Unbiased degree-preserving randomization of directed binary networks

E. S. Roberts[1,2] and A. C. C. Coolen[1,2,3]

[1]*Department of Mathematics, King's College London, The Strand, London WC2R 2LS, United Kingdom*

[2]*Randall Division of Cell and Molecular Biophysics, King's College London, New Hunts House, London SE1 1UL, United Kingdom*

[3]*London Institute for Mathematical Sciences, 35a South Street, Mayfair, London W1K 2XF, United Kingdom*

(Received 20 December 2011; published 5 April 2012)

Randomizing networks using a naive "accept-all" edge-swap algorithm is generally biased. Building on recent results for nondirected graphs, we construct an ergodic detailed balance Markov chain with nontrivial acceptance probabilities for directed graphs, which converges to a strictly uniform measure and is based on edge swaps that conserve all in and out degrees. The acceptance probabilities can also be generalized to define Markov chains that target any alternative desired measure on the space of directed graphs in order to generate graphs with more sophisticated topological features. This is demonstrated by defining a process tailored to the production of directed graphs with specified degree-degree correlation functions. The theory is implemented numerically and tested on synthetic and biological network examples.

DOI: 10.1103/PhysRevE.85.046103 PACS number(s): 89.75.Hc, 87.10.Rt, 87.18.−h

I. INTRODUCTION

When seeking to assess the statistical relevance of observations made in real networks, there are three routes available. One could generate null-model networks for hypothesis testing from scratch, constrained by the values of observed parameters in the real network (e.g., using the Molloy-Reed stub-joining method [1] or the Barabási-Albert preferential attachment model [2]). Alternatively, one could generate null-model networks by randomizing the original network, using dynamical rules that leave the values of relevant parameters invariant [3]. The final option is to use analytical methods to find ensemble averages for the observable of interest; see, e.g., [4–7].

The null-model approach is appealing in its conceptual simplicity. It effectively provides synthetic "data", which can be analyzed in the same way as the real data set. One can then learn which observed properties are particular to the real data set and which are common within the ensemble.

Applications of network null models are wide ranging and central to network science. Reference [8] applies null models to identify over-represented "motifs" in the transcriptional regulation network of *E. coli*. Reference [9] discusses adapting the Watts-Strogartz method to generating random networks to model power grids. Reference [10] explores motifs found within an interfirm network. Reference [11] uses network null models to study social networks. Reference [12] compares topological properties of interaction and transcription regulatory networks in yeast with randomized null-model networks and postulates that links between highly connected proteins are suppressed in protein interaction networks. Reference [13] discusses the challenges of specifying a suitable matrix null model in the field of ecology.

It is crucial that the synthetic networks generated as null models are representative of the underlying ensembles. Any inadvertent bias in the network generation process may invalidate the hypothesis test. It is therefore worrying that the two most popular methods to randomize or generate null networks are in fact known to be biased. That is, the processes do not reach every valid network in the ensemble with equal probability. The common implementation of the stub-joining method, where invalid edges are rejected but the process subsequently continues (as opposed to starting from the beginning), is known to be biased [14–16]. Even if upon invalid edge rejection the stub-joining process is restarted, it is not clear whether the graphs produced would be unbiased (we are not aware of any published proof). Reference [7] proposes a version of the stub-joining algorithm with an analytical correction to network observables. Similarly, the conventional accept-all edge-swap process (see, e.g., [17]) is also known to be biased [18]: graphs on which many swaps can be executed are generated more often. The effects of these biases may in the past not always have been serious [19], but using biased algorithms for producing null models is fundamentally unsound, and unacceptable when there are rigorous unbiased alternatives [18].

In this paper, we build on the work of [18] and [3] and define a Markov chain Monte Carlo (MCMC) process, based on ergodic in and out degrees, preserving edge-swap moves that act on *directed* networks. We first calculate correct move acceptance probabilities for the process to sample the space of all allowed directed graphs uniformly. We then extend the theory in order for the process to evolve to any desired measure on the space of directed graphs. Attention is paid to adapting our results for efficient numerical implementation. We also identify under which circumstances the error made by doing accept-all edge swaps is immaterial. We apply our theory to real and synthetic networks.

II. AN ERGODIC AND UNBIASED RANDOMIZATION PROCESS WHICH PRESERVES IN AND OUT DEGREES

A. Edge-swap moves

The canonical moves for degree-preserving randomization of graphs are the so-called edge swaps; see, e.g., [18,20,21]. The undirected version of the edge swap is illustrated in Fig. 1; a generalization to directed graphs is found in [3]. The authors of [3] define a move—which we will refer to as a *square swap*—starting from a set of four entries from the connectivity matrix $\boldsymbol{c} \in \{0,1\}^{N^2}$ of a directed binary N-node graph, defined by node pairs $\{(i_1,j_1),(i_1,j_2),(i_2,j_2),(i_2,j_1)\}$ such that the corresponding entries $\{c_{i_1j_1},c_{i_1j_2},c_{i_2j_2},c_{i_2j_1}\}$ are

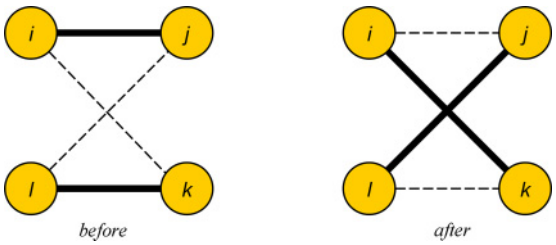

FIG. 1. (Color online) The undirected edge swap. This is the canonical choice for the elementary moves of an ergodic degree-preserving randomization process on undirected networks.

alternately 0 and 1, and not "structural" (i.e., they are allowed to vary). As for the undirected case, the elementary edge-swap move is defined by swapping the 0 and 1 entries, i.e., $\{c_{i_1 j_1}, c_{i_1 j_2}, c_{i_2 j_2}, c_{i_2 j_1}\} \to \{1-c_{i_1 j_1}, 1-c_{i_1 j_2}, 1-c_{i_2 j_2}, 1-c_{i_2 j_1}\}$. It is built into the definition that a move is not allowed if the pair of nodes to which an edge is rewired is already connected. The authors of [3] prove that if self-interactions are permitted, then the repeated application of such moves can transform any binary matrix $\boldsymbol{c}_A$ to any other binary matrix $\boldsymbol{c}_B$ with the same in and out degree distributions. However, if we require in addition that the diagonal elements of all $\boldsymbol{c}$ are 0 (i.e., we forbid self-interactions), then the edge swap defined above is no longer sufficient to ensure ergodicity. To remedy this problem, the authors of [3] introduce a further move, which we will call a *triangle swap*. This move also gives us the simplest demonstration of two valid configurations that cannot be connected by square-type swaps. The *square swap* and the *triangle swap* are illustrated in Fig. 2; in combination these two moves allow us to transform between any two directed binary matrices which have the same in and out degrees, even if self-interactions are forbidden [3].

A stochastic process defined as accepting all randomly selected moves from the above set is ergodic but biased. This was already observed in [3], where the authors proposed a "switch and hold" algorithm, which involves the number of states accessible in one move from a configuration (its mobility), and the maximum possible number of states accessible in one move from any network in the ensemble (the degrees of a hypergraph, in the language of later publications). In [18], the problem was studied for undirected graphs; it was shown how move acceptance probabilities should be defined to guarantee stochastic evolution by edge swapping to any desired measure on the space of nondirected graphs. The analysis in [18] is quite general, and briefly reproduced in Sec. II B below. Here we will adapt their calculations to directed graphs and include the new moves defined by [3]. This will result in a Markovian process based on edge swapping that will equilibrate to any desired measure on the space of *directed* graphs.

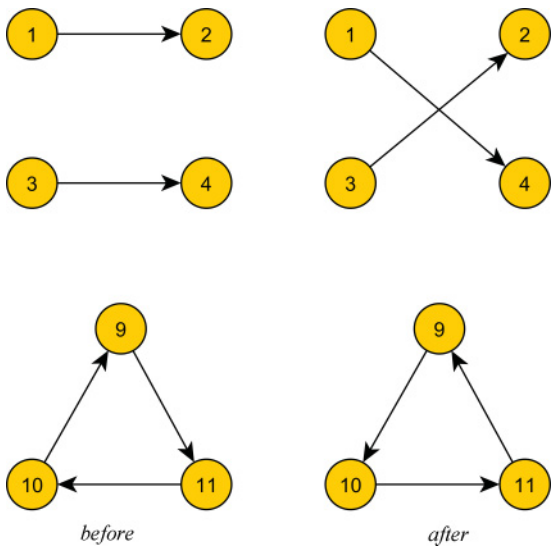

FIG. 2. (Color online) The *square swap* (top) and *triangle swap* (bottom). In combination, these two represent the canonical choice for the elementary moves of an ergodic degree-preserving randomization process on directed networks without self-interactions.

B. Outline of the general theory

This section briefly summarizes results of [18] which will be used in the next section. We define an adjacency matrix $\boldsymbol{c} = \{c_{ij}\}$, where $c_{ij} = 1$ if and only if there is a directed link from node j to node i. We denote the set of all such graphs as C. The aim is to define and study constrained Markov chains for the evolution of $\boldsymbol{c}$ in some subspace $\Omega \in C$. This is a discrete time stochastic process, where the probability $p_t(\boldsymbol{c})$ of observing a graph $\boldsymbol{c}$ at time t evolves according to

$$\forall \boldsymbol{c} \in \Omega : p_{t+1}(\boldsymbol{c}) = \sum_{\boldsymbol{c}' \in \Omega} W(\boldsymbol{c}|\boldsymbol{c}') p_t(\boldsymbol{c}'), \tag{1}$$

where $t \in \mathbb{N}$ and $W(\boldsymbol{c}|\boldsymbol{c}')$ is a transition probability. We require the process to have three additional properties:

(1) Each elementary move F can only act on a subset of all possible graphs.

(2) The process converges to the invariant measure

$$p_\infty(\boldsymbol{c}) = Z^{-1} e^{-H(\mathbf{c})},$$

where Z is a normalizing constant and $H(\boldsymbol{c})$ is called the Hamiltonian.

(3) Each move F has a unique inverse, which acts on the same subset of states as F itself.

The second property is the crucial one, in that it defines the equilibrium probability distribution of $\boldsymbol{c}$ after the Markov process has equilibrated. For example, aiming for a flat distribution is equivalent to requiring $H(\boldsymbol{c})$ to be constant.

We exclude the identity move from the set Φ of all moves, and we define an indicator function $I_F(\boldsymbol{c})$ where $I_F(\boldsymbol{c}) = 1$ if and only if the move $\boldsymbol{c} \to F\boldsymbol{c}$ is allowed. The transition probabilities are constructed to obey detailed balance,

$$\forall \boldsymbol{c}, \boldsymbol{c}' \in \Omega : W(\boldsymbol{c}|\boldsymbol{c}') p_\infty(\boldsymbol{c}') = W(\boldsymbol{c}'|\boldsymbol{c}) p_\infty(\boldsymbol{c}). \tag{2}$$

At each step, a candidate move $F \in \Phi$ is drawn with probability $q(F|\boldsymbol{c}')$, where $\boldsymbol{c}'$ is the current state. The move is accepted with some probability $A(F\boldsymbol{c}'|\boldsymbol{c}')$. Our aim is to define a suitable $A(F\boldsymbol{c}'|\boldsymbol{c}')$ in a way such that our process will achieve the desired equilibrium distribution $p_\infty(\boldsymbol{c})$.

The relationship between the transition probability and the probability of drawing and accepting a particular move is clearly

$$W(\boldsymbol{c}|\boldsymbol{c}') = \sum_{F \in \Phi} q(F|\boldsymbol{c}') \{\delta_{\mathbf{c}, F\mathbf{c}'} A(F\boldsymbol{c}'|\boldsymbol{c}') + \delta_{\mathbf{c},\mathbf{c}'} [1 - A(F\boldsymbol{c}'|\boldsymbol{c}')]\}. \tag{3}$$

Insertion into (2) then leads to the following condition, which must be satisfied by $A(F\boldsymbol{c}'|\boldsymbol{c}')$ and $q(F|\boldsymbol{c}')$:

$$(\forall \boldsymbol{c} \in \Omega)(\forall F \in \Phi) : q(F|\boldsymbol{c})A(F\boldsymbol{c}|\boldsymbol{c})e^{-H(\boldsymbol{c})} = q(F^{-1}|F\boldsymbol{c})A(\boldsymbol{c}|F\boldsymbol{c})e^{-H(F\boldsymbol{c})}. \quad (4)$$

We define the mobility $n(\boldsymbol{c})$ to be the number of moves which can act on each state: $n(\boldsymbol{c}) = \sum_{F\in\Phi} I_F(\boldsymbol{c})$. If the candidate moves are drawn randomly with equal probabilities from the set of all moves allowed to act, then we find that (4) reduces to

$$A(\boldsymbol{c}|\boldsymbol{c}') = \frac{n(\boldsymbol{c}')e^{-\frac{1}{2}[H(\boldsymbol{c})-H(\boldsymbol{c}')]}}{n(\boldsymbol{c}')e^{-\frac{1}{2}[H(\boldsymbol{c})-H(\boldsymbol{c}')]} + n(\boldsymbol{c})e^{\frac{1}{2}[H(\boldsymbol{c})-H(\boldsymbol{c}')]}}. \quad (5)$$

If we make the simplest choice, $H(\boldsymbol{c}) = \text{const}$, then the above process will asymptotically sample all graphs with the imposed degree sequence uniformly. To sample this constrained space of graphs with alternative nontrivial probabilities $p_\infty(\boldsymbol{c})$, we would choose $H(\boldsymbol{c}) = -\ln p_\infty(\boldsymbol{c}) + \text{const}$.

