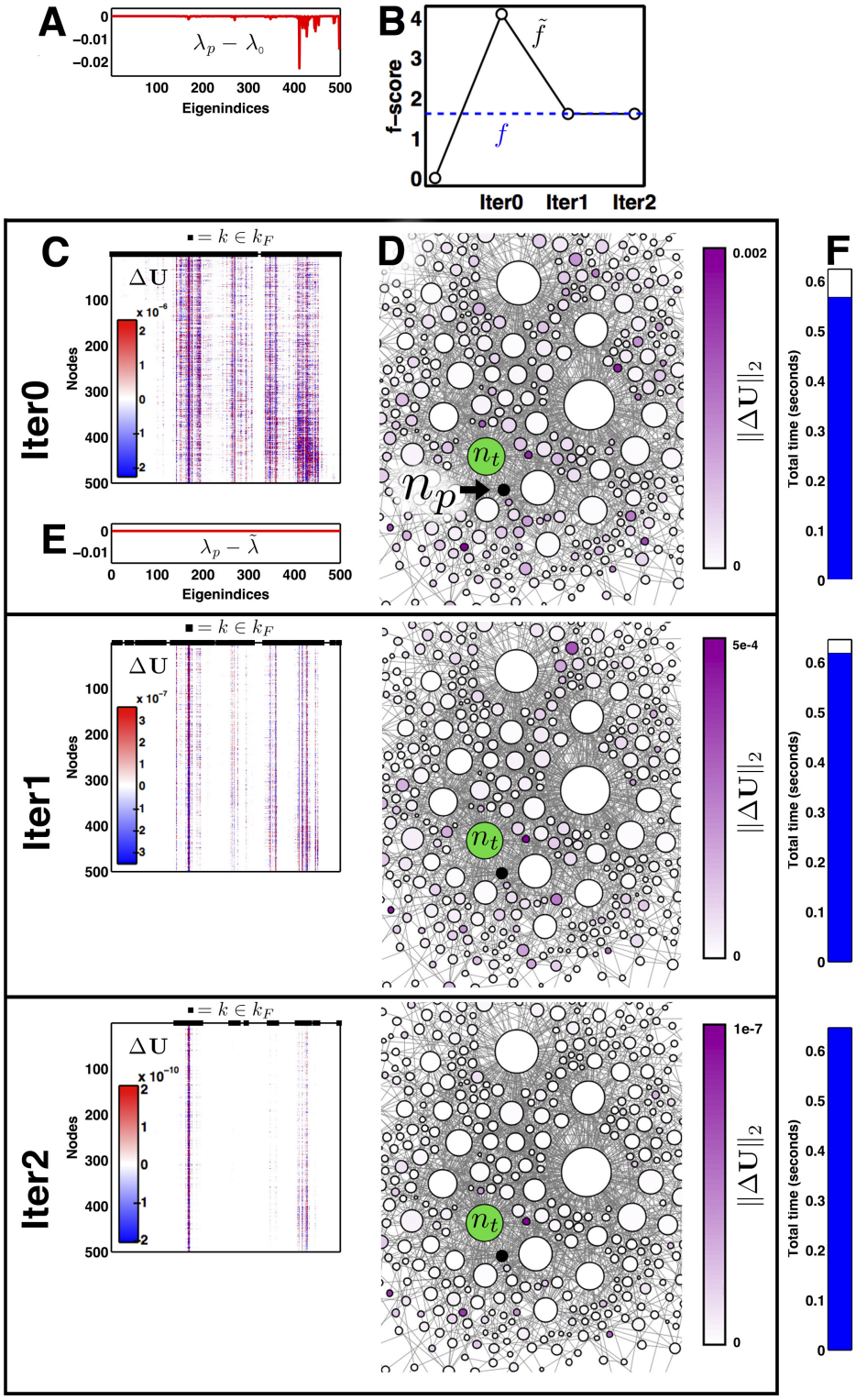

Figure 5: (Color online) Procedure visualization for $n_t = 498$, $n_p = 438 \in \mathcal{H}_{500}$ over three iterations. (A) Pre-procedure eigenvalue error, $\lambda_p - \lambda_0$. (B) F-score estimate $\tilde{f}$, black (open circles). True value, f, shown as dashed blue line. (C) Eigenvector update $\Delta\mathbf{U}$ (Eq. 9 and appendix line 16); rows are nodes (n), columns are eigenindices (k). Black squares positioned along the top horizontal axis of $\Delta\mathbf{U}$ indicate free eigenindices k_F (Eq. 13). (D) Magnitudes of eigenvector update displayed at each node n, $\|\Delta\mathbf{U}_{[n,1:N]}\|_2$. Only a subset of $\mathcal{H}_{500}$ is shown to illustrate changes in relative update magnitude. Target node $n_t = 498$, green; perturbed node $n_p = 438$, black (indicated by arrow). The magnitude of the updates decreases approximately two orders of magnitude each iteration. (E) Error of predicted eigenvalues ($\tilde{\lambda} - \lambda_p$) after one iteration, shown using the same axes as in (A). Eigenvalue predictions are only updated once (Eq. 8). (F) Aggregate runtime.

