Supplementary Material

Effect of Network Density on ω

The main text discusses the effect of network size on the form of the ω curve. Similar simulations were performed to examine the effects of varying the network density. To this end, four 1000-node regular lattice networks were simulated at varying densities (density = 1%, 5%, 10%, and 20%). Each network was rewired at random according to the Watts and Strogatz rewiring regime (Watts and Strogatz 1998). The resulting clustering coefficient, path length, and ω values are shown versus each corresponding rewire probability (Fig. S1).

In general, as the density of a network increases, the nodes of the network become more connected. Therefore the average path length in dense networks is generally shorter. Thus, the maximum improvement in path length due to random rewiring becomes limited for these dense networks, and ω of the lattice network cannot achieve very negative values. This explains the curve seen in Figure 2 for a network size of 100, where the lower tail of the omega curve does not achieve -1. This small network has an average degree of 16 and therefore a density of approximately 32%.

Real-World Networks

Small-world properties were investigated in 10 networks found in the literature and various databases. These networks include various biological, social, and technological networks. As seen in Table S1, the small-world coefficient, σ, indicates that most of these networks are small-world. In contrast, by using ω, the classification of some of these networks as small-world comes into question. Moreover, ω allows for ranking of these networks on a continuum between lattice and random. As shown in Figure 5B, σ does not characterize clustering in networks and is greatly influenced by clustering in the random network. These results parallel the results found in brain network data (Fig. 5A).

Network Latticization

Lattice networks were generated by using a modified version of the Sporns–Zwi "latticization" algorithm (Sporns and Zwi, 2004) found in the brain connectivity toolbox (www.brain-connectivity-toolbox.net) (Rubinov and Sporns 2010). The procedure is based on a Markov-chain algorithm that maintains node degree and swaps edges with uniform probability; however, swaps are carried out only if the resulting matrix has entries that are closer to the main diagonal. The algorithm requires two inputs: the adjacency matrix of the network and the number of iterations (i.e., the number of times each node is rewired on average). Using the algorithm in its present form generates a network that places most of the nodes along the main diagonal of the matrix. However, because this algorithm is based on random swaps, the clustering in the network is not necessarily maximized. Increasing the number of iterations may increase clustering, but this often comes at the cost of increased processing time. Although this algorithm works well with smaller networks, it becomes computationally burdensome with larger networks such as the Internet, which takes several hours to process.

To address these issues, a new latticization algorithm was developed that slightly modifies the Sporns–Zwi algorithm. The latticization algorithm used in this study uses a two-step procedure to produce a lattice network with optimized clustering (Fig. S3). The first step simply uses the Sporns–Zwi algorithm with five iterations on the original adjacency and produces an input matrix, A_{ij}(in) (i.e., the matrix that is used by the clustering optimization algorithm). Five iterations are used because it appears to strike a balance between processing time and networks with most nodes along the main diagonal. The second step uses the input matrix, A_{ij}(in), and a user-defined number of repetitions, R, to ultimately generate a lattice network, A_{ij}(latt). The clustering optimization algorithm uses the Sporns–Zwi algorithm, but uses one iteration, to produce an output matrix, A_{ij}(opt). Following this procedure, the clustering of A_{ij}(opt) is compared to A_{ij}(in). If A_{ij}(opt) has higher clustering, it replaces A_{ij}(in) as the new input matrix. If A_{ij}(opt) has lower clustering, then A_{ij}(in) is kept as the input matrix. By using the old or new input matrix, the clustering optimization procedure is performed until the number of user-defined repetitions is fulfilled. The resultant matrix, A_{ij}(latt), is a lattice network with optimized clustering. This latticization algorithm is useful for small and mid-sized networks because it ensures highly clustered networks. Furthermore, this algorithm takes less processing time than simply increasing the number of iterations in the Sporns–Zwi algorithm, which does not necessarily produce optimized clustering.

As mentioned earlier, larger networks become a computational burden for the Sporns–Zwi algorithm as well as for the latticization algorithm used in this study. To address this, an alternative method was developed for larger networks such as the Internet or voxel-based functional brain networks. Instead of using the latticization algorithm on the entire network, the algorithm is used on smaller subnetworks (or partitions) along the main diagonal. For example, instead of using the latticization algorithm on a full 5000×5000 matrix, smaller 500×500 partitions along the main diagonal can be latticized. Using this approach decreases processing time because smaller networks are used by the algorithm.

