Supplementary Information: Consensus clustering in complex networks

Andrea Lancichinetti & Santo Fortunato

1 Selection of the optimal number of runs and threshold τ

For any implementation of our method there are two parameters that need to be set before starting the computation: 1) the number r of partitions to be combined in the consensus matrix (see Methods); 2) the threshold τ used to filter the entries of the consensus matrix, to avoid that the latter becomes too dense, slowing down the procedure. In Figs. S1 and S2 we show how these numbers are chosen. Fig. S1 displays the Normalized Mutual Information (NMI) between the consensus partition and the planted partition of the benchmark graphs used for Fig. 1, for different values of r and a specific value of the mixing parameter μ. Each curve corresponds to a different value for the threshold τ, which equals 0, 0.5 and 0.7. Each panel presents the result of a different clustering algorithm; the value of μ varies for each method because consensus is the most effective the more diverse the input partitions are. Therefore we picked the value of μ at which the original method starts failing ($\mu = 0.7$ for Louvain, $\mu = 0.6$ for the LPM, $\mu = 0.65$ for SA and $\mu = 0.4$ for Clauset et al.. From Fig. S1 we deduce that for $r \approx 50$ one reaches an optimal partition with consensus clustering, which remains stable for larger values. This seems to hold regardless of the value of the threshold parameter τ. Therefore, in our tests of Fig. 1 we have taken r between 50 and 100.

In Fig. S2 we show instead how the threshold τ affects the results. We use the same benchmark graphs as in Fig. S1, and two values for the number of runs r: 20 and 50. The y-axis reports again the value of the NMI between the consensus and the planted partition of the benchmark. We see that the ranges of optimal values for τ depend on the clustering technique adopted. For Louvain, it is best to choose a low threshold, for the LPM τ-values in the range $[0.3, 0.7]$ give optimal results, for SA the best value span a shorter range (from 0.5 to 0.7) and for Clauset et al. the best results are obtained for fairly high values of the threshold.

2 On the effectiveness of consensus clustering

To understand why consensus clustering is so effective at detecting the clusters of the LFR benchmarks, we discuss here a simpler model which resembles the LFR benchmarks. We focus on modularity optimization because it is easier to understand how consensus improves the method.

We consider a graph with C cliques of n_c vertices each. The cliques are connected by placing $C * h$ edges between randomly chosen pairs of vertices, where the vertices of each pair belong to different cliques. It can be proven that for this kind of graph, the modularity function is optimal

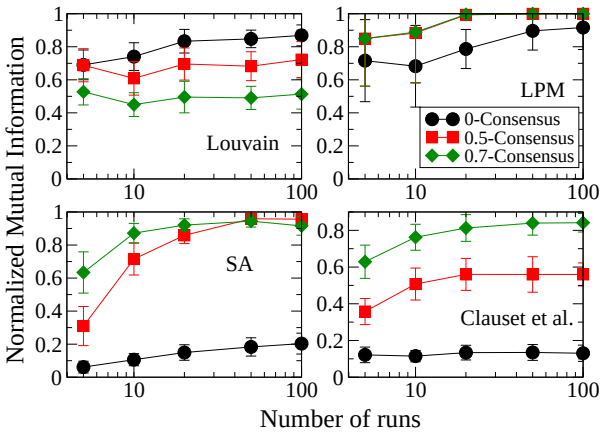

Figure S1: Accuracy of consensus clustering with the number of input runs r. Each panel reports the NMI between the planted partition of the LFR benchmark graphs used for Fig. 1, at a given μ (see text), as a function of the number of runs for a specific method. The symbols refer to three different choices for the threshold parameter τ: 0 (circles), 0.5 (squares), 0.7 (diamonds).

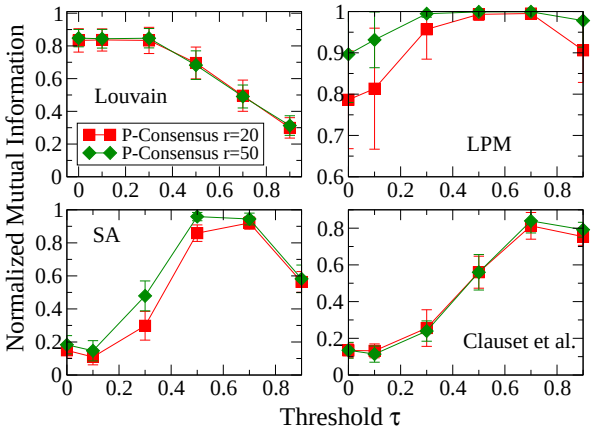

Figure S2: Accuracy of consensus clustering with the threshold parameter τ. Each panel reports the NMI between the planted partition of the LFR benchmark graphs used for Fig. S1, as a function of τ for a specific method. The symbols refer to two different choices for the number of runs r: 20 (squares), 50 (diamonds).

for a partition of $\sqrt{M}$ modules of equal size, where M is the number of edges. This result has been proved for a ring of cliques, but it is straightforward to verify that the same proof can be extended to this case.