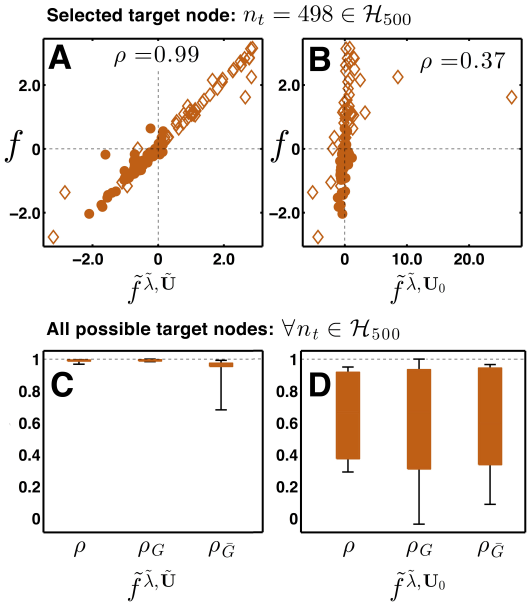

Figure 6: (Color online) Both perturbed eigenvalues and eigenvectors must be estimated for accurate f-score prediction. (A) F-score scatter plot for representative target node $n_t = 492$ in network $\mathcal{H}_{500}$. Vertical axis is the exact f-score f, horizontal axis is the predicted f-score $\tilde{f}$, for all nodes $n_p \neq 492 \in \mathcal{H}_{500}$. Diamonds denote neighbors of n_t ($n_p \in G_{n_t}$), dots foreigners ($n_p \in \bar{G}_{n_t}$). (B) Estimated f-scores $\tilde{f}$ computed from *unperturbed* eigenvectors $\mathbf{U}_0$ and estimated eigenvalues $\tilde{\lambda}$; axes as in (A). (C) and (D) The distribution of prediction accuracy for all target nodes in $\mathcal{H}_{500}$; f-scores are computed using both perturbed (C) and unperturbed (D) eigenvectors $\mathbf{U}$. A correlation of $\rho = 1.0$ means perfect prediction accuracy. Accuracy over only neighbors of each n_t is labeled ρ_G, accuracy for foreigners of each n_t is labeled $\rho_{\bar{G}}$, and correlation over all perturbed nodes is labeled as ρ. Box limits indicate upper and lower quartiles; whiskers show complete data range.

$\tilde{f}^{\tilde{\lambda},\tilde{\mathbf{U}}}$ (Fig. 6A), whereas those predicted using only updated eigenvalues are denoted $\tilde{f}^{\tilde{\lambda},\mathbf{U}_0}$ (Fig. 6B). As is apparent from the low correlation in panel B, both λ and $\mathbf{U}$ must be estimated in response to node removal if we want to accurately model f-scores for neighbors of n_t. This point should be emphasized because many centrality metrics are based only on perturbing eigenvalues and not eigenvectors [59, 72]. Panels A and B illustrate this point specifically for a single chosen n_t, but panel C shows that this discrepancy is consistent across many target nodes: correlation ρ suffers unless both $\tilde{\lambda}$ and $\tilde{\mathbf{U}}$ are estimated with perturbation theory.

Figure 7 illustrates f-score accuracy and efficiency across the six tested networks. In all panels the horizontal axis gives the relative degree of n_t; this allows us to observe that high correlations (ρ), low normalized root mean squared error (NRMSE), and modest speedup values are all consistent for highly-to-lowly connected target nodes. Each datapoint in Fig. 7B specifically is defined:

$$\text{NRMSE} = \frac{\sqrt{\frac{1}{|N_p|}\sum_{n_p \in N_p}(\tilde{f}_{n_p} - f_{n_p})^2}}{|\max f_{n_p} - \min f_{n_p}|}. \qquad (14)$$

Regarding efficiency, our procedure is about as fast as using brute force matrix inversion for networks with $N < 500$, but for larger networks we see a consistent algorithmic advantage (Fig. 7C).