Equation (4) also shows what would happen if we were to sample with $A(\boldsymbol{c}|\boldsymbol{c}') \equiv 1$ for all $(\boldsymbol{c},\boldsymbol{c}')$, i.e., for accept-all edge swapping: the detailed balance condition would give

$$(\forall \boldsymbol{c} \in \Omega)(\forall F \in \Phi) : \frac{e^{-H(\boldsymbol{c})}}{n(\boldsymbol{c})} = \frac{e^{-H(F\boldsymbol{c})}}{n(F\boldsymbol{c})}. \quad (6)$$

For this to be satisfied, we require both sides of the expression to evaluate to a constant. Hence $e^{-H(\boldsymbol{c})} \propto n(\boldsymbol{c})$, so the naive process will converge to the nonuniform measure,

$$p_\infty(\boldsymbol{c}) = Z^{-1}n(\boldsymbol{c}). \quad (7)$$

This is the undesirable bias of accept-all edge swapping. It has a clear intuitive explanation. The mobility $n(\boldsymbol{c})$ is the number of allowed moves on network $\boldsymbol{c}$, which is equal to the number of inverse moves through which $\boldsymbol{c}$ can be reached in one step from another member of the ensemble. The likelihood of seeing a network $\boldsymbol{c}$ upon equilibration of the process is proportional to the number of entry points that $\boldsymbol{c}$ offers the process.

C. Calculation of the mobilities for directed networks

Since the two types of moves required for ergodic evolution of directed graphs, viz., the *square swap* and the *triangle swap*, are clearly distinct, the mobility of a graph $\boldsymbol{c}$ is given by $n(\boldsymbol{c}) = n_\square(\boldsymbol{c}) + n_\triangle(\boldsymbol{c})$, where $n_\square(\boldsymbol{c})$ and $n_\triangle(\boldsymbol{c})$ count the number of possible moves of each type that can be executed on $\boldsymbol{c}$.

To find $n_\square(\boldsymbol{c})$, we need to calculate how many distinct link-alternating cycles of length 4 can be chosen in graph $\boldsymbol{c}$. We exclude self-interactions, so our cycles must involve four distinct nodes. The total number of such moves can be written as

$$n_\square(\boldsymbol{c}) = \frac{1}{2}\sum_{ijk\ell} \bar\delta_{jk}\bar\delta_{\ell i}\bar\delta_{ik}\bar\delta_{j\ell}c_{ij}c_{k\ell}\bar c_{kj}\bar c_{i\ell}, \quad (8)$$

where the prefactor compensates for the symmetry, and where we used the shorthand $\bar c_{kj} = 1 - c_{kj}$ and $\bar\delta_{jk} = 1 - \delta_{jk}$. Expanding this shorthand in (8) gives, after some further bookkeeping of terms and with $(\boldsymbol{c}^\dagger)_{ij} = c_{ji}$,

$$n_\square(\boldsymbol{c}) = \frac{1}{2}\text{Tr}(\boldsymbol{c}\boldsymbol{c}^\dagger\boldsymbol{c}\boldsymbol{c}^\dagger) - \sum_{ij} k_i^{\text{in}}c_{ij}k_j^{\text{out}} + \text{Tr}(\boldsymbol{c}\boldsymbol{c}^\dagger\boldsymbol{c}) + \frac{1}{2}\text{Tr}(\boldsymbol{c}^2) + \frac{1}{2}N^2\langle k\rangle^2 - \sum_j k_j^{\text{in}}k_j^{\text{out}}, \quad (9)$$

with $\langle k\rangle = N^{-1}\sum_i k_i^{\text{in}} = N^{-1}\sum_i k_i^{\text{out}}$. We next repeat the calculation for the case of the *triangle swap*. For easier manipulations, we introduce a new matrix $\boldsymbol{c}^\ddagger$ of double links, defined via $(\boldsymbol{c}^\ddagger)_{ij} = c_{ij}c_{ji}$. We then find

$$\begin{aligned} n_\triangle(\boldsymbol{c}) &= \frac{1}{3}\sum_{ijk} \bar\delta_{ij}\bar\delta_{jk}\bar\delta_{ki}c_{ij}c_{jk}c_{ki}\bar c_{ji}\bar c_{kj}\bar c_{ik} \\ &= \frac{1}{3}\{\text{Tr}(\boldsymbol{c}^3) - 3\text{Tr}(\boldsymbol{c}^\ddagger\boldsymbol{c}^2) + 3\text{Tr}(\boldsymbol{c}^{\ddagger 2}\boldsymbol{c}) + -\text{Tr}(\boldsymbol{c}^{\ddagger 3})\} \\ &= \frac{1}{3}\text{Tr}[(\boldsymbol{c} - \boldsymbol{c}^\ddagger)^3]. \end{aligned} \quad (10)$$

In combination, (9) and (10) give us an explicit and exact formula for the graph mobility $n(\boldsymbol{c}) = n_\square(\boldsymbol{c}) + n_\triangle(\boldsymbol{c})$ and, hence, via (5), a fully exact MCMC process for generating random graphs with prescribed sequences and any desired probability measure in the standard form $Z^{-1}\exp[-H(\boldsymbol{c})]$. Since (9) and (10) cannot be written in terms of the degree sequence only, neglecting the mobility (as with accept-all edge swapping) would always introduce a bias into the sampling process.

III. PROPERTIES AND IMPACT OF GRAPH MOBILITY

A. Bounds on the mobility

We will now derive bounds on the sizes of the mobility terms. This will help us characterize degree distributions for which the error due to accept-all edge swapping is not expected to be material.

We first observe that

$$n_\triangle(\boldsymbol{c}) = \frac{1}{3}\sum_{ijk} c_{ij}(1-c_{ji})c_{jk}(1-c_{kj})c_{ki}(1-c_{ik}) \leqslant \frac{1}{3}\text{Tr}(\boldsymbol{c}^3).$$

Hence, the mobility $n(\boldsymbol{c}) = n_\square(\boldsymbol{c}) + n_\triangle(\boldsymbol{c})$ obeys

$$n(\boldsymbol{c}) \leqslant \frac{1}{2}\text{Tr}(\boldsymbol{c}\boldsymbol{c}^\dagger\boldsymbol{c}\boldsymbol{c}^\dagger) - \sum_{ij} k_i^{\text{in}}c_{ij}k_j^{\text{out}} + \text{Tr}(\boldsymbol{c}\boldsymbol{c}^\dagger\boldsymbol{c}) + \frac{1}{2}\text{Tr}(\boldsymbol{c}^2) + \frac{1}{2}N^2\langle k\rangle^2 - \sum_j k_j^{\text{in}}k_j^{\text{out}} + \frac{1}{3}\text{Tr}(\boldsymbol{c}^3). \quad (11)$$

We find upper bounds for most of the terms above by applying the simple inequality $c_{ij}c_{kl} \leqslant \frac{1}{2}(c_{ij} + c_{kl})$, which gives, e.g.,

$$\text{Tr}(\boldsymbol{c}\boldsymbol{c}^\dagger\boldsymbol{c}) \leqslant \frac{N}{2}[\langle k^{\text{in}2}\rangle + \langle k^{\text{out}2}\rangle], \quad \text{Tr}(\boldsymbol{c}^2) \leqslant N\langle k\rangle, \quad (12)$$

$$\text{Tr}(\boldsymbol{c}\boldsymbol{c}^\dagger\boldsymbol{c}\boldsymbol{c}^\dagger) \leqslant \sum_{ij} k_i^{\text{in}}c_{ij}k_i^{\text{out}}, \quad \text{Tr}(\boldsymbol{c}^3) \leqslant N\langle k^{\text{in}}k^{\text{out}}\rangle. \quad (13)$$

An upper bound on $\sum_{ij} k_i^{\rm in} c_{ij} k_j^{\rm out}$ follows from the observation that if $c_{ij} = 1$, then certainly $k_i^{\rm in} \geqslant 1$ and $k_j^{\rm out} \geqslant 1$. Hence,

$$\sum_{ij} k_i^{\rm in} c_{ij} k_j^{\rm out} \geqslant \frac{1}{2} \sum_{ij} \left[c_{ij} k_j^{\rm out} + k_i^{\rm in} c_{ij} \right] = \frac{1}{2} N [\langle k^{\rm in2} \rangle + \langle k^{\rm out2} \rangle]. \tag{14}$$

Combining (12)–(14) with (11) then gives

$$n(\boldsymbol{c}) \leqslant \frac{N}{2} \left[N \langle k \rangle^2 + \langle k \rangle + \frac{1}{2} (\langle k^{\rm in2} \rangle + \langle k^{\rm out2} \rangle) - \frac{4}{3} \langle k^{\rm in} k^{\rm out} \rangle \right]. \tag{15}$$

Next we calculate a lower bound for $n(\boldsymbol{c})$. We use simple identities such as

$$\mathrm{Tr}(\boldsymbol{c}^2) \geqslant 0, \quad n_{\triangle}(\boldsymbol{c}) \geqslant 0, \quad \mathrm{Tr}(\boldsymbol{c}\boldsymbol{c}^{\dagger}\boldsymbol{c}) \geqslant 0, \tag{16}$$

and

$$\mathrm{Tr}(\boldsymbol{c}\boldsymbol{c}^{\dagger}\boldsymbol{c}\boldsymbol{c}^{\dagger}) \geqslant \frac{1}{2} \sum_{ijk\ell} c_{ji} c_{jk} c_{\ell k} c_{\ell i} (\delta_{j\ell} + \delta_{ik}) = N [\langle k^{\rm in2} \rangle + \langle k^{\rm out2} \rangle]. \tag{17}$$

We now find

$$n(\boldsymbol{c}) \geqslant \frac{1}{2} N [\langle k^{\rm in2} \rangle + \langle k^{\rm out2} \rangle] + \frac{1}{2} N^2 \langle k \rangle^2 - \sum_j k_j^{\rm in} k_j^{\rm out} - \sum_{ij} k_i^{\rm in} c_{ij} k_j^{\rm out}. \tag{18}$$

We finally need an upper bound for $\sum_{ij} k_i^{\rm in} c_{ij} k_j^{\rm out}$, which we write in terms of $k_{\rm max}^{\rm in} = \max_i k_i^{\rm in}$ and $k_{\rm max}^{\rm out} = \max_i k_i^{\rm out}$:

$$\sum_{ij} k_i^{\rm in} c_{ij} k_j^{\rm out} \leqslant \frac{1}{2} \sum_{ij} \left[k_{\rm max}^{\rm in} c_{ij} k_j^{\rm out} + k_i^{\rm in} c_{ij} k_{\rm max}^{\rm out} \right] = \frac{1}{2} N \left[k_{\rm max}^{\rm in} \langle k^{\rm out2} \rangle + k_{\rm max}^{\rm out} \langle k^{\rm in2} \rangle \right]. \tag{19}$$

We thus obtain our lower bound for the mobility,

$$n(\boldsymbol{c}) \geqslant \frac{N}{2} \left[N \langle k \rangle^2 + \langle (k^{\rm in} - k^{\rm out})^2 \rangle - k_{\rm max}^{\rm in} \langle k^{\rm out2} \rangle - k_{\rm max}^{\rm out} \langle k^{\rm in2} \rangle \right]. \tag{20}$$

B. Identification of graph types most likely to be biased by accept-all edge swapping

We know from (5) that unbiased sampling of graphs, i.e., $p(\boldsymbol{c}) = 1/|\Omega|$ for all $\boldsymbol{c} \in \Omega$, requires using the following state-dependent acceptance probabilities in the edge-swap process:

$$A(\boldsymbol{c}|\boldsymbol{c}') = [1 + n(\boldsymbol{c})/n(\boldsymbol{c}')]^{-1}. \tag{21}$$