The "sliding window" algorithm detailed here was used on all the larger networks in this study (>1000 nodes). As seen in Figure S3, the algorithm uses a three-step procedure to generate a lattice network. The initial step still uses the Sporns–Zwi algorithm with five iterations. This step was used for all matrices regardless of network size because it provides a good starting point for the latticization algorithm. The second and third steps use the input matrix, A_{ij}(in); a user-defined number of repetitions, R; and a window size, which represents the size of the partitions to be latticized.

Before use of the sliding window procedure, corner latticization is performed. Assuming a ring-like structure to the network, nodes toward the beginning and end of the main diagonal are considered close to each other. Connections among these nodes populate the corners of the matrix; thus, the sliding window procedure bypasses these nodes. On the basis of the input window size, the corner latticization step extracts four equally sized corners from the input matrix and joins them to form a smaller "corner matrix." By using the

latticization algorithm detailed in Figure S3, the corner matrix is latticized. After latticization, the corner matrix is split into four parts and reinserted into the input matrix, replacing the entries of the original matrix. With use of this new matrix, the sliding window procedure is used on the full input matrix. Along the main diagonal, a partition equal to the input window size is extracted, latticized, and reinserted into the matrix. Moving half the distance of the window along the main diagonal, another partition is extracted and latticized as before. The sliding window procedure continues until the entire matrix is covered, resulting in a lattice network.

Network Resources

Biological, social, and technological networks were obtained from various sources. All networks in this study were analyzed as unweighted and undirected graphs. For disconnected graphs, network analysis was done on the largest component of the network. The US airlines network (Batagelj and Mrvar, 2006) was obtained from the Pajek datasets (http://vlado.fmf.uni-lj.si/pub/networks/data/). The e-mail (Guimerà et al., 2003), jazz (Gleiser and Danon, 2003), and *C. elegans* metabolic (Duch and Arenas, 2005) network were obtained from Alex Arenas's network datasets (http://deim.urv.cat/~aarenas/data/welcome.htm). The karate (Zachary 1977), word adjacency (Newman 2006), football (Girvan and Newman, 2002), dolphin (Lusseau et al., 2003), and Internet networks were obtained from Mark Newman's network data sets (www-personal.umich.edu/~mejn/netdata/). The Internet network is from unpublished data by the University of Oregon Route Views Project (http://routeviews.org/).

Brain Imaging Data Collection

Scanning protocol

Brain imaging data were collected from 11 healthy older adults as part of a separate study evaluating an exercise program (Burdette et al., 2010). All data reported here are from post-treatment scans with participants in the control or the treatment group. Functional magnetic resonance imaging (fMRI) data were collected on a 1.5-T GE twin-speed LX scanner with a birdcage head coil (GE Medical Systems, Milwaukee, WI). Functional imaging was performed by using multi-slice gradient-echo planar images (repetition time/echo time, 2000 msec/40 msec; field of view, 24 cm (frequency) × 15 cm (phase); matrix size, 96 × 86; number of slices, 40; thickness, 5 mm; no skip; voxel resolution, 3.75 mm × 3.75 mm × 5 mm). Participants performed no task but were asked to keep their eyes open for the 6-min 20-sec (190 images) resting fMRI. Images were motion corrected, normalized to Montreal Neurological Institute space, and resliced to a 4 × 4 × 5-mm voxel size using SPM99 (Wellcome Trust Centre for Neuroimaging, London, United Kingdom). All participants gave written informed consent, and the study was approved by the Institutional Review Board at Wake Forest University School of Medicine, Winston-Salem, North Carolina.