A summary of efficiency and accuracy statistics is pro-

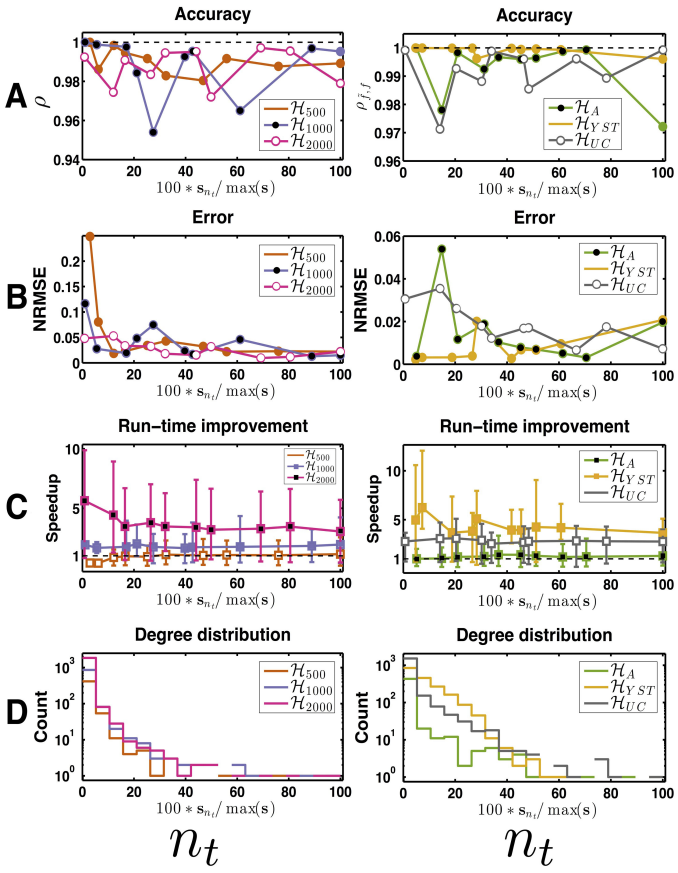

Figure 7: (Color online) F-score accuracy and efficiency for synthetic and real networks. Synthetic networks left, real networks right. Horizontal axis in all panels denotes the weighted degree of n_t as a percentage of the maximally-weighted node, $\max_{n_t} \mathbf{s}_{n_t \in \mathcal{H}}$. Target nodes n_t were selected by binning all nodes into 20 equal bins according to degree and then randomly selecting 10 target nodes equally spaced across nonempty bins. (A) Accuracy as determined by correlation of predicted f-scores, $\tilde{f}$, with ground truth f-scores, f, denoted ρ. (B) Normalized root mean squared error (Eq. 14). (C) Run-time improvement against direct method, where whiskers show maximum and minimum values. (D) Weighted degree distributions for all nodes n. Colors indicate network selection. See Table II for a summary of these results.

Table II: Accuracy and efficiency of predicted f-scores. Algorithm accuracy evaluated with correlation ρ, Spearman rank correlation ρ_s, and root mean squared error normalized by the range of exact scores, $\overline{\mathsf{NRMSE}}$. As controls we also show accuracies for f-score estimates derived without eigenvector updates, $\overline{\mathsf{NRMSE}}_{\tilde{\lambda},\mathbf{U}_0}$ and those derived from the intact spectrum, $\overline{\mathsf{NRMSE}}_{\lambda_0,\mathbf{U}_0}$ (which equates to $\tilde{f}_{n_p}=0$). The overline indicates weighted average over all tested n_t's, i.e., over all NRMSE values in Fig. 7B. Some n_p nodes are tested more than once with different target nodes n_t, so total n_p count can exceed the network size.

	Total n_t	Total n_p	ρ	ρ_s	$\overline{\mathsf{NRMSE}}_{\tilde{\lambda},\tilde{\mathbf{U}}}$	$\overline{\mathsf{NRMSE}}_{\tilde{\lambda},\mathbf{U}_0}$	$\overline{\mathsf{NRMSE}}_{\lambda_0,\mathbf{U}_0}$	Avg. speedup
$\mathcal{H}_{500}$	10	607	0.99	0.98	0.027	0.181	0.192	1.05
$\mathcal{H}_{1000}$	10	837	0.99	0.98	0.026	0.173	0.200	1.82
$\mathcal{H}_{2000}$	10	1880	0.99	0.99	0.021	0.108	0.144	3.38
$\mathcal{H}_{A}$	10	880	0.99	0.99	0.012	0.102	0.109	1.28
$\mathcal{H}_{YST}$	10	550	1.00	0.99	0.009	0.174	0.234	4.27
$\mathcal{H}_{UC}$	10	1117	0.99	0.97	0.016	0.096	0.127	2.83

vided in Table II. Because ground truth f_{n_p} values are often near zero, we ask as a control what accuracy is obtainable if λ or $\mathbf{U}$ are not updated. Table II therefore provides the average normalized room mean squared error when $\mathbf{U}$ is not updated but λ is ($\overline{\mathsf{NRMSE}}_{\tilde{\lambda},\mathbf{U}_0}$), and the same statistic is given for when all $\tilde{f}_{n_p}$'s are assumed to be zero ($\overline{\mathsf{NRMSE}}_{\lambda_0,\mathbf{U}_0}$). Again it is clear that both λ and $\mathbf{U}$ must be updated to ensure good f_{n_p} accuracy.