We now investigate under which conditions one will in large graphs effectively find $n(\boldsymbol{c})/n(\boldsymbol{c}') \to 1$ for all $\boldsymbol{c}, \boldsymbol{c}' \in \Omega$, so that the sampling bias would be immaterial. Let us define

$$\Delta n = \max_{\boldsymbol{c}, \boldsymbol{c}' \in \Omega} |n(\boldsymbol{c}) - n(\boldsymbol{c}')| = \max_{\boldsymbol{c} \in \Omega} n(\boldsymbol{c}) - \min_{\boldsymbol{c} \in \Omega} n(\boldsymbol{c}). \tag{22}$$

Using the two bounds (15) and (20), we immediately obtain

$$\begin{aligned} \Delta n &\leqslant \frac{N}{2} \left[\langle k \rangle - \frac{1}{2} (\langle k^{\rm in2} \rangle + \langle k^{\rm out2} \rangle) + \frac{2}{3} \langle k^{\rm in} k^{\rm out} \rangle + k_{\rm max}^{\rm in} \langle k^{\rm out2} \rangle + k_{\rm max}^{\rm out} \langle k^{\rm in2} \rangle \right] \\ &= \frac{N}{2} \left[\langle k \rangle - \frac{1}{6} (\langle k^{\rm in2} \rangle + \langle k^{\rm out2} \rangle) - \frac{1}{3} \langle (k^{\rm in} - k^{\rm out})^2 \rangle + k_{\rm max}^{\rm in} \langle k^{\rm out2} \rangle + k_{\rm max}^{\rm out} \langle k^{\rm in2} \rangle \right] \\ &\leqslant \frac{N}{2} \left[\langle k \rangle + k_{\rm max}^{\rm in} \langle k^{\rm out2} \rangle + k_{\rm max}^{\rm out} \langle k^{\rm in2} \rangle \right]. \end{aligned} \tag{23}$$

Clearly, $1 - \Delta n/n(\boldsymbol{c}) \leqslant n(\boldsymbol{c}')/n(\boldsymbol{c}) \leqslant 1 + \Delta n/n(\boldsymbol{c})$, so in view of (21), we are interested in the ratio $\Delta n/n(\boldsymbol{c})$, for which we find

$$\frac{\Delta n}{n(\boldsymbol{c})} \leqslant \frac{\langle k \rangle + k_{\rm max}^{\rm in} \langle k^{\rm out2} \rangle + k_{\rm max}^{\rm out} \langle k^{\rm in2} \rangle}{N \langle k \rangle^2 - k_{\rm max}^{\rm in} \langle k^{\rm out2} \rangle - k_{\rm max}^{\rm out} \langle k^{\rm in2} \rangle}. \tag{24}$$

So we can be confident that the impact of the graph mobility on the correct acceptance probabilities (21) is immaterial if

$$\frac{1}{\langle k \rangle} + \frac{2}{\langle k \rangle^2} \left(k_{\rm max}^{\rm in} \langle k^{\rm out2} \rangle + k_{\rm max}^{\rm out} \langle k^{\rm in2} \rangle \right) \ll N. \tag{25}$$

We see from this that we can apply the accept-all edge-swap process with confidence when we are working with a large network with a narrow degree distribution.

IV. MOBILITIES OF SIMPLE GRAPH EXAMPLES

In this section, we confirm the validity of the mobility formulas (9) and (10) for several simple examples of directed graphs.

(1) *Two isolated bonds*: Here we have $c_{12} = 1$, $c_{34} = 1$, and $c_{ij} = 0$ for all $(i,j) \notin \{(1,2),(3,4)\}$. It is immediately clear that $\boldsymbol{c}^{\updownarrow} = \boldsymbol{0}$, and

$$\sum_{ij} k_i^{\rm in} c_{ij} k_j^{\rm out} = 2, \quad \sum_j k_j^{\rm out} k_j^{\rm in} = 0, \quad \langle k \rangle = \frac{2}{N}, \quad \mathrm{Tr}(\boldsymbol{c}^2) = \mathrm{Tr}(\boldsymbol{c}^3) = \mathrm{Tr}(\boldsymbol{c}\boldsymbol{c}^{\dagger}\boldsymbol{c}) = 0, \quad \mathrm{Tr}(\boldsymbol{c}\boldsymbol{c}^{\dagger}\boldsymbol{c}\boldsymbol{c}^{\dagger}) = 2.$$

Insertion into (9) and (10) gives $n_{\square}(\boldsymbol{c}) = 1$ and $n_{\triangle}(\boldsymbol{c}) = 0$. As we would expect, only one (square) move is permitted.

(2) *Isolated triangle*: This example is defined by $c_{12} = c_{23} = c_{31} = 1$, with $c_{ij} = 0$ for all $(i,j) \notin \{(1,2),(2,3),(3,1)\}$. Again we have $\boldsymbol{c}^{\updownarrow} = \boldsymbol{0}$, but now

$$\sum_{ij} k_i^{\rm in} c_{ij} k_j^{\rm out} = 3, \quad \sum_j k_j^{\rm out} k_j^{\rm in} = 3, \quad \langle k \rangle = \frac{3}{N}, \quad \mathrm{Tr}(\boldsymbol{c}^2) = \mathrm{Tr}(\boldsymbol{c}\boldsymbol{c}^{\dagger}\boldsymbol{c}) = 0, \quad \mathrm{Tr}(\boldsymbol{c}^3) = 3, \quad \mathrm{Tr}(\boldsymbol{c}\boldsymbol{c}^{\dagger}\boldsymbol{c}\boldsymbol{c}^{\dagger}) = 3.$$

This results in $n_{\square}(\boldsymbol{c}) = 0$ and $n_{\triangle}(\boldsymbol{c}) = 1$. The only possible move is reversal of the directed triangle.

(3) *Complete (fully connected) graph*: Here $c_{ij} = 1 - \delta_{ij}$, and no edge swaps are possible. All nodes have $k_i^{\rm in} = k_i^{\rm out} = N - 1$, and since $\boldsymbol{c}^{\updownarrow} = \boldsymbol{c}$, we know that $n_{\triangle}(\boldsymbol{c}) = 0$. This connectivity matrix, also featured in [18], has eigenvalues $\lambda = N - 1$ (multiplicity 1) and $\lambda = -1$ (multiplicity $N - 1$).

Hence,

$$\sum_{ij} k_i^{\text{in}} c_{ij} k_j^{\text{out}} = N(N-1)^3, \quad \sum_j k_j^{\text{out}} k_j^{\text{in}} = N(N-1)^2,$$

$$\text{Tr}(\boldsymbol{c}^2) = \sum_i \lambda_i^2 = N(N-1),$$

$$\text{Tr}(\boldsymbol{c}\boldsymbol{c}^\dagger \boldsymbol{c}) = \text{Tr}(\boldsymbol{c}^3) = \sum_i \lambda_i^3 = N(N-1)(N-2),$$

$$\text{Tr}(\boldsymbol{c}\boldsymbol{c}^\dagger \boldsymbol{c}\boldsymbol{c}^\dagger) = \text{Tr}(\boldsymbol{c}^4) = \sum_i \lambda_i^4 = (N-1)[(N-1)^3+1].$$

Assembling the entire expression for the square mobility term (9) indeed gives the correct value $n_\square(\boldsymbol{c}) = 0$.

(4) *Directed spanning ring*: This directed graph, defined by $c_{ij} = \delta_{i+1,j}$ modulo N, gives a ring with a flow around it. We choose $N > 2$. Once more, $\boldsymbol{c}^\updownarrow = \boldsymbol{0}$, and we obtain for the relevant terms

$$\sum_{ij} k_i^{\text{in}} c_{ij} k_j^{\text{out}} = \sum_j k_j^{\text{out}} k_j^{\text{in}} = N, \quad \langle k \rangle = 1,$$

$$\text{Tr}(\boldsymbol{c}^2) = \text{Tr}(\boldsymbol{c}^3) = \text{Tr}(\boldsymbol{c}\boldsymbol{c}^\dagger \boldsymbol{c}) = 0, \quad \text{Tr}(\boldsymbol{c}\boldsymbol{c}^\dagger \boldsymbol{c}\boldsymbol{c}^\dagger) = N.$$

The final result, $n_\square(\boldsymbol{c}) = \frac{1}{2}N(N-3)$ and $n_\triangle(\boldsymbol{c}) = 0$, is again what we would expect. As soon as one first bond to participate in an edge swap is picked (for which there are N options), there are $N-3$ possibilities for the second (since the already picked bond and its neighbors are forbidden). The factor 2 corrects for overcounting.

(5) *Bidirectional spanning ring*: Our final example is the nondirected version of the previous graph, viz., $c_{ij} = \delta_{i,j-1} + \delta_{i,j+1}$ modulo N, with $N > 2$. Since $\boldsymbol{c}^\updownarrow = \boldsymbol{c}$, we have $n_\triangle(\boldsymbol{c}) = 0$. Now

$$\sum_{ij} k_i^{\text{in}} c_{ij} k_j^{\text{out}} = 8N, \quad \sum_j k_j^{\text{out}} k_j^{\text{in}} = 4N, \quad \langle k \rangle = 2,$$

$$\text{Tr}(\boldsymbol{c}^2) = 2N, \quad \text{Tr}(\boldsymbol{c}^3) = \text{Tr}(\boldsymbol{c}\boldsymbol{c}^\dagger \boldsymbol{c}) = 0,$$

$$\text{Tr}(\boldsymbol{c}\boldsymbol{c}^\dagger \boldsymbol{c}\boldsymbol{c}^\dagger) = 6N.$$

We thereby find $n_\square(\boldsymbol{c}) = 2N(N-4)$. This is double the mobility evaluated in [18], since every move in the undirected version of the network corresponds to two possible moves in the directed version of the network.

V. A RANDOMIZATION PROCESS WHICH PRESERVES DEGREES AND TARGETS DEGREE-DEGREE CORRELATIONS

So far we applied formula (5) for the canonical acceptance probabilities for directed graph edge swapping to the problem of generating graphs with prescribed in and out degrees $(\boldsymbol{k}^{\text{in}}, \boldsymbol{k}^{\text{out}})$ and a uniform measure. Here we consider how to generate graphs which, in addition, display certain degree correlations. We first rewrite (5) as

$$A(\boldsymbol{c}|\boldsymbol{c}') = \left[1 + \frac{n(\boldsymbol{c})}{n(\boldsymbol{c}')} e^{H(\boldsymbol{c})-H(\boldsymbol{c}')}\right]^{-1}. \tag{26}$$

These probabilities (26) ensure that the edge-swapping process evolves into the stationary state on $\Omega = \{\boldsymbol{c} \in \{0,1\}^{N^2} | \boldsymbol{k}^{\text{in}}(\boldsymbol{c}) = \boldsymbol{k}^{\text{in}}, \boldsymbol{k}^{\text{out}}(\boldsymbol{c}) = \boldsymbol{k}^{\text{out}}\}$ defined by $p_\infty(\boldsymbol{c}) = Z^{-1}\exp[-H(\boldsymbol{c})]$. The full degree-degree correlation structure of a directed graph $\boldsymbol{c}$ is captured by the joint degree distribution of connected nodes,

$$W(\vec{k},\vec{k}'|\boldsymbol{c}) = \frac{1}{N\langle k \rangle} \sum_{ij} c_{ij} \delta_{\vec{k},\vec{k}_i(\boldsymbol{c})} \delta_{\vec{k}',\vec{k}_j(\boldsymbol{c})}, \tag{27}$$

with $\vec{k} = (k^{\text{in}}, k^{\text{out}})$. The maximum entropy distribution on Ω, viz., all directed graphs with prescribed in and out degree sequences, which has the distribution (27) imposed as a soft constraint, i.e., $\sum_{\boldsymbol{c}\in\Omega} p(\boldsymbol{c}) W(\vec{k},\vec{k}'|\boldsymbol{c}) = W(\vec{k},\vec{k}')$ for all $(\vec{k},\vec{k}')$, is

$$p(\boldsymbol{c}) = Z^{-1} \prod_i \delta_{\vec{k}_i,\vec{k}_i(\boldsymbol{c})} \prod_{ij} \left\{ \frac{\langle k \rangle}{N} Q(\vec{k}_i,\vec{k}_j) \delta_{c_{ij},1} + \left[1 - \frac{\langle k \rangle}{N} Q(\vec{k}_i,\vec{k}_j)\right] \delta_{c_{ij},0} \right\} \tag{28}$$