Network analysis

Whole-brain functional connectivity was evaluated by using graph theory methods on a voxel-by-voxel basis (Eguíluz et al., 2005; Hayasaka and Laurienti 2010; van den Heuvel et al., 2008). The fMRI time courses were extracted for each voxel in gray matter (approximately 15,000) and band-pass–filtered to remove signal outside the range of 0.009–0.08 Hz (Fox et al., 2005; van den Heuvel et al., 2008). Network analysis was based on subject-specific gray-matter tissue maps with mean white matter and cerebrospinal fluid signal regressed from the filtered time series to account for physiological noise. The six rigid-body motion parameters from the motion correction process were also regressed from the time series. A correlation matrix was produced by computing the Pearson correlation coefficient between all voxel pairs within fMRI time series. A threshold was applied to the correlation matrix, whereby voxel pairs above the threshold were considered connected and assigned a value of 1, and voxel pairs below the threshold were considered not connected and assigned a value of 0. This binary matrix produces an undirected, unweighted adjacency matrix (A_{ij}) representing the whole-brain functional connectivity for each participant. To make networks across subjects comparable, the threshold was defined such that the relationship between the number of nodes N and the average node degree K was the same across all subjects. Specifically, the threshold was defined so that $S = \log(N)/\log(k)$ with $S = 2.5$ across all subjects for the study (Stam et al., 2007; Supekar et al., 2008). From the adjacency matrix, the following graph metrics were calculated at each node and were averaged to yield means for the entire network: degree (k), clustering coefficient (C), and minimum path length (L). Details on calculating these graph metrics for an undirected, unweighted graph can be found in Rubinov and Sporns (2010).

References

Batagelj V, Mrvar A. 2006. Pajek datasets. http://vlado.fmf.uni-lj.si/pub/networks/data/ Last accessed Sep. 8, 2011.

Burdette JH, Laurienti PJ, Espeland MA, et al. 2010. Using network science to evaluate exercise-associated brain changes in older adults. Front Aging Neurosci 2:23.

Duch J, Arenas A. 2005. Community detection in complex networks using extremal optimization. Phys Rev E Stat Nonlin Soft Matter Phys 72:027104.

Eguíluz VM, Chialvo DR, Cecchi GA, Baliki M, Apkarian AV. 2005. Scale-free brain functional networks. Phys Rev Lett 94:018102.

Fox MD, Snyder AZ, Vincent JL, Corbetta M, Van Essen DC, Raichle ME. 2005. The human brain is intrinsically organized into dynamic, anticorrelated functional networks. Proc Natl Acad Sci U S A 102:9673–9678.

Girvan M, Newman MEJ. 2002. Community structure in social and biological networks. Proc Natl Acad Sci U S A 99:7821–7826.

Gleiser P, Danon L. 2003. Community structure in jazz. Advanc Complex Syst 6:565–573.

Guimerà R, Danon L, Díaz-Guilera A, Giralt F, Arenas A. 2003. Self-similar community structure in a network of human interactions. Phys Rev E Stat Nonlin Soft Matter Phys 68:065103.

Hayasaka S, Laurienti PJ. 2010. Comparison of characteristics between region- and voxel-based network analyses in resting-state fMRI data. Neuroimage 50:499–508.

Lusseau D, Schneider K, Boisseau OJ, Haase P, Slooten E, Dawson SM. 2003. The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. Behav Ecol Sociobiol 54(4): 396–405.

Newman MEJ. 2006. Finding community structure in networks using the eigenvectors of matrices. Phys Rev E Stat Nonlin Soft Matter Phys 74:036104.

Rubinov M, Sporns O. 2010. Complex network measures of brain connectivity: uses and interpretations. Neuroimage 52:1059–1069.

Sporns O, Zwi J. 2004. The small world of the cerebral cortex. Neuroinformatics 2:145–162.
Stam C, Jones B, Nolte G, Breakspear M, Scheltens P. 2007. Small-world networks and functional connectivity in Alzheimer's disease. Cereb Cortex 17:92–99.
Supekar K, Menon V, Rubin D, Musen M, Greicius MD. 2008. Network analysis of intrinsic functional brain connectivity in Alzheimer's disease. PLoS Comput Biol 4: e1000100.
van den Heuvel MP, Stam CJ, Boersma M, Hulshoff Pol HE. 2008. Small-world and scale-free organization of voxel-based resting-state functional connectivity in the human brain. Neuroimage 43:528–539.
Watts DJ, Strogatz SH. 1998. Collective dynamics of 'small-world' networks. Nature 393:440–442.
Zachary WW. 1977. An information flow model for conflict and fission in small groups. J Anthropol Res 33:452–473.