IV. CONCLUSIONS

Graph-spectra-derived centrality measures have proven useful for many network modeling tasks [73–76]. At least for Markov-type networks that evolve temporally, we think a concrete interpretation of centrality is provided by the spectral formulation of mean first passage times. Indeed, Eq. 6 formulates squared row vectors of $\mathbf{U}$ into a convenient quantity $\bar{\tau}_{n_t}$ where we do not need to inspect individual eigenfrequencies in order to assess the topological importance of n_p [22]. That is, individual elements of $\mathbf{U}_{[n_p,1:N]}$ may ambiguously increase or decrease upon network perturbation, but we can always interpret an f-score to signify that node n_p helps ($f_{n_p}>0$) or hinders ($f_{n_p}<0$) graph transitions to n_t. Interestingly, these small changes in transit times manifest themselves in various and discontiguous regions of the Laplacian spectrum (Figs. 3A and 5C), precluding use of many traditional sparse eigensolvers.

However, our primary focus has been to show that, algorithmically, careful selection of eigenpairs k_F can produce a less expensive approximation $\tilde{f}$ that avoids the fundamental matrix $\mathbf{Z}$. This selection cannot be made by comparing the intact and perturbed spectra (since it would require directly computing the latter), but we can guess that nodes with large Rayleigh quotient gradients (Appendix line 5) will reveal eigenpairs that either (1) will move substantially upon node perturbation (k_F) or that (2) will remain stationary (k_L). Iterative application of first-order perturbation theory to both $\tilde{\lambda}$ and $\tilde{\mathbf{U}}$ for only this selected subspace (k_F) then provides an approximate perturbed spectrum faster than dense eigendecomposition (Fig. 7C).

Because f-scores are usually linear functions of the perturbation magnitude $\epsilon\in[0,1]$, it is not necessary to completely remove node n_p from the graph and problematically decrement the rank of $\mathbf{U}$. Instead, we chose a very small ϵ so that the eigenvector shifts are small and linear estimates are accurate. This approach has the additional advantage that nodes are never disconnected from the primary graph component when a strict bottleneck node is perturbed. In these situations the f-score cannot fairly be viewed as the change in transit times were n_p to be removed since some paths to n_t would become impossible. The interpretation in these cases should be that f_{n_p} represents changes in transit times were n_p to be almost completely removed from the network.

There are many ways of describing what happens to a network when it is damaged or altered [57, 77, 78]. F-scores contribute to this discussion as well because it is sometimes robustness at some target node that is more important than global network stability, and f-scores reveal exactly that. Though many networks in the biological and social sciences surpass in size those considered here, coarse-graining methods [53] can be applied so that the resultant network is amenable to our method.

V. Additional Information

A. Author contributions

AS and CC wrote the manuscript and prepared all figures. AS was a predoctoral trainee supported by National Institutes of Health (NIH) T32 training grant T32 EB009403 as part of the HHMI-NIBIB Interfaces Initiative. This work was supported by the National Institutes of Health (grants 1R01GM105978 and 5R01GM099738).

B. Competing financial interests

The authors declare no competing financial interests.

C.

1.

APPENDIX: Fast f-score estimation

INPUT: Laplacians $\mathbf{L}$ and $\mathbf{L}_p$ of network $\mathcal{H}$, target node index n_t, and perturbed node indices N_p
OUTPUT: $\tilde{f}\left(n_p, n_t, \mathcal{H}\right)\ \forall n_p \in N_p$.