(see [22]), in which $Q(\vec{k},\vec{k}') = W(\vec{k},\vec{k}')/p(\vec{k})p(\vec{k}')$ and $p(\vec{k}) = p(k^{\text{in}},k^{\text{out}})$. It should be clear due to the context whether we are referring to $p(\boldsymbol{c})$ (the probability of observing a certain network $\boldsymbol{c}$ in the ensemble) or $p(\vec{k})$ (the probability of finding a node within a network with k^{in} incoming and k^{out} outgoing edges). It is now trivial, following [18], to ensure that our MCMC process evolves to the measure (28) by choosing $H(\boldsymbol{c}) = -\ln p(\boldsymbol{c})$ in the probabilities (26). This gives

$$A(\boldsymbol{c}|\boldsymbol{c}') = \left\{1 + \frac{n(\boldsymbol{c})}{n(\boldsymbol{c}')} \prod_{ij} \frac{\frac{\langle k \rangle}{N} Q(\vec{k}_i,\vec{k}_j) c'_{ij} + \left[1 - \frac{\langle k \rangle}{N} Q(\vec{k}_i,\vec{k}_j)\right](1-c'_{ij})}{\frac{\langle k \rangle}{N} Q(\vec{k}_i,\vec{k}_j) c_{ij} + \left[1 - \frac{\langle k \rangle}{N} Q(\vec{k}_i,\vec{k}_j)\right](1-c_{ij})}\right\}^{-1} = \left\{1 + \frac{n(\boldsymbol{c})}{n(\boldsymbol{c}')} \prod_{ij} \left[\frac{\frac{\langle k \rangle}{N} Q(\vec{k}_i,\vec{k}_j)}{1 - \frac{\langle k \rangle}{N} Q(\vec{k}_i,\vec{k}_j)}\right]^{c'_{ij}-c_{ij}}\right\}^{-1}. \tag{29}$$

If the proposed move is a *square swap*, it is characterized by four distinct nodes (i,j,k,ℓ), and takes us from a graph $\boldsymbol{c}'$ with $c'_{ij}c'_{k\ell}\bar{c}'_{kj}\bar{c}'_{i\ell} = 1$ to a new graph $\boldsymbol{c}$ with $\bar{c}_{ij}\bar{c}_{k\ell}c_{kj}c_{i\ell} = 1$ (leaving all other N^2-4 bond variables unaffected). For such moves, the acceptance probabilities (29) become

$$A_\square(\boldsymbol{c}|\boldsymbol{c}') = \left\{1 + \frac{n(\boldsymbol{c})}{n(\boldsymbol{c}')} \frac{\left[\frac{N}{\langle k \rangle Q(\vec{k}_i,\vec{k}_j)} - 1\right]\left[\frac{N}{\langle k \rangle Q(\vec{k}_i,\vec{k}_\ell)} - 1\right]}{\left[\frac{N}{\langle k \rangle Q(\vec{k}_i,\vec{k}_j)} - 1\right]\left[\frac{N}{\langle k \rangle Q(\vec{k}_k,\vec{k}_\ell)} - 1\right]}\right\}^{-1}. \tag{30}$$

For large N, we may choose to approximate this by

$$A_\square(\boldsymbol{c}|\boldsymbol{c}') \approx \left[1 + \frac{n(\boldsymbol{c})}{n(\boldsymbol{c}')} \frac{Q(\vec{k}_i,\vec{k}_j) Q(\vec{k}_k,\vec{k}_\ell)}{Q(\vec{k}_k,\vec{k}_j) Q(\vec{k}_i,\vec{k}_\ell)}\right]^{-1}. \tag{31}$$

If the proposed move is a *triangle edge swap*, it is characterized by three distinct nodes (i,j,k), and takes us from a graph $\boldsymbol{c}'$ with $c'_{ij}c'_{jk}c'_{ki}\bar{c}'_{ji}\bar{c}'_{kj}\bar{c}'_{ik} = 1$ to a new graph $\boldsymbol{c}$ with $\bar{c}_{ij}\bar{c}_{jk}\bar{c}_{ki}c_{ji}c_{kj}c_{ik} = 1$ (leaving all other N^2-6 bond variables unaffected). Now the acceptanceprobabilities (29)

become

$$A_{\triangle}(\boldsymbol{c}|\boldsymbol{c}') = \left\{1 + \frac{n(\boldsymbol{c})}{n(\boldsymbol{c}')} \frac{\left[\frac{N}{\langle k\rangle Q(\vec{k}_j,\vec{k}_i)} - 1\right]\left[\frac{N}{\langle k\rangle Q(\vec{k}_k,\vec{k}_j)} - 1\right]\left[\frac{N}{\langle k\rangle Q(\vec{k}_i,\vec{k}_k)} - 1\right]}{\left[\frac{N}{\langle k\rangle Q(\vec{k}_i,\vec{k}_j)} - 1\right]\left[\frac{N}{\langle k\rangle Q(\vec{k}_j,\vec{k}_k)} - 1\right]\left[\frac{N}{\langle k\rangle Q(\vec{k}_k,\vec{k}_i)} - 1\right]}\right\}^{-1}. \quad (32)$$

For large N, we may choose to approximate this by

$$A_{\triangle}(\boldsymbol{c}|\boldsymbol{c}') = \left[1 + \frac{n(\boldsymbol{c})}{n(\boldsymbol{c}')} \frac{Q(\vec{k}_i,\vec{k}_j)Q(\vec{k}_j,\vec{k}_k)Q(\vec{k}_k,\vec{k}_i)}{Q(\vec{k}_j,\vec{k}_i)Q(\vec{k}_k,\vec{k}_j)Q(\vec{k}_i,\vec{k}_k)}\right]^{-1}. \quad (33)$$

VI. NUMERICAL SIMULATIONS OF THE CANONICAL RANDOMIZATION PROCESS

In this section, we describe numerical simulations of our canonical MCMC graph randomization process and its accept-all edge-swapping counterpart, applied to synthetic networks and to biological signaling networks. We use the Mersenne twister random-number generator from [23]. For numerical implementation, we use expressions for the incremental change in the mobility terms following a single edge-swap move (similar to how this was done for nondirected networks [18])—see the Appendix. This avoids having to calculate $n(\boldsymbol{c})$ after each move, which would involve repeated matrix multiplications. We find that the increase in running time relative to the naive approach using our adjusted acceptance probability is not material. For example, in the real biological examples shown in Figs. 8 and 9, both approaches equilibrate within minutes using a desktop personal computer. The difference in running times between the two is less than 10%. Targeting a specific degree-degree correlation does carry a significant penalty in terms of computer time due to the need to calculate a more complicated acceptance probability at every step. However, it is likely that the implementation could be substantially sped up and optimized if required for a real application. Full source code (in C++) and Windows executables of our implementation are available on request.

We find that the most convenient marker of sampling bias in randomization is the mobility $n(\boldsymbol{c})$ itself, which we will therefore use to monitor the dynamics of the process. For the synthetic networks discussed in Secs. VI A and VI B, we can calculate the mobility for each type of network, hence we can directly relate it to the proportion of time that the process actually spends in each configuration versus the expected proportion of time for an unbiased process. For the real biological networks, since our postulate is that the "biased" process favors networks with higher mobilities, it seems reasonable to assume that the (running) average mobility over the course of the process will be the statistic that most clearly illustrates the difference between the two properties.

A. "Split-flow" network

A "split-flow" network (see, e.g., [19]) is built as follows. Node $i = 1$ has degrees $(k_1^{\text{in}},k_1^{\text{out}}) = (0,K)$; we have K nodes $(i = 2,\ldots,K+1)$ with degrees $(k_i^{\text{in}},k_i^{\text{out}}) = (1,1)$, and a final node with degrees $(k_{K+2}^{\text{in}},k_{K+2}^{\text{out}}) = (K,0)$. There exist two types of graphs with this specified degree sequence. The first is shown in the left of Fig. 3. The second type is obtained from the first by choosing two of the K "inner nodes", of which one will cease to receive a link from $i = 1$ and the second will cease to provide a link to $i = K+2$; so the mobility of the left graph is $n(\boldsymbol{c}) = K(K-1)$. On the right-hand-side configurations in Fig. 3, we can execute three possible square edge-swap types: returning to the previous state (1 such move), changing the internal node that is not receiving a link from $i = 1$ $(K-2$ such moves), or changing the internal node that is not sending a link to $i = K+2$ $(K-2$ such moves), giving a total mobility for the graphs on the right of $n(\boldsymbol{c}) = 2K-3$. The total number of such split-flow networks is $|\Omega| = K(K-1)+1$.

Figure 4 shows graph randomization dynamics for a split-flow network with $K = 25$, comparing accept-all edge swapping [which would sample graphs with the bias $p(\boldsymbol{c}) = n(\boldsymbol{c})/\sum_{\boldsymbol{c}'\in\Omega} n(\boldsymbol{c}')$] to the canonical edge-swap process (21) that is predicted to give unbiased sampling of graphs, $p(\boldsymbol{c}) = 1/|\Omega|$. The predicted expectation values of the mobilities in the two sampling protocols are

$$\text{accept all:}\quad \langle n(\mathbf{c})\rangle = \frac{\sum_{\mathbf{c}\in\Omega} n^2(\mathbf{c})}{\sum_{\mathbf{c}\in\Omega} n(\mathbf{c})} = \frac{5K^2-13K+9}{2(K-1)} \approx 58.52,$$

$$\text{canonical:}\quad \langle n(\mathbf{c})\rangle = \frac{\sum_{\mathbf{c}\in\Omega} n(\mathbf{c})}{|\Omega|} = \frac{2K(K-1)^2}{1+K(K-1)} \approx 47.92.$$

The simulation results confirm these quantitative predictions (see caption of Fig. 4 for details) and underlines the sampling bias caused by accept-all edge swapping, as well as the lack of such a bias in our canonical MCMC process.

B. "Nearly hardcore" networks

"Nearly hardcore" networks are another example of graphs for which accept-all edge-swap sampling is known to exhibit a significant bias [18]. The directed version of such networks

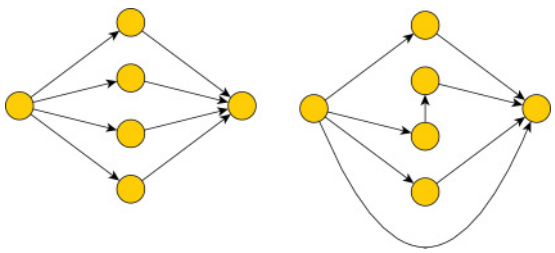

FIG. 3. (Color online) The possible realizations of a split-flow-type network, with $N = K+2$. The left-hand configuration has a mobility of $K(K-1)$; there is only 1 such configuration. The right-hand configuration has mobility of $2K-3$; there are $K(K-1)$ such configurations.

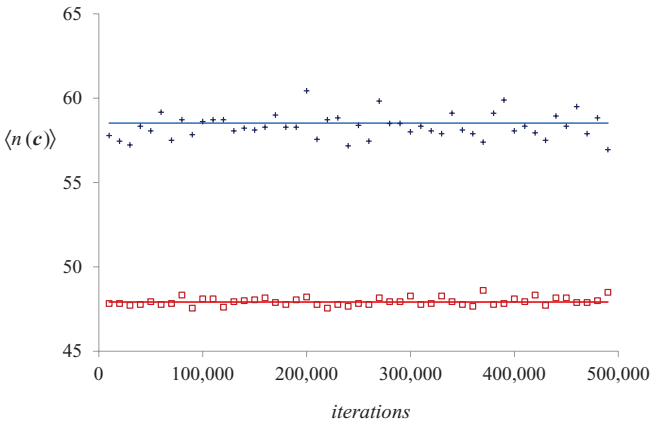

FIG. 4. (Color online) Comparison for split-flow networks with $K = 25$ of randomization via accept-all edge swapping (squares) vs edge swapping with the canonical acceptance probabilities (crosses). The mobility $\langle n(\boldsymbol{c})\rangle$ is used as a dynamical observable since its expectation value is sensitive to sampling bias. Each marker gives the average mobility over 10 000 iterations. Observed values are in good agreement with theoretical predictions: $\langle n(\boldsymbol{c})\rangle \approx 58.32$ for accept-all edge swapping (predicted: 58.52, shown by upper solid line) vs $\langle n(\boldsymbol{c})\rangle \approx 47.95$ for correct edge swapping (predicted: 47.92, shown by lower solid line).

is constructed from a single isolated bond plus a complete subgraph of size $K = N - 2$; see Fig. 5. Triangle swaps are not possible. From the graph shown in the figure (the "mobile" state, A), there are $K(K-1)$ ways to choose two nodes of the core to combine with the two noncore nodes to form an edge-swap quartet, hence this state has $n_A(\boldsymbol{c}) = K(K-1)$. After an edge swap, the graph in Fig. 5 is replaced by one in which one noncore node receives a link from the core,

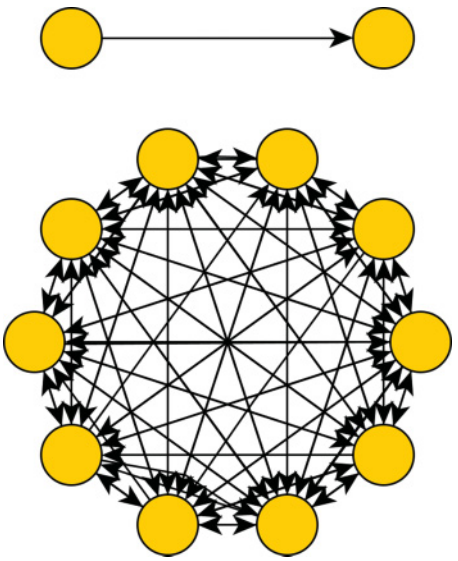

FIG. 5. (Color online) The directed version of a "nearly hardcore" network. Given the imposed degree sequences, there are only two types of graphs: the one shown here and the one obtained via an edge swap that involves the nodes of the isolated link and two nodes from the core.