Since there is a high number of combinations to group the cliques together in order to reach the optimal number of modules, we expect that, on average, each clique will be joined to some of its neighboring cliques with roughly equal probability. If we call g the average number of neighboring cliques, the probability that two neighboring cliques are found in the same cluster is simply $\frac{g}{2h}$ (we recall that we placed $C * h$ links).

If $C \gg 1$ and h is small enough so that the network of cliques is sparse, there will be a very small number of edges between the cliques grouped in the same module. The smallest number of edges necessary to keep n vertices connected is $n - 1$ (which gives a tree-like structure), and in such a case every vertex has an average degree $\langle k \rangle \approx 2$, when $n \gg 1$. In the case of a tree, we would have that $g \approx 2$. The probability for two cliques to be connected if their distance in the clique network is d, will be, in general,

$$p_d \approx \frac{1}{h(2h-1)^{d-1}}. \tag{S1}$$

Fig. S3 shows that this approximation is not bad especially for high values of C. In the following plots we considered $h = 10$, $n_c = 10$.

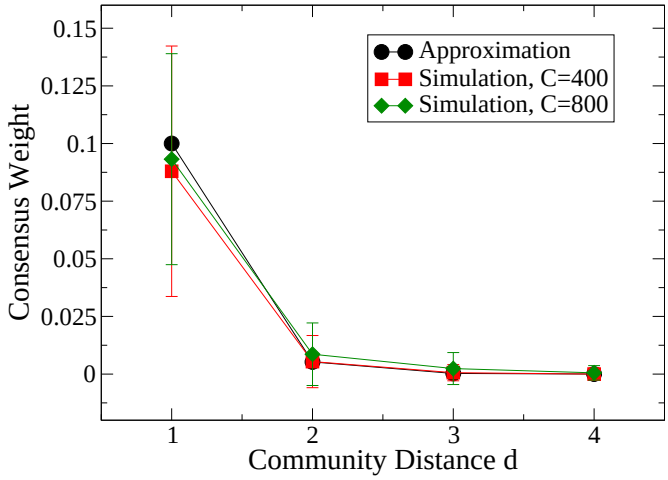

Figure S3: Probability of joining cliques in our stylized model graph with communities. Consensus weight indicates the probability that a pair of cliques are joined together. This is plotted as a function of the distance between cliques in the community network, i.e. the graph where cliques are to be seen as supervertices.

Eventually, we might expect that choosing a value of the threshold $\tau > p_1$, the consensus matrix will consist of C disconnected cliques. Modularity optimization instead would always merge cliques together in larger clusters. Indeed, the NMI between the planted partition and the input partitions is much lower than the NMI between the planted and the consensus partition already for $\tau = 0.3$ (Fig. S4).

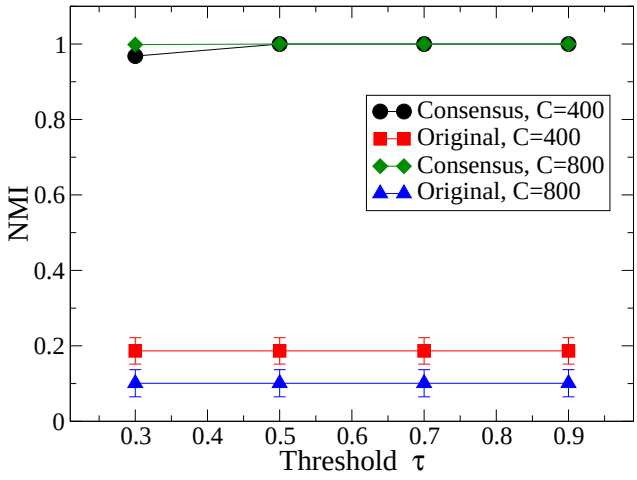

Figure S4: Fidelity of consensus partition. Normalized Mutual Information between the planted partition and the input partitions (squares and diamonds) and between the planted and the consensus partition (circles and triangles) as a function of the threshold τ. We consider two values for the number of cliques of the model network, 400 and 800.