$(\mathbf{U}_0, \lambda) \leftarrow \mathsf{eig}\left(\mathbf{L}\right)$ ▷ Direct eigendecomposition

$\mathbf{U} \leftarrow \mathbf{U}_0$

$\bar{\tau}^k_{n_t} \leftarrow \left(\frac{N}{N-1}\right)\left(\frac{s u^2_{kn_t} - \left(\mathbf{s}^T \mathbf{u}_k\right) u_{kn_t}}{\lambda_k}\right)\ \forall k \neq 1$

Predict free/locked modes, k_F, k_L, by estimating $\Delta\bar{\tau}^k_{n_t}$

for $n_p \in N_p$ **do**

$\mathbf{U}_{[n_p \cup G_{n_p}, 2:N]} \leftarrow \mathbf{U}_{[n_p \cup G_{n_p}, 2:N]} - \nabla r\left(\mathbf{U}_{[n_p \cup G_{n_p}, 2:N]}\right)$ ▷ see main text Eq. 12

$\mathbf{U}_k = \mathbf{U}_k / \|\mathbf{U}_k\|$ ▷ Normalize all columns of $\mathbf{U}$

$\tilde{\Delta}\bar{\tau}^k_{n_t} \leftarrow \left(\frac{N}{N-1}\right)\left(\frac{s_p u^2_{kn_t} - \left(\mathbf{s}_p^T \mathbf{u}_k\right) u_{kn_t}}{\lambda_k}\right) - \bar{\tau}^k_{n_t},\ \forall k \neq 1$

$k_F \leftarrow \underset{k}{\mathsf{find}}\left(|\tilde{\Delta}\bar{\tau}^k_{n_t}| > \bar{\tau}^*\right),\ k_L \leftarrow \{2 \dots N\} \setminus k_F$ ▷ Select free/locked eigenpairs

Estimate perturbed eigenvalues

Select $\epsilon \sim 10^{-4}$

$\mathbf{U} \leftarrow \mathbf{U}_0$

$\tilde{\lambda}_k \leftarrow k + \epsilon * \left(\left(\mathbf{U}_k.^2\right)^T \mathbf{L}_{n_p} - 2\lambda_k u^2_{kk}\right)\ \forall k \neq 1$

Generate matrix of update weights: $\Lambda_{ij} = \left(\tilde{\lambda}_i - \tilde{\lambda}_j\right)^{-1},\ \Lambda_{ii} = 0,\ i, j \in \{2 \dots N\}$

Update U iteratively until $\tilde{f}(n_p, n_t, \mathcal{H})$ converges

$\mathsf{iter} \leftarrow 0$

Store free node indices: $n_F = \{n_p \cup n_g\}$ ▷ only n_p and neighborhood eligible for update

while $\mathsf{converged} == 0$ **do** ▷ Begin iteration for $\tilde{f}_{n_p}$

$\Delta\mathbf{U}_{[1:N,k_F]} \leftarrow \mathbf{U}_{[1:N,k_F]}\left\{\mathbf{U}^T_{[n_F,k_F]}\left(\mathbf{L}_{p[n_F,1:N]}\mathbf{U}_{[1:N,k_F]} - \mathbf{U}_{[n_F,k_F]} * \mathbf{I}\tilde{\lambda}_{k_F}\right) . * \Lambda_{[k_F,k_F]}\right\}$ ▷ see Eq. 9

$\tilde{\mathbf{U}} \leftarrow \mathbf{U} + \Delta\mathbf{U}$

$\tilde{\bar{\tau}}^k_{n_t} \leftarrow \left(\frac{N}{N-1}\right)\left(\frac{s_p \tilde{u}^2_{kn_t} - \left(\mathbf{s}_p^T \tilde{\mathbf{u}}_k\right) \tilde{u}_{kn_t}}{\tilde{\lambda}_k}\right),\ \forall k \in k_F$ ▷ Compute updated $\tilde{\bar{\tau}}^k_{n_t}$

$\tilde{f}_{\mathsf{iter}}(n_p, n_t) \leftarrow (1/\epsilon) * \frac{\sum_{k=2}^N \tilde{\bar{\tau}}^k - \sum_{k=2}^N \bar{\tau}^k}{\sum_{k=2}^N \bar{\tau}^k}$ ▷ Estimate new f_{n_p}

$\mathsf{converged} \leftarrow \left|\tilde{f}_{\mathsf{iter}} - \tilde{f}_{\mathsf{iter}-1}\right| / \left|\tilde{f}_{\mathsf{iter}-1}\right| < f^*$

if !converged **then**

$k_F \leftarrow \mathsf{find}\left(|\tilde{\Delta}\bar{\tau}^{k_F}_{n_t}| > \bar{\tau}^*\right)$

$n_F \leftarrow \{1 \dots N\}$ ▷ All nodes now eligible for update

$\mathbf{U} \leftarrow \tilde{\mathbf{U}}$

$\mathsf{iter} \leftarrow \mathsf{iter} + 1$

end if

end while

end for

[1] Vittoria Colizza, Alain Barrat, Marc Barthélemy, and Alessandro Vespignani, "The role of the airline transportation network in the prediction and predictability of global epidemics." Proceedings of the National Academy of Sciences of the United States of America **103**, 2015–2020 (2006).