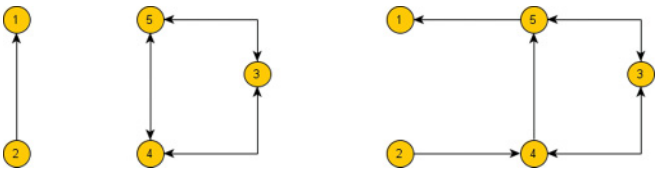

FIG. 6. (Color online) Illustration of the edge swap that transforms a nearly hardcore graph from state A to one of the type B states.

and the other sends a link to the core; see Fig. 6. There are $K(K-1)$ such graphs, to be called type B, hence the total number of nearly hardcore graphs is $|\Omega| = K(K-1)+1$. From each type B graph, the inverse swap can be applied, plus $2(K-2)$ further moves that each equate to the replacement of one of the core nodes involved in the previous swap by another. Hence, $n_B(\boldsymbol{c}) = 2K-3$. These statements are confirmed by formula (9).

The predicted expectation values of the mobilities in the two sampling protocols, namely, accept-all edge swapping [which would sample graphs with the bias $p(\boldsymbol{c}) = n(\boldsymbol{c})/\sum_{\boldsymbol{c}'\in\Omega} n(\boldsymbol{c}')$] and the canonical edge-swap process (21) [predicted to give unbiased sampling of graphs $p(\boldsymbol{c}) = 1/|\Omega|$] are

$$\text{accept all:}\quad \langle n(\boldsymbol{c})\rangle = \frac{n_A^2(\boldsymbol{c}) + K(K-1)n_B^2(\boldsymbol{c})}{n_A(\boldsymbol{c}) + K(K-1)n_B(\boldsymbol{c})} = \frac{5K^2 - 13K + 9}{2(K-1)},$$

$$\text{canonical:}\quad \langle n(\boldsymbol{c})\rangle = \frac{n_A(\boldsymbol{c}) + K(K-1)n_B(\boldsymbol{c})}{1 + K(K-1)} = \frac{2K(K-1)^2}{1 + K(K-1)}.$$

Figure 7 shows graph randomization dynamics for a nearly hardcore network with $K = 18$ (so $N = 20$). Here the theory,

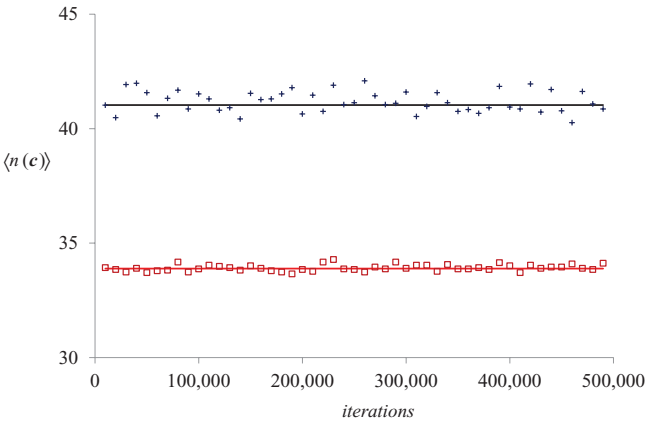

FIG. 7. (Color online) Comparison for nearly hardcore networks with $K = 18$ of randomization via accept-all edge swapping (squares) vs edge swapping with the canonical acceptance probabilities (crosses). Each marker gives the average mobility over 10 000 iterations. Observed mobility values are again in good agreement with theoretical predictions: $\langle n(\boldsymbol{c})\rangle \approx 41.09$ for accept all (predicted: 41.03, shown by upper solid line) vs $\langle n(\boldsymbol{c})\rangle \approx 33.92$ for correct edge swapping (predicted: 33.89, shown by lower solid line).

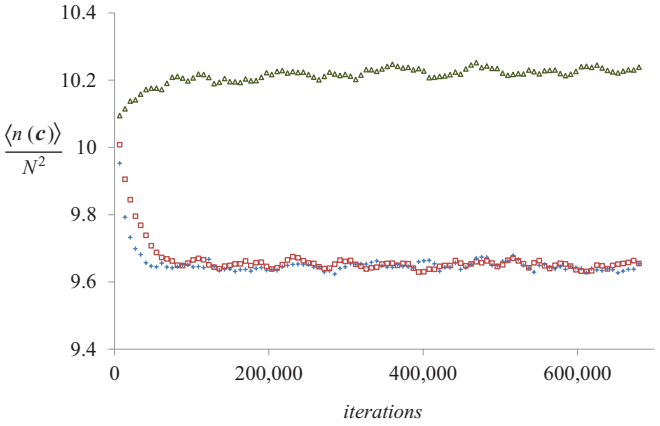

FIG. 8. (Color online) Randomization dynamics for the gene regulation network data of [24]. The observable shown is a running average of the normalized average square mobility $\langle n_{\square}(\boldsymbol{c})\rangle/N^2$. We compare accept-all edge swapping (+), canonical edge swapping aimed at uniform sampling of all graphs with the biological degree sequence of the biological network (□), and canonical edge swapping aimed at uniform sampling of all graphs with the degree sequence $(\vec{k}_1,\ldots,\vec{k}_N)$ *and* the degree-degree correlation kernel $W(\vec{k},\vec{k}')$ of the biological network (△). To demonstrate that the above processes are effectively shuffling the networks, we measure the Hamming distance between the start and end networks. The Hamming distance is defined as $\frac{1}{2E}\sum_{ij}|c_{ij}^{\rm start}-c_{ij}^{\rm end}|$, where E is the total number of edges. A value of zero would indicate that the start and end networks were identical. A value of 1 would indicate that the start and end networks had no edges in common. Hamming distances between the start and end networks of □, +, and △ were 0.8, 0.8, and 0.75, respectively.

i.e., the previous two formulas, predicts that we should see $\langle n(\boldsymbol{c})\rangle \approx 41.03$ for accept-all edge swapping, and $\langle n(\boldsymbol{c})\rangle \approx 33.89$ for unbiased sampling. Again the simulation results confirm our predictions (see caption of Fig. 7 for details).

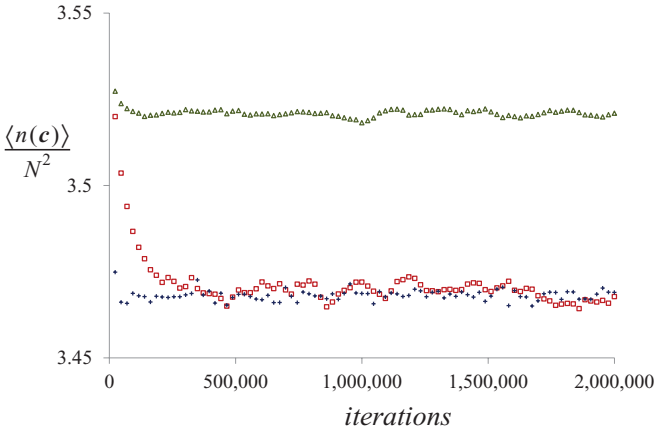

FIG. 9. (Color online) Randomization dynamics for the gene regulation network of [25]. The key and the axes are the same as in Fig. 8. Hamming distances between the start and end networks of □, +, and △ were 0.94, 0.94, and 0.86, respectively. Please also see Fig. 10 for a demonstration of how the △ process effectively targets the degree-degree correlation of the original network.

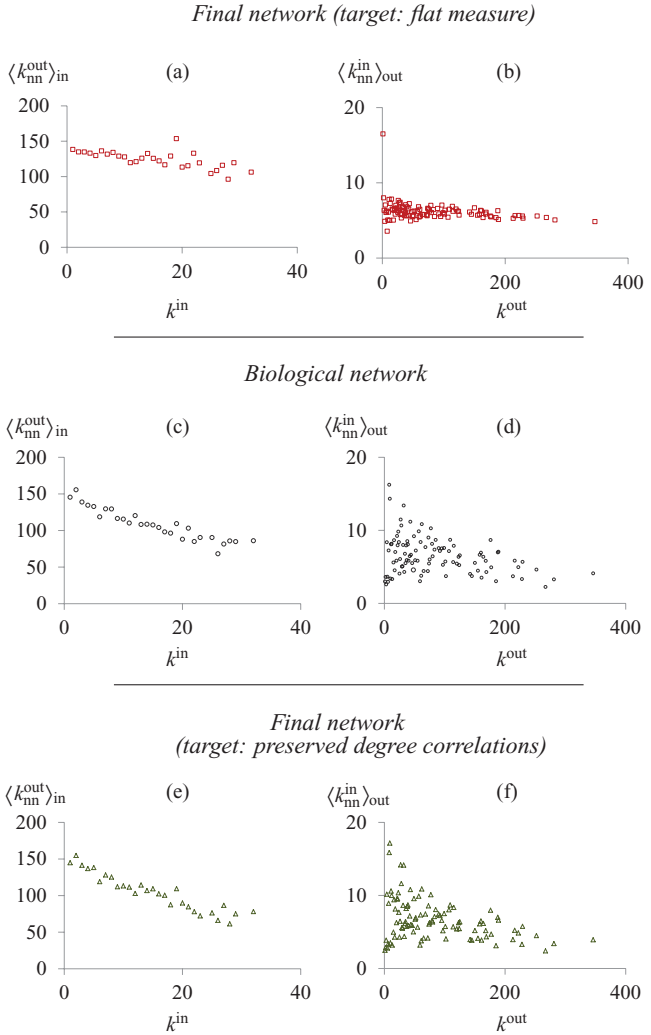

FIG. 10. (Color online) These charts summarize the (c), (d) degree-degree correlations observed in the original network, (a), (b) the final network after the process targeting the flat measure, and (e), (f) the process tailored to preserve the degree-degree correlation pattern of the original network. The data used is based on [25] and the process shown in Fig. 9. (a), (c), and (e) summarize the correlation between the in degree of a node and the average out degree $\langle k_{\rm nn}^{\rm out}\rangle_{\rm in}$ of its in neighbors. (b), (d), and (f) summarize the correlation between the out degree of a node and the average in degree $\langle k_{\rm nn}^{\rm in}\rangle_{\rm out}$ of its out neighbors. This representation was chosen as a widely adopted and easy-to-interpret measure of the assortativity of a directed network. The purpose of these charts is to demonstrate that the process targeting degree-degree correlations successfully reproduces the key features of the assortativity of the real networks. In particular, the pronounced downward slope in the last four charts brings to mind the work of [12] on the related problem of protein interaction networks. Reference [12] also observed a characteristic downward slope to the assortativity charts from their data, and postulated that this may be a key "design" feature of real networks, contributing greater stability and improved specificity.

C. Application to gene regulation networks

Gene regulation networks are important examples of directed biological networks. Figures 8 and 9 show numerical results of the randomization dynamics applied to the gene

regulation network data of [24] (with $N = 5654$ nodes) and [25] (with $N = 3865$ nodes), respectively. We apply all three randomization processes discussed so far in this paper, viz., accept-all edge swapping, canonical edge swapping aimed at uniform sampling of all graphs with the degree sequences of the biological network, and canonical edge swapping aimed at uniform sampling of all graphs with the degree sequence $(\vec{k}_1, \ldots, \vec{k}_N)$ *and* (on average) the degree-degree correlation kernel $W(\vec{k},\vec{k}')$ of the biological network.