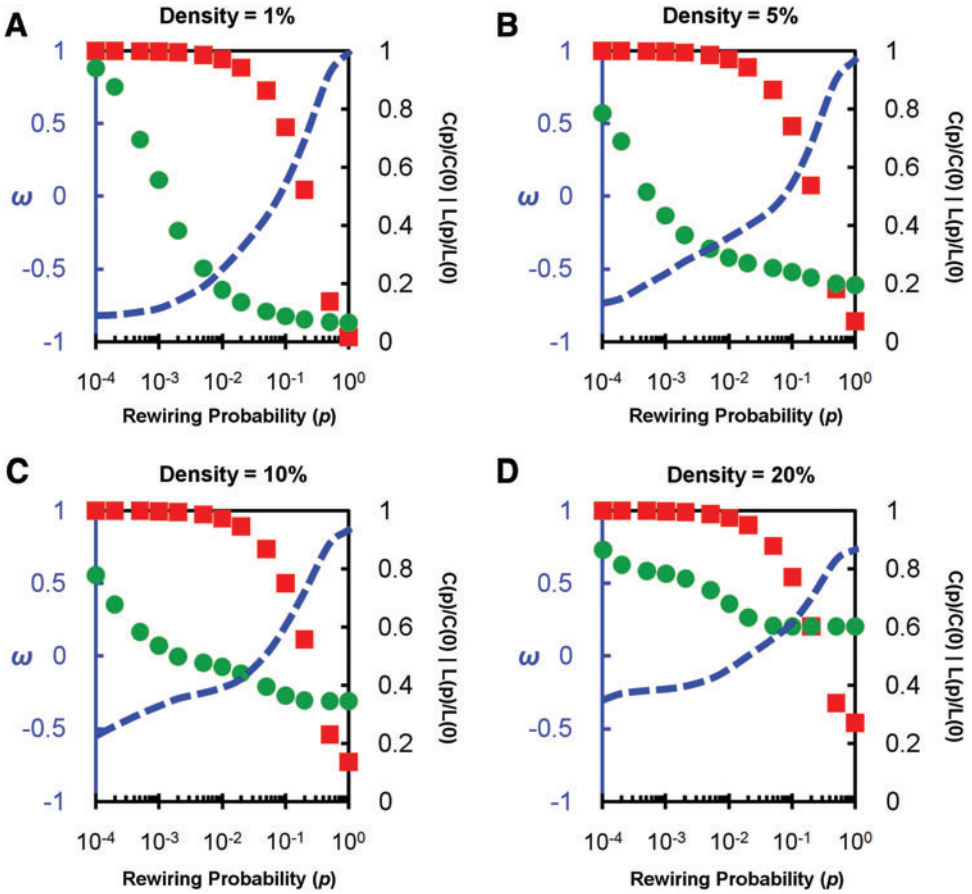

FIG. S1. Effect of network density on ω. Four 1000-node networks were simulated at densities of 1% **(A)**, 5% **(B)**, 10% **(C)**, and 20% **(D)**. Each network was rewired at a given rewire probability to span the range from lattice to small-world to random. Clustering and path length as well as ω values for each network are shown.

Table S1. Network Statistics of Several Well-Known Biological, Social, and Technological Networks

Network	N	K	C	C_{rand}	C_{latt}	L	L_{rand}	σ	ω
Karate (Zachary 1977)	35	4.46	0.55	0.31	0.65	2.41	2.24	1.66	0.08
Dolphin (Lusseau et al., 2003)	62	5.13	0.26	0.10	0.57	3.36	2.71	2.03	0.35
Word adjacency (Newman 2006)	112	7.59	0.17	0.19	0.69	2.54	2.49	0.89	0.73
Football (Girvan and Newman, 2002)	115	10.66	0.40	0.08	0.67	2.51	2.24	4.67	0.29
Jazz (Gleiser and Danon, 2003)	198	27.70	0.62	0.26	0.76	2.24	1.99	2.08	0.08
US airlines (1997) (Batagelj and Mrvar, 2006)	332	12.81	0.63	0.43	0.73	2.74	2.48	1.32	0.04
Caenorhabditis elegans (metabolic) (Duch and Arenas, 2005)	453	8.94	0.65	0.28	0.80	2.66	2.50	2.18	0.12
E-mail (Guimerà et al., 2003)	1133	9.62	0.22	0.09	0.55	3.60	3.27	8.14	0.56
Protein interactions (Jeong et al., 2001)	1539	2.67	0.07	0.04	0.19	6.81	5.69	1.47	0.47
Internet	22,963	4.22	0.23	0.09	0.68	3.84	3.58	2.28	0.51

The networks were each obtained from previous work as referenced. N, network size; K, degree; C, clustering coefficient; C_{rand}, clustering of an equivalent random network; C_{latt}, clustering of an equivalent lattice network; L, path length; σ and ω, small-worldness metrics.