[2] Vittoria Colizza, Romualdo Pastor-Satorras, and Alessandro Vespignani, "Reaction–diffusion processes and metapopulation models in heterogeneous networks," Nature Physics **3**, 276–282 (2007).

[3] J Memmott, N M Waser, and M V Price, "Tolerance of pollination networks to species extinctions," Proceedings of the Royal Society B: Biological Sciences **271**, 2605–2611 (2004).

[4] Maksim Kitsak, Lazaros K Gallos, Shlomo Havlin, Fredrik Liljeros, Lev Muchnik, H Eugene Stanley, and Hernán A Makse, "Identification of influential spreaders in complex networks," Nature Physics **6**, 888–893 (2010).

[5] Mauricio Barahona and Louis M Pecora, "Synchronization in small-world systems." Physical Review Letters **89**, 054101 (2002).

[6] Réka Albert and Albert-Laszlo Barabasi, "Statistical mechanics of complex networks," Reviews of modern physics **74**, 47 (2002).

[7] Piet Van Mieghem, "Epidemic phase transition of the SIS type in networks," EPL (Europhysics Letters) **97**, 48004 (2012).

[8] Ed Bullmore and Olaf Sporns, "Complex brain networks: graph theoretical analysis of structural and functional systems," Nature Reviews Neuroscience **10**, 186–198 (2009).

[9] A T Lawniczak, A Gerisch, and K Maxie, "Effects of randomly added links on a phase transition in data network traffic models," Proc of the 3rd International DCDIS Conference (2003).

[10] John D Chodera and Vijay S Pande, "The social network (of protein conformations)." Proceedings of the National Academy of Sciences of the United States of America **108**, 12969–12970 (2011).

[11] Giuliano Andrea Pagani and Marco Aiello, "The Power Grid as a complex network: A survey," Physica A: Statistical Mechanics and its Applications **392**, 2688–2700 (2013).

[12] D J Watts and S H Strogatz, "Collective dynamics of 'small-world' networks." Nature **393**, 440–442 (1998).

[13] B A Prakash, J Vreeken, and C Faloutsos, "Efficiently spotting the starting points of an epidemic in a large graph," Knowledge and information systems (2014).

[14] Alain Barrat, Marc Barthélemy, and Alessandro Vespignani, *Dynamical Processes on Complex Networks* (Cambridge University Press, 2008).

[15] Hui Wang, Jinyuan Huang, Xiaomin Xu, and Yanghua Xiao, "Damage attack on complex networks," Physica A: Statistical Mechanics and its Applications , 1–15 (2014).

[16] Ricardo Gutiérrez, Irene Sendiña-Nadal, Massimiliano Zanin, David Papo, and Stefano Boccaletti, "Targeting the dynamics of complex networks." Scientific Reports **2**, 396–396 (2012).

[17] Venky Soundararajan and Murali Aravamudan, "Global connectivity of hub residues in Oncoprotein structures encodes genetic factors dictating personalized drug response to targeted Cancer therapy," Scientific Reports **4**, 7294 (2014).

[18] Michele Benzi and Christine Klymko, "A matrix analysis of different centrality measures," arXiv preprint arXiv:1312.6722 (2013).

[19] Gergana Bounova and Olivier de Weck, "Overview of metrics and their correlation patterns for multiple-metric topology analysis on heterogeneous graph ensembles," Physical Review E **85**, 016117 (2012).

[20] Eleanor R Brush, David C Krakauer, and Jessica C Flack, "A Family of Algorithms for Computing Consensus about Node State from Network Data," PLoS computational biology **9**, e1003109 (2013).

[21] L da F Costa, F A Rodrigues, G Travieso, and P R Villas Boas, "Characterization of complex networks: A survey of measurements," Advances in Physics **56**, 167–242 (2007).

[22] Piet Van Mieghem, "Graph eigenvectors, fundamental weights and centrality metrics for nodes in networks," arXiv preprint arXiv:1401.4580 (2014).

[23] Gregory R Bowman and Vijay S Pande, "Protein folded states are kinetic hubs." Proceedings of the National Academy of Sciences of the United States of America **107**, 10890–10895 (2010).

[24] Alex Dickson and Charles Brooks, III, "Quantifying hub-like behavior in protein folding networks." Journal of Chemical Theory and Computation **8**, 3044–3052 (2012).

[25] Alex Dickson and Charles Brooks, III, "Native States of Fast-Folding Proteins Are Kinetic Traps," Journal of the American Chemical Society **135**, 4729–4734 (2013).

[26] Hongxiao Liu and Zhongzhi Zhang, "Laplacian spectra of recursive treelike small-world polymer networks: Analytical solutions and applications," The Journal of Chemical Physics **138**, 114904 (2013).