In contrast to the synthetic examples in the previous section, in gene regulation networks we do not observe significant divergence between accept-all versus canonical edge-swap randomization; this is similar to what was observed earlier for the randomization of protein-protein interaction networks in [18]. We also see that in both cases, the biological network is significantly more mobile than the typical network with the same degree sequence. However, Figs. 8 and 9 suggests that the set of networks that share both the degree sequence *and* the degree correlations with the biological one (and hence resemble more closely the biological network under study) all have high mobilities.

The implementation of degree-degree correlation targeting directly has the effect of severely reducing the space of graphs through which the process can pass, hence we would expect finite-size effects to be more pronounced. The process would be less restricted, and hence more natural, with a smoothed target degree-degree correlation. There is a trade-off between the flexibility of the process and the accuracy of the targeting. We have used a light Gaussian smoothing, generalizing what was used in [26] to the higher dimension we need. The best choice target degree-degree correlation—including decisions about smoothing—will very much depend on the particular problem being studied.

VII. CONCLUSION

In this paper, we have built on the work of [3] and [18] to define an ergodic and unbiased stochastic process for randomizing directed binary non-self-interacting networks, which keeps the number of in and out connections of each node constant. The result takes the form of a canonical Markov chain Monte Carlo (MCMC) algorithm based on simple directed edge swaps and triangle reversals, with nontrivial move acceptance probabilities that are calculated from the current state of the network only. The acceptance probabilities correct for the entropic bias in accept-all edge-swap randomization, which is caused by the state dependence of the number of moves that can be executed (the mobility of a graph).

Our process is precise for any network size and network topology, and sufficiently versatile to allow random directed graphs with the correct in and out degree sequence to be generated with arbitrary desired sampling probabilities. The algorithm can be used, e.g., to generate truly unbiased random directed graphs with imposed degrees for hypothesis testing (in contrast to the "edge stub" algorithm or the accept-all edge-swap algorithm, both of which are biased) or to generate more sophisticated null models, which inherit from a real network both the degree sequence *and* the degree correlations but are otherwise random and unbiased.

Our core insight is similar to [27] and [3]. However, our work takes the formalism further, and generates a direct adjustment to the MCMC based on the current state of the network only, rather than a retrospective adjustment to the observed process [27] or a search of the entire state space [3]. Moreover, our approach can be generalized to generate more tailored null models (e.g., our example of targeting a specified degree-degree correlation).

Our earlier papers, i.e., [28] (undirected case) and [22] (directed case), also consider constrained ensembles of random graphs, but from the point of view of rigorously quantifying the entropy of the ensembles. These papers may be of interest to researchers looking to study, quantify, and compare the topological properties of their networks in a more information-theoretic framework. The prequel to the current paper [18] would be of interest to researchers doing numerical studies on undirected networks.

We have derived bounds to predict for which degree sequences the differences between accept-all and correct randomization (i.e., the effects of sampling bias) will be negligible. The application to synthetic networks showed a large discrepancy between the accept-all and correct randomization processes, and good agreement with our theoretical predictions for the values of key observables that are affected by the entropic bias of incorrect randomization. For the biological networks which we studied (gene regulation networks), we find the differences between correct and incorrect sampling in the space of graphs with imposed degree sequences to be negligible. However, this cannot be relied upon to continue in future studies, especially when network data sets become less sparse, or randomization processes which target more complicated topological observables are used. Since our method provably samples graphs uniformly, it can be used to test whether, for a given real network, the naive edge swapping and the correct approach deviate materially.

Biological signaling networks tend to have "fat-tailed" distributions with low average degree and relatively high clustering levels, whereas in a graph ensemble defined by prescribing in and out degree sequences and uniform graph probabilities, graphs will typically have $O(1)$ triangles per node or less. Hence, if we run edge-swap processes on such ensembles, by the time equilibration is approached, the algorithm will typically be moving through networks with low clustering, where the change in mobility coming from those terms that "count" triangles will be very low. However, this will be different if we target a nonflat measure, for instance if we generate graphs with degree-degree correlations. Since biological degree-degree correlations seem to be associated with clustering, it will become increasingly dangerous to assume that the sampling bias caused by using accept-all edge-swap dynamics will be modest.

Given that precise and practical alternatives are now available, we feel that there is no justification for the use of biased graph randomization processes. In those cases where we seek to generate unbiased random directed graphs with in and out degrees identical to some observed network, our canonical MCMC process would take the observed graph as its seed and take care of the required unbiased sampling. In those cases where we specify degree sequences *ab initio*, without having a seed graph, one may use the Molloy-Reed

algorithm to generate a (biased) seed graph prior to running our algorithm.

In addition to being rigorously free of entropic sampling bias, our present canonical MCMC process is also able to generate directed degree-constrained networks with any arbitrary specified sampling probabilities. We have shown examples of the generation of synthetic graphs with precisely controlled expectation values for the degree-degree correlation kernels, where the imposed sampling measure is a maximum entropy distribution on the set of graphs with prescribed degrees, with degree correlations imposed as a soft constraint. Degree correlation is a promising candidate to define a better null model, as it has been observed in the literature to act as a "signature" distinguishing different types of networks (e.g., [29,30]).

The exact connection which we established between the MCMC parameters and the resulting equilibrium distribution means that our approach can be used to define an edge-swapping MCMC process which is directly related to dynamical processes defined by others (e.g., it should be possible to define a process which is equivalent to taking degree distribution as a soft constraint, such as considered in [6]). This could act as a bridge between different methods of analyzing network observables in random graph ensembles.

Other directions for future research could be to look at weighted networks (e.g., to integrate our ideas with those in papers such as [31]) or at bipartite networks (which also have interesting applications; see, e.g., [32]). Furthermore, it would seem appropriate in the field of network hypothesis testing to take more seriously the nontrivial number of short loops in biological signaling systems. Whenever we randomize within the large amorphous space of graphs that inherit from the biological network only the degree sequence, we are effectively running dynamics on graphs that are locally treelike, where (conveniently) the mobility issues are minor. But we know already that this large set will typically produce null models that are very much unlike biological networks for that same reason. How informative are small p values in this context?

ACKNOWLEDGMENTS

This study was supported by the Biotechnology and Biological Sciences Research Council of the United Kingdom. It is our pleasure to thank Thomas Schlitt for providing gene regulation network data. We are grateful to the anonymous referees for their constructive and encouraging comments.

APPENDIX: EFFICIENT CALCULATION OF CHANGES IN MOBILITY TERMS FOLLOWING ONE MOVE

The calculation of the mobility $n(\boldsymbol{c})$ terms is computationally heavy. Given that our moves are simple and standard, we follow the alternative route in [18] and derive formulas for calculating the *change* in mobility due to one move so that we can avoid repeated heavy matrix multiplications at each time step.

1. Change in $n_{\square}(\boldsymbol{c})$ following one square-type move

Without loss of generality, define our square move to be the transformation between matrix c and x, involving four nodes (a,b,c,d), such that, for all (i,j), $x_{ij} = c_{ij} + \Delta_{ij}$, with

$$\Delta_{ij} = \delta_{ia}\delta_{jd} + \delta_{ic}\delta_{jb} - \delta_{ia}\delta_{jb} - \delta_{ic}\delta_{jd}.$$

We now determine the overall change induced in $n_{\square}(\boldsymbol{c})$ by finding the impact of an edge swap on each term in (9) on the right-hand side of the expression above.

(a) Term 1:

$$\begin{aligned}
&\mathrm{Tr}(\boldsymbol{x}\boldsymbol{x}^{\dagger}\boldsymbol{x}\boldsymbol{x}^{\dagger}) - \mathrm{Tr}(\boldsymbol{c}\boldsymbol{c}^{\dagger}\boldsymbol{c}\boldsymbol{c}^{\dagger}) \\
&\quad = \sum_{ijkm}[c_{ij}c_{kj}c_{km}c_{im} - (c_{ij}+\Delta_{ij})(c_{kj}+\Delta_{kj}) \\
&\qquad \times (c_{km}+\Delta_{km})(c_{im}+\Delta_{im})] \\
&\quad = \Delta_{ij}c_{kj}c_{km}c_{im} + \cdots + \Delta_{ij}\Delta_{kj}c_{km}c_{im} + \cdots \\
&\qquad + \Delta_{ij}\Delta_{kj}\Delta_{km}c_{im} + \cdots + \Delta_{ij}\Delta_{kj}\Delta_{km}\Delta_{im},
\end{aligned}$$

where $\cdots$ refers in each case to three similar terms (with their appropriate indices). Let us inspect what happens when two Δ terms are multiplied together. We might have the first suffix repeated, the second suffix repeated, or no repeated suffixes:

$$\begin{aligned}
\Delta_{ij}\Delta_{im} &= 2[\delta_{jd}(\delta_{md}-\delta_{mb}) + \delta_{jb}(\delta_{mb}-\delta_{md})], \\
\Delta_{ij}\Delta_{kj} &= 2[\delta_{ia}(\delta_{ka}-\delta_{kc}) + \delta_{ic}(\delta_{kc}-\delta_{ka})].
\end{aligned} \tag{A1}$$

One immediately observes that

$$\begin{aligned}
\sum_{ijkm}\Delta_{ij}\Delta_{kj}\Delta_{km}\Delta_{im} = 4\sum_{ik}[&\delta_{ia}\delta_{ia}(\delta_{ka}\delta_{ka}+\delta_{kc}\delta_{kc}) \\
&+ \delta_{ic}\delta_{ic}(\delta_{kc}\delta_{kc}+\delta_{ka}\delta_{ka})] = 16.
\end{aligned}$$

To handle two Δ terms with different suffixes, we use

$$\Delta_{ij}c_{kj} = c_{kb}\,(\delta_{ic}-\delta_{ia}) + c_{kd}\,(\delta_{ia}-\delta_{ic})\,, \tag{A2}$$

which leads us to

$$\sum_{ijkm}\Delta_{ij}c_{kj}\Delta_{km}c_{im} = 4.$$

Returning to the result (A1), it follows that

$$\begin{aligned}
\Delta_{ij}\Delta_{kj}c_{im}c_{km} &= 2\,[\delta_{ia}(\delta_{ka}-\delta_{kc}) + \delta_{ic}(\delta_{kc}-\delta_{ka})]\,c_{im}c_{km} \\
&= 2\left(k_a^{\mathrm{in}} + k_c^{\mathrm{in}}\right) - 4c_{am}c_{cm},
\end{aligned}$$

and the symmetric term gives

$$\Delta_{ij}\Delta_{im}c_{kj}c_{km} = 2\left(k_d^{\mathrm{out}} + k_b^{\mathrm{out}}\right) - 4c_{id}c_{ib}.$$

For the third-order terms, we combine (A1) and (A2):

$$\begin{aligned}
&\sum_{ijkm}\Delta_{ij}\Delta_{im}\Delta_{kj}c_{km} \\
&\quad = 2\sum_{ik}\{[\delta_{ia}(\delta_{ka}-\delta_{kc}) + \delta_{ic}(\delta_{kc}-\delta_{ka})] \\
&\qquad \times [\delta_{ia}c_{kd} + \delta_{ic}c_{kb} - \delta_{ia}c_{kb} - \delta_{ic}c_{kd}]\} \\
&\quad = 2(c_{ad} - c_{cd} - c_{ab} + c_{cb} - c_{ab} + c_{cb} + c_{ad} - c_{cd}) = -8.
\end{aligned}$$

By permutation of suffixes, all such terms evaluate to -8. Finally, we turn to the four terms where only one Δ appears, corresponding to permutations of $\Delta_{ij}c_{kj}c_{km}c_{im} = c_{kd}c_{km}c_{am} + c_{kb}c_{km}c_{cm} - c_{kb}c_{km}c_{am} - c_{kd}c_{km}c_{cm}$. Adding up

all of the separate elements above, we obtain the change in the square mobility term due to one application of a square move:

$$\begin{aligned}\Delta\left[\tfrac{1}{2}\mathrm{Tr}(\boldsymbol{c}\boldsymbol{c}^{\dagger}\boldsymbol{c}\boldsymbol{c}^{\dagger})\right] &= 2\left(k_d^{\mathrm{out}}+k_b^{\mathrm{out}}+k_a^{\mathrm{in}}+k_c^{\mathrm{in}}\right)+2(c_{kd}c_{km}c_{am}\\ &\quad +c_{kb}c_{km}c_{cm}-c_{kb}c_{km}c_{am}-c_{kd}c_{km}c_{cm})\\ &\quad -4(c_{id}c_{ib}+c_{am}c_{cm}+1).\end{aligned} \tag{A3}$$