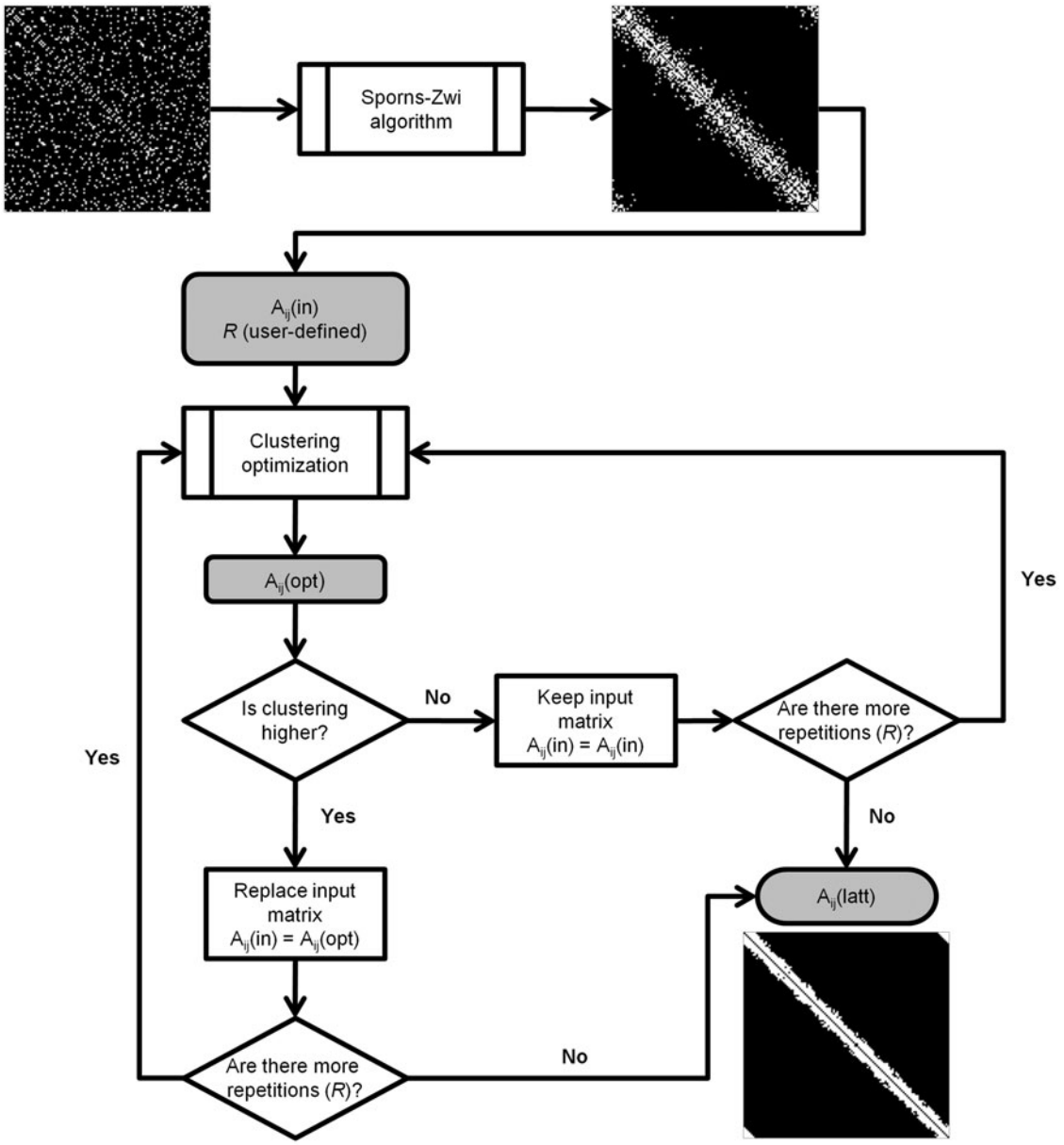

FIG. S2. Schematic of network latticization procedure.