[27] B Aditya Prakash, Jilles Vreeken, and Christos Faloutsos, "Spotting culprits in epidemics: How many and which ones?" IEEE International Conference on Data Mining **12**, 11–20 (2012).

[28] Patrick N McGraw and Michael Menzinger, "Laplacian spectra as a diagnostic tool for network structure and dynamics," Physical Review E **77**, 031102 (2008).

[29] Scott D Pauls and Daniel Remondini, "Measures of centrality based on the spectrum of the Laplacian," Physical Review E **85**, 066127 (2012).

[30] Gitanjali Yadav and Suresh Babu, "NEXCADE: Perturbation Analysis for Complex Networks," PLoS ONE **7**, e41827 (2012).

[31] Andrew Y Ng, Alice X Zheng, and Michael I Jordan, "Link analysis, eigenvectors and stability," International Joint Conference on Artificial Intelligence **17**, 903–910 (2001).

[32] Gourab Ghoshal and Albert-Laszlo Barabasi, "Ranking stability and super-stable nodes in complex networks," Nature Communications **2**, 392–7 (2011).

[33] Ernesto Estrada and Juan Rodríguez-Velázquez, "Subgraph centrality in complex networks," Physical Review E **71**, 056103 (2005).

[34] Ernesto Estrada, Naomichi Hatano, and Michele Benzi, "The physics of communicability in complex networks," Physics Reports **514**, 89–119 (2012).

[35] Juan Chen, Jun-an Lu, Choujun Zhan, and Guanrong Chen, "Laplacian spectra and synchronization processes on complex networks," in *Handbook of Optimization in Complex Networks* (Springer, 2012) pp. 81–113.

[36] Remi Monasson, "Diffusion, localization and dispersion relations on "small-world" lattices," The European Physical Journal B-Condensed Matter and Complex Systems **12**, 555–567 (1999).

[37] C Li, H Wang, W de Haan, C J Stam, and Piet Van Mieghem, "The correlation of metrics in complex networks with applications in functional brain networks," Journal of Statistical Mechanics: Theory and Experiment **2011**, P11018 (2011).

[38] Andrej Savol and Chakra S Chennubhotla, "Quantifying the Sources of Kinetic Frustration in Folding Simulations of Small Proteins," Journal of Chemical Theory and Computation **10**, 2964–2974 (2014).

[39] Zhongzhi Zhang, Alafate Julaiti, Baoyu Hou, Hongjuan Zhang, and Guanrong Chen, "Mean first-passage time for random walks on undirected networks," The European Physical Journal B **84**, 691–697 (2011).

[40] Peter G Doyle and James Laurie Snell, *Random Walks and Electric Networks*, Carus Monographs (Mathematical Association of America, Washington, 1984).

[41] J G Kemeny and James Laurie Snell, *Finite Markov Chains* (Springer Verlag, New York, 1976).

[42] Murray Shanahan, "Metastable chimera states in community-structured oscillator networks." Chaos **20**, 013108–013108 (2010).

[43] Pablo Villegas, Paolo Moretti, and Miguel A Muñoz, "Frustrated hierarchical synchronization and emergent complexity in the human connectome network." Scientific Reports **4**, 5990–5990 (2014).

[44] Ulrike von Luxburg, Agnes Radl, and Matthias Hein, "Hitting and commute times in large graphs are often misleading," arXiv preprint arXiv:1003.1266 (2010).

[45] Ulrike von Luxburg, Agnes Radl, and Matthias Hein, "Getting lost in space: Large sample analysis of the commute distance," Advances in Neural Information Processing Systems **23**, 2622–2630 (2010).

[46] M E J Newman, "Power laws, Pareto distributions and Zipf's law," Audio and Electroacoustics Newsletter, IEEE (2004), 10.1080/00107510500052444.

[47] Lars Kiemer, Stefano Costa, Marius Ueffing, and Gianni Cesareni, "WI-PHI: A weighted yeast interactome enriched for direct physical interactions," Proteomics **7**, 932–943 (2007).

[48] Pietro Panzarasa, Tore Opsahl, and Kathleen M Carley, "Patterns and Dynamics of Users' Behavior and Interaction: Network Analysis of an Online Community," Journal of the American Society for Information Science and Technology **60**, 911–932 (2009).

[49] László Lovász, "Random walks on graphs: A survey," Combinatorics, Paul erdos is eighty **2**, 1–46 (1993).

[50] Ulrike von Luxburg, "A tutorial on spectral clustering," Statistics and Computing **17**, 395–416 (2007).

[51] William Hager, "Updating the inverse of a matrix." SIAM review , 221–2339 (1989).