(b) Term 2:

$$\begin{aligned}\Delta\left[\sum_{ij}k_i^{\mathrm{in}}c_{ij}k_j^{\mathrm{out}}\right] &= \sum_{ij}k_i^{\mathrm{in}}(x_{ij}-c_{ij})k_j^{\mathrm{out}}\\ &= \sum_{ij}k_i^{\mathrm{in}}[\delta_{ia}\delta_{jd}+\delta_{ic}\delta_{jb}\\ &\quad -\delta_{ia}\delta_{jb}-\delta_{ic}\delta_{jd}]k_j^{\mathrm{out}}\\ &= k_a^{\mathrm{in}}k_d^{\mathrm{out}}+k_c^{\mathrm{in}}k_b^{\mathrm{out}}-k_a^{\mathrm{in}}k_b^{\mathrm{out}}-k_c^{\mathrm{in}}k_d^{\mathrm{out}}.\end{aligned} \tag{A4}$$

(c) Term 3:

$$\begin{aligned}&\mathrm{Tr}(\mathbf{x}\mathbf{x}^{\dagger}\mathbf{x})-\mathrm{Tr}(\boldsymbol{c}\boldsymbol{c}^{\dagger}\boldsymbol{c})\\ &= \sum_{ijk}[c_{ij}+\Delta_{ij}][c_{kj}+\Delta_{kj}][c_{ki}+\Delta_{ki}]-c_{ij}c_{kj}c_{ki}\\ &= \sum_{ijk}[\Delta_{kj}c_{ij}c_{ki}+\Delta_{ij}c_{kj}c_{ki}+\Delta_{ki}c_{ij}c_{kj}+\Delta_{ij}\Delta_{kj}c_{ki}\\ &\quad +\Delta_{kj}\Delta_{ki}c_{ij}+\Delta_{ij}\Delta_{ki}c_{kj}+\Delta_{ij}\Delta_{kj}\Delta_{ki}].\end{aligned}$$

The product of two Δ terms gives

$$\begin{aligned}\Delta_{ij}\Delta_{kj} &= \delta_{ik}[\delta_{jd}(\delta_{ia}-\delta_{ic})+\delta_{jb}(\delta_{ic}-\delta_{ia})]\\ &\quad -\delta_{jd}(\delta_{ia}\delta_{kc}+\delta_{ic}\delta_{ka})-\delta_{jb}(\delta_{ic}\delta_{ka}+\delta_{ia}\delta_{kc}),\end{aligned}$$

but $\Delta_{ij}\Delta_{ki}=0$, and in a straightforward way, we obtain

$$\begin{aligned}\sum_{ijk}\Delta_{kj}c_{ij}c_{ki} &= \sum_{ijk}[\delta_{ka}\delta_{jd}+\delta_{kc}\delta_{jb}-\delta_{ka}\delta_{jb}-\delta_{kc}\delta_{jd}]c_{ij}c_{ki}\\ &= \sum_{i}[c_{ai}c_{id}+c_{ci}c_{ib}-c_{ai}c_{ib}-c_{ci}c_{id}].\end{aligned}$$

Assembling all terms and their symmetric equivalents leads to an expression which can be summarized as

$$\begin{aligned}\Delta[\mathrm{Tr}(\boldsymbol{c}\boldsymbol{c}^{\dagger}\boldsymbol{c})] &= MutN(a,d)+MutN(c,b)-MutN(a,b)\\ &\quad -MutN(c,d)-2(c_{bd}+c_{db}+c_{ac}+c_{ca}),\end{aligned} \tag{A5}$$

where

$$MutN(\alpha,\beta)=\sum_i[c_{\alpha i}c_{i\beta}+c_{\alpha i}c_{\beta i}+c_{i\alpha}c_{i\beta}]. \tag{A6}$$

(d) Term 5:

$$\begin{aligned}\Delta[\mathrm{Tr}(\boldsymbol{c}^2)] &= \mathrm{Tr}(\boldsymbol{x}^2)-\mathrm{Tr}(\boldsymbol{c}^2)\\ &= \sum_{ij}[c_{ij}+(\delta_{ia}\delta_{jd}+\delta_{ic}\delta_{jb}-\delta_{ia}\delta_{jb}-\delta_{ic}\delta_{jd})]\\ &\quad \times[c_{ji}+(\delta_{ja}\delta_{id}+\delta_{jc}\delta_{ib}-\delta_{ja}\delta_{ib}-\delta_{jc}\delta_{id})]\\ &\quad -\mathrm{Tr}(\boldsymbol{c}^2)\\ &= 2(c_{da}+c_{bc}-c_{ba}-c_{dc}).\end{aligned} \tag{A7}$$

(e) Terms 4 and 6: The two terms $\frac{1}{2}N^2\langle k\rangle^2$ and $\sum_i k_i^{\mathrm{out}}k_i^{\mathrm{in}}$ do not change, since our stochastic process conserves all degrees.

In combination, the above ingredients lead us to the following updated formula for the square mobility (9) as a result of the edge swap (A1):

$$\begin{aligned}\Delta n_{\square} &= 2\left(k_d^{\mathrm{out}}+k_b^{\mathrm{out}}+k_a^{\mathrm{in}}+k_c^{\mathrm{in}}\right)+2(c_{kd}c_{km}c_{am}+c_{kb}c_{km}c_{cm}\\ &\quad -c_{kb}c_{km}c_{am}-c_{kd}c_{km}c_{cm})-4(c_{id}c_{ib}+c_{am}c_{cm}+1)\\ &\quad -\left[k_a^{\mathrm{in}}k_d^{\mathrm{out}}+k_c^{\mathrm{in}}k_b^{\mathrm{out}}-k_a^{\mathrm{in}}k_b^{\mathrm{out}}-k_c^{\mathrm{in}}k_d^{\mathrm{out}}\right]\\ &\quad +MutN(a,d)+MutN(c,b)-MutN(a,b)\\ &\quad -MutN(c,d)-2(c_{bd}+c_{db}+c_{ac}+c_{ca})\\ &\quad +c_{da}+c_{bc}-c_{ba}-c_{dc}.\end{aligned} \tag{A8}$$

2. Change in $n_{\Delta}(c)$ following one square-type move

The different terms in the triangle mobility term (to be called Term 7, Term 8, Term 9, and Term 10 to avoid confusion with the previous section) are

$$n(\boldsymbol{c})_{\triangle}=\tfrac{1}{3}\mathrm{Tr}(\boldsymbol{c}^3)-\mathrm{Tr}(\boldsymbol{c}^{\updownarrow}\boldsymbol{c}^2)+\mathrm{Tr}(\boldsymbol{c}^{\updownarrow 2}\boldsymbol{c})-\tfrac{1}{3}\mathrm{Tr}(\boldsymbol{c}^{\updownarrow 3}).$$

(a) Term 7:

$$\begin{aligned}\Delta\mathrm{Tr}(\boldsymbol{c}^3) &= \sum_{ijk}[x_{ij}x_{jk}x_{ki}-c_{ij}c_{jk}c_{ki}]\\ &= 3\sum_i(c_{ia}c_{di}+c_{ic}c_{bi}-c_{ia}c_{bi}-c_{ic}c_{di}).\end{aligned}$$

(b) Term 8: Here we have to inspect first how the matrix $\boldsymbol{c}^{\updownarrow}$ of double bonds is affected by a square move:

$$x_{ij}^{\updownarrow}=c_{ij}^{\updownarrow}+\Delta_{ij}^{\updownarrow}+\Delta_{ji}^{\updownarrow}$$

with

$$\Delta_{ij}^{\updownarrow}=\delta_{ia}\delta_{jd}c_{da}+\delta_{ic}\delta_{jb}c_{bc}-\delta_{ia}\delta_{jb}c_{ba}-\delta_{ic}\delta_{jd}c_{dc}.$$

It follows that

$$\begin{aligned}\Delta\mathrm{Tr}(\boldsymbol{c}^{\updownarrow}\boldsymbol{c}^2) &= \mathrm{Tr}(\boldsymbol{x}^{\updownarrow}\boldsymbol{x}^2)-\mathrm{Tr}(\boldsymbol{c}^{\updownarrow}\boldsymbol{c}^2)=\sum_{ijk}(c_{ij}^{\updownarrow}+\Delta_{ij}^{\updownarrow}+\Delta_{ji}^{\updownarrow})\\ &\quad \times(c_{jk}+\Delta_{jk})(c_{ki}+\Delta_{ki})-\mathrm{Tr}(\boldsymbol{c}^{\updownarrow}\boldsymbol{c}^2).\end{aligned}$$

Arguments similar to those employed before show that $\sum_j\Delta_{ij}^{\updownarrow}\Delta_{jk}=\sum_i\Delta_{ij}^{\updownarrow}\Delta_{ki}=0$, whereas the remaining two "compound" terms give

$$\begin{aligned}&\sum_{ijk}\Delta_{ji}^{\updownarrow}\Delta_{jk}c_{ki}\\ &= \sum_{ijk}(\delta_{ja}\delta_{id}c_{da}+\delta_{jc}\delta_{ib}c_{bc}-\delta_{ja}\delta_{ib}c_{ba}-\delta_{jc}\delta_{id}c_{dc})\\ &\quad \times(\delta_{ja}\delta_{kd}+\delta_{jc}\delta_{kb}-\delta_{ja}\delta_{kb}-\delta_{jc}\delta_{kd})c_{ki}\\ &= (c_{da}+c_{dc})\delta_{id}(\delta_{kd}-\delta_{kb})c_{ki}\\ &\quad +(c_{bc}+c_{ba})\delta_{ib}(\delta_{kb}-\delta_{kd})c_{ki}\\ &= -(c_{da}+c_{dc})c_{bd}-(c_{bc}+c_{ba})c_{db},\end{aligned}$$

and

$$\begin{aligned}\sum_{ijk}\Delta^{\updownarrow}_{ji}c_{jk}\Delta_{ki}&=\sum_{ijk}(\delta_{ja}\delta_{id}c_{da}+\delta_{jc}\delta_{ib}c_{bc}-\delta_{ja}\delta_{ib}c_{ba}-\delta_{jc}\delta_{id}c_{dc})\\&\quad\times(\delta_{ka}\delta_{id}+\delta_{kc}\delta_{ib}-\delta_{ka}\delta_{ib}-\delta_{kc}\delta_{id})c_{jk}\\&=(c_{da}+c_{ba})\delta_{ja}(\delta_{ka}-\delta_{kc})c_{jk}\\&\quad+(c_{bc}+c_{dc})\delta_{jc}(\delta_{kc}-\delta_{ka})c_{jk}\\&=-[(c_{da}+c_{ba})c_{ac}+(c_{bc}+c_{dc})c_{ca}].\end{aligned}$$

The product of three Δs can be immediately seen to be zero by earlier arguments (repeated suffix in different positions). The other terms evaluate as follows:

$$\begin{aligned}\sum_{ijk}\Delta^{\updownarrow}_{ij}c_{jk}c_{ki}&=\sum_{\beta\in\{a,c\},\alpha\in\{d,b\}}\mathbb{I}(\alpha,\beta)c_{\alpha k}c_{\alpha\beta}c_{k\beta},\\\sum_{ijk}\Delta_{ki}c^{\updownarrow}_{ij}c_{jk}&=\sum_{\beta\in\{a,c\},\alpha\in\{d,b\}}\mathbb{I}(\alpha,\beta)c^{\updownarrow}_{\alpha k}c_{k\beta},\\\sum_{ijk}\Delta_{jk}c^{\updownarrow}_{ij}c_{ki}&=\sum_{\beta\in\{a,c\},\alpha\in\{d,b\}}\mathbb{I}(\alpha,\beta)c_{\alpha k}c^{\updownarrow}_{k\beta},\end{aligned}\tag{A9}$$

where $\mathbb{I}(\alpha,\beta)$ is an indicator function which evaluates to 1 if bond (α,β) is created by the present move, to -1 if the bond (α,β) is destroyed, and zero otherwise. Similarly,

$$\begin{aligned}\sum_{ijk}\Delta^{\updownarrow}_{ji}c_{jk}c_{ki}&=\sum_{ijk}c_{jk}c_{ki}\times(\delta_{ja}\delta_{id}c_{da}+\delta_{jc}\delta_{ib}c_{bc}\\&\quad-\delta_{ja}\delta_{ib}c_{ba}-\delta_{jc}\delta_{id}c_{dc})\\&=\sum_{k}(c_{ak}c_{kd}c_{da}+c_{ck}c_{kb}c_{bc}\\&\quad-c_{ak}c_{kb}c_{ba}-c_{ck}c_{kd}c_{dc}).\end{aligned}$$