3 Stability and fidelity of consensus partitions

In Figs. 3 and 4 we have shown that the consensus partitions are more stable than the best partitions. However trivial partitions, like the one where all vertices are together in the same cluster, would be the most stable possible, although they would be completely unrelated to the input partitions. To prove that the consensus partitions are actually very close to the input ones, Fig. S5 shows the Normalized Mutual Information of the input partitions among themselves (the average value of NMI among all pairs of different partitions) and the average value of NMI between the input partitions and the consensus partition, for different values of the threshold τ, for the neural network of *C. elegans*. Fig. S6 shows the same plot for the APS citation network of papers published in 1960. Indeed the consensus partition is often even closer to the input partitions than the latter are to each other, with the additional advantage of being more stable. This does not hold

only when the threshold is too high, because in this case the consensus partition is made of small clusters, since too many connections carry a weight under the threshold and are deleted. Otherwise this should explain why the consensus partition is more representative than the input. In Figs. S5 and S6 we considered 50 input partitions, but the results are practically the same if we take 100 of them.

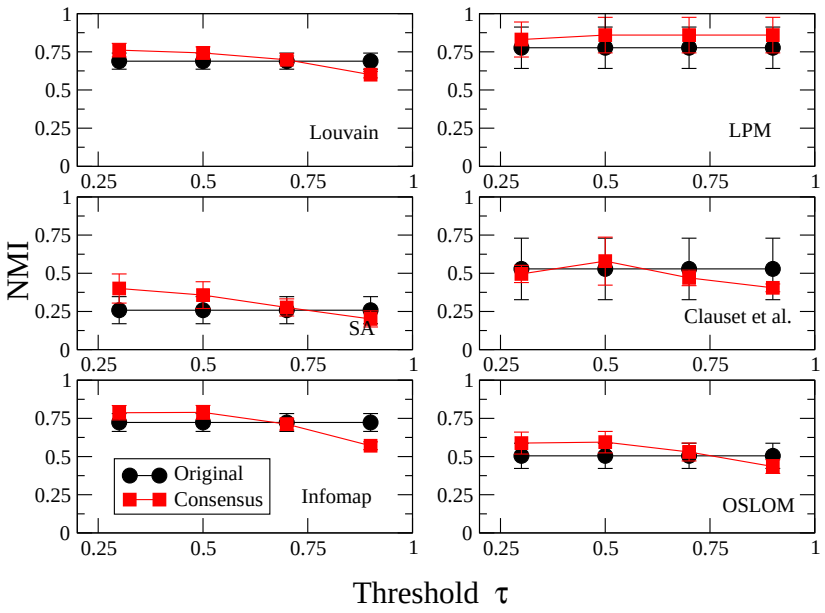

Figure S5: Fidelity of consensus partition. The black curve reports the average NMI between pairs of input partitions, the red one is the average NMI between the input partitions and the consensus one, for the neural network of *C. elegans*.

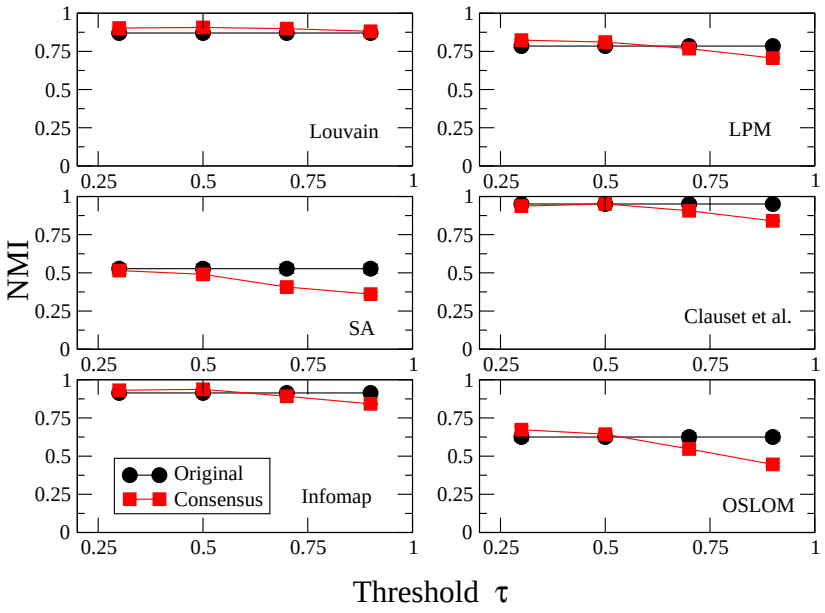

Figure S6: Fidelity of consensus partition. Same plot as Fig. S5 for a snapshot of the APS citation network, with the papers published in 1960 and those cited by them.