[52] Yuan Lin and Zhongzhi Zhang, "Random walks in weighted networks with a perfect trap: An application of Laplacian spectra." Physical Review E **87**, 062140 (2013).

[53] David Gfeller and Paolo De Los Rios, "Spectral coarse graining and synchronization in oscillator networks." Physical Review Letters **100**, 174104–174104 (2008).

[54] Stéphane S Lafon and Ann B AB Lee, "Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization." IEEE Transactions on Pattern Analysis and Machine Intelligence **28**, 1393–1403 (2006).

[55] Dilip Krishnan, Raanan Fattal, and Richard Szeliski, "Efficient preconditioning of laplacian matrices for computer graphics," ACM Transactions on Graphics **32**, 1 (2013).

[56] X Qi, E Fuller, Q Wu, Y Wu, and C Q Zhang, "Laplacian centrality: A new centrality measure for weighted networks," Information Sciences (2012), 10.1016/j.ins.2011.12.027.

[57] Piet Van Mieghem, Dragan Stevanović, Fernando Kuipers, Cong Li, Ruud van de Bovenkamp, Daijie Liu, and Huijuan Wang, "Decreasing the spectral radius of a graph by link removals." Physical Review E **84**, 016101 (2011).

[58] Alexander C Kalloniatis, "From incoherence to synchronicity in the network Kuramoto model." Physical Review E **82**, 066202–066202 (2010).

[59] Attilio Milanese, Jie Sun, and Takashi Nishikawa, "Approximating spectral impact of structural perturbations in large networks," Physical Review E **81**, 046112 (2010).

[60] Steve Butler, "Interlacing for weighted graphs using the normalized Laplacian," Electronic Journal of Linear Algebra **16**, 87 (2007).

[61] Aida Abiad, Miquel A Fiol, Willem H Haemers, and Guillem Perarnau, "An interlacing approach for bounding the sum of Laplacian eigenvalues of graphs," Linear Algebra and its Applications **448**, 11–21 (2014).

[62] Baofeng Wu, Jiayu Shao, and Xiying Yuan, "Deleting vertices and interlacing Laplacian eigenvalues," Chinese Annals of Mathematics, Series B **31**, 231–236 (2010).

[63] J H Wilkinson, *The algebraic eigenvalue problem* (Oxford University Press, 1965).

[64] X L Liu and C S Oliveira, "Iterative modal perturbation and reanalysis of eigenvalue problem," Communications in Numerical Methods in Engineering **19**, 263–274 (2003).

[65] David MacKay, *Information theory, inference, and learning algorithms* (Cambridge University Press, 2003).

[66] V Hernandez, J E Roman, A Tomas, and V Vidal, "Arnoldi methods in SLEPc," SLEPc Technical Report STR-4 (2007).

[67] Loyd Trefethen and David Bau, *Numerical Linear Algebra* (SIAM, Philadelphia, 1997).

[68] MATLAB, *version 7.14.0.739 (R2012a)* (The MathWorks Inc., Natick, Massachusetts).

[69] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy, "Gephi: an open source software for exploring and manipulating networks." ICWSM , 361–362 (2009).

[70] Lev Muchnik, *Complex Networks Package for MatLab (Version 1.6)* (www.levmuchnik.net).

[71] T Verma, N A M Araújo, and Hans J Herrmann, "Revealing the structure of the world airline network," Scientific Reports **4** (2014), 10.1038/srep05638.

[72] Juan G Restrepo, Edward Ott, and Brian R Hunt, "Characterizing the dynamical importance of network nodes and links." Physical Review Letters **97**, 094102 (2006).

[73] Stefano Boccaletti, V Latora, Y Moreno, and M Chavez, "Complex networks: Structure and dynamics," Physics

reports (2006).
[74] Dragoš Cvetković, Peter Rowlinson, and Slobodan Simić, *An Introduction to the Theory of Graph Spectra* (Cambridge University Press, 2009).
[75] Ernesto Estrada and Naomichi Hatano, "A vibrational approach to node centrality and vulnerability in complex networks," Physica A: Statistical Mechanics and its Applications **389**, 3648–3660 (2010).
[76] M T Schaub, J Lehmann, and S N Yaliraki, "Structure of complex networks: Quantifying edge-to-edge relations by failure-induced flow redistribution," Network Science **2**, 66–89 (2014).
[77] D Liu, H Wang, and Piet Van Mieghem, "Spectral perturbation and reconstructability of complex networks," Physical Review E **81**, 016101 (2010).
[78] Ernesto Estrada, Eusebio Vargas-Estrada, and Hiroyasu Ando, "Communicability Angles Reveal Critical Edges for Network Consensus Dynamics," ArXiv e-prints (2015).