Putting all of these subterms together yields

$$\begin{aligned}\Delta[\mathrm{Tr}(\boldsymbol{c}^{\updownarrow}\boldsymbol{c}^2)]&=\sum_{\beta\in\{a,c\},\alpha\in\{d,b\}}\mathbb{I}(\alpha,\beta)\sum_{k}[c_{\alpha\beta}(c_{\alpha k}c_{k\beta}+c_{\beta k}c_{k\alpha})\\&\quad+c^{\updownarrow}_{\alpha k}c_{k\beta}+c_{\alpha k}c^{\updownarrow}_{k\beta}]-c_{bd}(c_{da}+c_{dc})\\&\quad+c_{ac}(c_{da}+c_{ba})+c_{db}(c_{bc}+c_{ba})\\&\quad+c_{ca}(c_{bc}+c_{dc}).\end{aligned}\tag{A10}$$

(c) Terms 9 and 10: The same steps as followed to calculate Term 8 can be also be applied to Terms 9 and 10. In combination, the above ingredients lead us to the following updated formula for the triangle mobility (10) as a result of the edge swap (A1):

$$\begin{aligned}\Delta n_{\triangle}&=\sum_{\beta\notin\{1,3\},\alpha\in\{4,2\}}\mathbb{I}(\alpha,\beta)\sum_{k}[c_{\alpha k}c_{k\beta}-c_{\alpha\beta}(c_{\alpha k}c_{k\beta}+c_{\beta k}c_{k\alpha})\\&\quad-c^{\updownarrow}_{\alpha k}c_{k\beta}-c_{\alpha k}c^{\updownarrow}_{k\beta}+c^{\updownarrow}_{\alpha k}c^{\updownarrow}_{k\beta}-c_{\alpha\beta}(c^{\updownarrow}_{\alpha k}c^{\updownarrow}_{k\beta}+c^{\updownarrow}_{k\alpha}c^{\updownarrow}_{\beta k})\\&\quad+c_{\alpha\beta}(c^{\updownarrow}_{\alpha k}c_{k\beta}+c_{\alpha k}c^{\updownarrow}_{k\beta}+c^{\updownarrow}_{k\alpha}c_{\beta k}+c_{k\alpha}c^{\updownarrow}_{\beta k})]\\&\quad-c_{bd}(c_{db}-1)[c_{da}(1-c_{ba})+c_{dc}(1-c_{bc})]\\&\quad-c_{ac}(c_{ca}-1)[c_{da}(1-c_{dc})+c_{ba}(1-c_{bc})]\\&\quad-c_{db}(c_{bd}-1)[c_{bc}(1-c_{dc})+c_{ba}(1-c_{da})]\\&\quad-c_{ca}(c_{ac}-1)[c_{bc}(1-c_{ba})+c_{dc}(1-c_{da})].\end{aligned}\tag{A11}$$

3. Change in $n_{\square(c)}$ following one triangle-type move

The triangle move is a transformation from network $\boldsymbol{c}$ to network $\boldsymbol{x}$, characterized by $x_{ij}=c_{ij}+\Omega_{ij}$, with

$$\Omega_{ij}=\delta_{ib}\delta_{ja}+\delta_{ic}\delta_{jb}+\delta_{ia}\delta_{jc}-\delta_{ia}\delta_{jb}-\delta_{ib}\delta_{jc}-\delta_{ic}\delta_{ja}.\tag{A12}$$

The terms which make up the square mobility term are

$$\begin{aligned}n_{\square}(\boldsymbol{c})&=\frac{1}{2}\mathrm{Tr}(\boldsymbol{c}\boldsymbol{c}^{\dagger}\boldsymbol{c}\boldsymbol{c}^{\dagger})-\sum_{ij}k^{\mathrm{in}}_{i}c_{ij}k^{\mathrm{out}}_{j}+\mathrm{Tr}(\boldsymbol{c}\boldsymbol{c}^{\dagger}\boldsymbol{c})\\&\quad+\frac{1}{2}N^2\langle k\rangle^2+\frac{1}{2}\mathrm{Tr}(\boldsymbol{c}^2)-\sum_{i}k^{\mathrm{out}}_{i}k^{\mathrm{in}}_{i}.\end{aligned}$$

(a) Term 2:

$$\sum_{ij}k^{\mathrm{in}}_{i}\Omega_{ij}k^{\mathrm{out}}_{j}=\sum_{\alpha,\beta\in\{1,2,3\}}\mathbb{I}(\alpha,\beta)k^{\mathrm{in}}_{\alpha}k^{\mathrm{out}}_{\beta}.$$

(b) Term 3:

$$\begin{aligned}\Delta[\mathrm{Tr}(\boldsymbol{c}\boldsymbol{c}^{\dagger}\boldsymbol{c})]&=\mathrm{Tr}(\boldsymbol{x}\boldsymbol{x}^{\dagger}\boldsymbol{x})-\mathrm{Tr}(\boldsymbol{c}\boldsymbol{c}^{\dagger}\boldsymbol{c})\\&=\sum_{ijk}(c_{ij}+\Omega_{ij})(c_{kj}+\Omega_{kj})\\&\quad\times(c_{ki}+\Omega_{ki})-c_{ij}c_{kj}c_{ki}.\end{aligned}$$

We consider each subterm separately:

$$\begin{aligned}\sum_{ijk}\Omega_{ij}c_{kj}c_{ki}&=\sum_{ijk}\Omega_{ki}c_{kj}c_{ij}=0,\\\sum_{ijk}\Omega_{kj}c_{ij}c_{ki}&=\sum_{\alpha,\beta\in\{1,2,3\}}\mathbb{I}(\alpha,\beta)\sum_{i}c_{\alpha i}c_{i\beta}.\end{aligned}$$

Clearly,

$$\sum_{ijk}\Omega_{ij}\Omega_{kj}=\sum_{ijk}\Omega_{ki}\Omega_{kj}=0,$$

since the Ω kills any suffix repeated in the same position. Furthermore,

$$\sum_{i}\Omega_{ij}\Omega_{ki}=\sum_{\alpha,\beta\in\{a,b,c\}}(1-\delta_{\alpha\beta})\delta_{j\alpha}\delta_{k\beta}-2\sum_{\alpha\in\{a,b,c\}}\delta_{j\alpha}\delta_{k\alpha},$$

hence

$$\sum_{ijk}\Omega_{ij}\Omega_{ki}c_{kj}=3.$$

So it follows that

$$\mathrm{Tr}(\boldsymbol{x}\boldsymbol{x}^{\dagger}\boldsymbol{x})-\mathrm{Tr}(\boldsymbol{c}\boldsymbol{c}^{\dagger}\boldsymbol{c})=3+\sum_{\alpha,\beta\in\{a,b,c\}}\mathbb{I}(\alpha,\beta)\sum_{i}c_{\alpha i}c_{i\beta}.$$

(c) Term 5: We observe that $\sum_{ij}c_{ji}\Omega_{ij}=3$ and $\sum_{ij}\Omega_{ij}\Omega_{ji}=-6$. We conclude that $\Delta[\mathrm{Tr}(\boldsymbol{c}^2)]=0$. This is as expected, since double bonds cannot participate in a *triangle swap*.

(d) Term 1: Finally we return to Term 1 using the various shortcuts derived above. We recall that a suffix repeated in the same position sends the term to zero. Hence, we already know that all terms featuring the product of three or four Ω terms will be zero. Next,

$$\sum_{j}\Omega_{ij}c_{kj}=\sum_{\alpha,\beta\in\{1,2,3\}}\mathbb{I}(\alpha,\beta)\delta_{i\alpha}c_{k\beta}.$$

From this it follows that

$$\sum_{ij} \Omega_{ij} c_{kj} \Omega_{km} c_{im} = 0.$$

Finally,

$$\sum_{ijkm} \Omega_{ij} c_{kj} c_{km} c_{im} = c_{km}[c_{k1}(c_{2m} - c_{3m}) + c_{k2}(c_{3m} - c_{1m}) + c_{k3}(c_{1m} - c_{2m})]$$

(and similarly with the other terms related to this one by simple permutations). Overall, we thus find

$$\Delta \mathrm{Tr}(\boldsymbol{c}\boldsymbol{c}^\dagger \boldsymbol{c}\boldsymbol{c}^\dagger) = 4 \sum_{km} c_{km} \sum_{\alpha,\beta \in \{a,b,c\}} \mathbb{I}(\alpha,\beta) c_{\alpha m} c_{k\beta}.$$

Collecting all of these terms together, we see that the expected change in the square mobility term after the application of a single triangle-type move is

$$\Delta[n_\square] = \sum_{\alpha,\beta \in \{a,b,c\}} \mathbb{I}(\alpha,\beta) \Big[c_{\alpha i} c_{i\beta} + k_\beta^{\mathrm{out}} k_\alpha^{\mathrm{in}} + 2 \sum_{km} c_{km} c_{\alpha m} c_{k\beta} \Big] + 3. \tag{A13}$$

4. Change in $n_\Delta(c)$ following one triangle-type move

This final incremental term is best evaluated by an algorithm which, for each edge created or destroyed, searches for monodirected triangles that have been created or destroyed.

[1] M. Molloy and B. Reed, Random Struct. Algor. **6**, 161 (1995).
[2] R. Albert and A. L. Barabási, Rev. Mod. Phys. **74**, 47 (2002).
[3] A. R. Rao, R. Jana, and S. Bandyopadhyay, Sankhya: Indian J. Stat. Ser. A **58** (1996).
[4] T. Squartini, G. Fagiolo, and D. Garlaschelli, Phys. Rev. E **84**, 046118 (2011).
[5] T. Squartini, G. Fagiolo, and D. Garlaschelli, Phys. Rev. E **84**, 046117 (2011).
[6] T. Squartini and D. Garlaschelli, New J. Phys. **13**, 083001 (2011).
[7] C. I. Del Genio, H. Kim, Z. Toroczkai, and K. E. Bassler, PLoS ONE **5**, e10012 (2010).
[8] S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon, Nature Genet. **31**, 64 (2002).
[9] G. A. Pagani and M. Aiello, e-print arXiv:1105.3338.
[10] T. Ohnishi, H. Takayasu, and M. Takayasu, J. Econ. Inter. Coord. **5**, 171 (2010).
[11] M. E. J. Newman, D. J. Watts, and S. H. Strogatz, Proc. Natl. Acad. Sci. USA **99**, 2566 (2002).
[12] S. Maslov and K. Sneppen, Science **296**, 910 (2002).
[13] N. J. Gotelli and W. Ulrich, Oikos **121**, 171 (2012).
[14] O. D. King, Phys. Rev. E **70**, 058101 (2004).
[15] H. Klein-Hennig and A. K. Hartmann, Phys. Rev. E **85**, 026101 (2012).
[16] H. Kim, C. I. D. Genio, K. E. Bassler, and Z. Toroczkai, New J. Phys. **14**, 023012 (2012).
[17] S. Itzkovitz, R. Milo, N. Kashtan, G. Ziv, and U. Alon, Phys. Rev. E **68**, 026127 (2003).
[18] A. Coolen, A. De Martino, and A. Annibale, J. Stat. Phys. **136**, 1035 (2009).
[19] R. Milo, N. Kashtan, S. Itzkovitz, M. E. J. Newman, and U. Alon, e-print arXiv:cond-mat/0312028.
[20] J. J. Seidel, in *In Colloquio Internazionale sulle Teorie Combinatorie Tomo I*, edited by A. Doe (Accad. Naz. Lincei, Rome, 1973), p. 481.
[21] R. Taylor, in *Combinatorial Mathematics VIII*, edited by K. McAvaney (Springer, Berlin, 1981).
[22] E. S. Roberts, T. Schlitt, and A. C. C. Coolen, J. Phys. A **44**, 275002 (2011).
[23] M. Matsumoto and T. Nishimura, ACM Trans. Model. Comput. Simul. **8**, 3 (1998).
[24] T. R. Hughes *et al.*, Cell **102**, 109 (2000).
[25] C. T. Harbison *et al.*, Nature (London) **431**, 99 (2004).
[26] L. P. Fernandes, A. Annibale, J. Kleinjung, A. C. C. Coolen, and F. Fraternali, PLoS ONE **5**, e12083 (2010).
[27] Y. Artzy-Randrup and L. Stone, Phys. Rev. E **72**, 056708 (2005).
[28] A. Annibale, A. C. C. Coolen, L. P. Fernandes, F. Fraternali, and J. Kleinjung, J. Phys. A **42**, 485001 (2009).
[29] S. Maslov, K. Sneppen, and A. Zaliznyak, Physica A **333**, 529 (2004).
[30] M. E. J. Newman, Phys. Rev. Lett. **89**, 208701 (2002).
[31] G. Ansmann and K. Lehnertz, Phys. Rev. E **84**, 026103 (2011).
[32] G. Basler, O. Ebenhöh, J. Selbig, and Z. Nikoloski, Bioinformatics **27**, 1397 (2011).