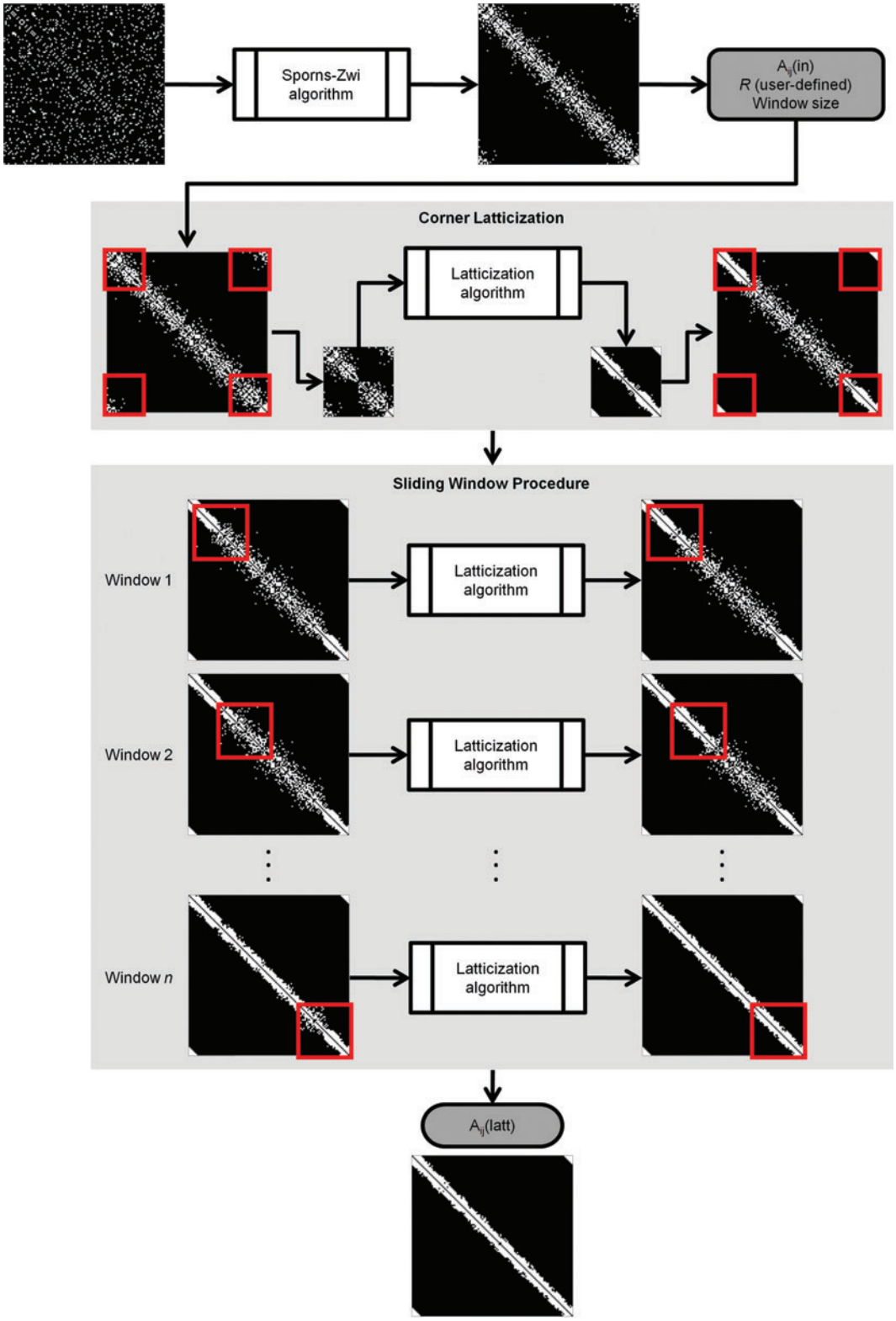

FIG. S3. Schematic of latticization procedure for larger networks. In larger networks, corner latticization and the sliding window procedures are used to speed up processing time.