Effective Augmentation of Complex Networks

Jinjian Wang¹, Xinghuo Yu¹ & Lewi Stone²

Networks science plays an enormous role in many aspects of modern society from distributing electrical power across nations to spreading information and social networking amongst global populations. While modern networks constantly change in size, few studies have sought methods for the difficult task of optimising this growth. Here we study theoretical requirements for augmenting networks by adding source or sink nodes, without requiring additional driver-nodes to accommodate the change i.e., conserving structural controllability. Our “effective augmentation” algorithm takes advantage of clusters intrinsic to the network topology, and permits rapidly and efficient augmentation of a large number of nodes in one time-step. “Effective augmentation” is shown to work successfully on a wide range of model and real networks. The method has numerous applications (e.g. study of biological, social, power and technological networks) and potentially of significant practical and economic value.

Network expansion or augmentation is a ubiquitous feature in our rapidly growing technological society. It manifests in numerous and diverse scientific fields that include indexing and searching of the internet¹, information dissemination on social networks²³, financial and banking networks⁴⁻⁵, power distribution networks⁶, epidemic outbreaks in human populations⁷, and a vast range of computer applications from virus spread⁸⁻⁹ to cyber warfare¹⁰. Expansion often occurs naturally, as in social networks, where new users may join and instantly create new connections. For other networks, increases in demand often naturally lead to growth. For example, higher electricity usage requires the connection of new power stations and loads to the existing power network to balance the extra demand. In biological contexts, where many exciting frontier network projects are now appearing, questions arise as for example, how can one design a “brain on a chip” by engineering neuronal network growth¹¹.

An immediate and important question, then, is how connections should be added from the existing network so that a certain optimal performance is achieved such as maintaining or enhancing overall controllability. Despite the widespread necessity to augment networks, and the intense scientific interest in network science, very little is known about how one might begin to expand a network subject to controllability constraints. Here we provide the first systematic study that sets out to solve some of these problems.

To address the question, a necessary stepping stone is to understand the controllability of complex networks¹²⁻¹⁹. According to control theory²⁰⁻²¹, a linear time-invariant system that is controllable has the capability of being driven from any initial state to any desired final state within finite time. It is well known that controllability can be achieved by modifying the state of a small number of nodes in the network denoted as driver nodes¹²,²²,²³. Recent advances in applying control theory to complex networks show that the minimum number of drivers ($N_D$) necessary to yield control over the whole network can be simply determined by the graph theoretic technique of maximum matching¹²,²⁴ which is just the maximum set of edges that do not share start or end nodes²⁵. A node is defined to be ‘matched’ if one matching edge points to it. $N_D$ equals to number of unmatched nodes. The set of unmatched nodes is referred to as the Minimum Driver node Set (MDS)¹²,²⁶. Correspondingly, we denote the Minimum Terminal node Set (MTS) as the set of unmatched nodes in the maximum matching of its transpose network (Supplementary Fig. S1).

We study node augmentation of arbitrary directed networks while insisting that $N_D$ remain unchanged. In the present work we analyse the addition of nodes having only a single edge so that they are either source nodes (having zero in-degree $k_{in} = 0$) or sink nodes (having zero out-degree $k_{out} = 0$). Figure 1a provides an example of a 57 node network that can be fully controlled with $N_D = 11$. In Fig. 1b, the network in Fig. 1a is augmented by using our method and adding 10 new nodes (shown in triangles which includes 8 source nodes and 2 sink nodes) while $N_D = 11$ remains unchanged. However, had these ten nodes been augmented randomly, it is highly unlikely that $N_D$ would remain constant, and conceivably $N_D$ could have increased by as much as ten.

¹School of Engineering, RMIT University, Melbourne, 3000, Australia. ²School of Sciences, RMIT University, Melbourne, 3000, Australia. Correspondence and requests for materials should be addressed to X.Y. (email: x.yu@rmit.edu.au)
It is always possible to connect a new source node to an existing driver node without changing \(N_D\). This is a basic result from structural control theory based on the stem/cactus structure of the network\(^{12}\). Thus adding source nodes to the current driver nodes is considered a trivial solution to the augmentation problem, and therefore one that we do not discuss in depth or make use of in this paper. Similarly adding a sink node to a terminal node should be considered as a trivial solution to the augmentation problem. Hence, the main goal of the present paper is encapsulated in the following questions: How can one add source nodes to an existing network other than to the current driver nodes (the trivial solution), while ensuring \(N_D\) remains unchanged? Similarly how can one add sink nodes in a non-trivial way under the same constraint? And is it possible to find multiple solutions that can accomplish the same goal?

In general we are interested in determining many possible solutions to the augmentation problem. The main advantage of having multiple solutions to choose from is: with different applications, new nodes might need to be connected to quite different positions in the network in order to take into account spatial constraints. The position may need to be selected from a set of possible solutions based on the modified networks reliability, stability, vulnerability etc. For example, adding ten new nodes (substations, load, and buses) to a power network, in a way that minimises cost, maximises system reliability while ensuring \(N_D\) is unchanged. Our methodology finds multiple combinations of nodes other than the trivial existing drivers to connect to. Naturally, we also acknowledge that there is always the option of connecting to the existing drivers (i.e., the trivial solution).

Network augmentation via the addition of edge connections rather than adding nodes has previously been examined from the controllability perspective\(^{27-29}\). In these studies, augmentation was implemented by only adding new edges, with the aim of decreasing \(N_D\) and therefore enhancing controllability. However, the growth of networks involves not only increasing the number of edges but also the number of nodes. For example, in a supply chain network\(^{30}\), new storage warehouses (nodes) may need to be built in order to meet gradually increasing commodity demands. To our knowledge, currently there are no publications that address the augmentation of nodes to a network subject to controllability constraints, as is the goal of our work here.

Results

Node classification. To explore source augmentation, we examine the consequence of connecting a single source test node to any node \(x\) in an arbitrary directed network \(G(A)\), where \(A \in \mathbb{R}^{N \times N}\) is the state matrix and \(N\) is total number of nodes. We define the node \(x\) to be: i) Source Invariant (SI) if \(N_D\) does not change, and ii) Source Redundant (SR) if \(N_D\) increases. Obviously all driver nodes in a network are SI. Thus, in the example of Fig. 2a, nodes 1, 2, 3 are SI. Adding a source node to any of these nodes leaves \(N_D\) unchanged (Fig. 2b). Nodes 4 and 5 are SR (Fig. 2c).

Similarly, for sink augmentation, it is useful to examine what happens when a sink test node is added to any node \(x\) in the network (Fig. 2d). We define node \(x\) to be: i) Sink Invariant (KI) if \(N_D\) remains unchanged (e.g., node 2, 3, 5 in Fig. 2e); ii) Sink Redundant (KR) if \(N_D\) increases. It is easy to prove that source nodes are always SI and sink nodes are KI. Then, we have \(N_{SI} + N_{SR} = N, N_{KI} + N_{KR} = N\) where \(N_{SI}, N_{SR}, N_{KI}\) and \(N_{KR}\) are the number of nodes of each type. We have proved that a network's full set of SI nodes is the union of all nodes in every possible MDS. Similarly, full set of KI nodes is the union of all nodes in every possible MTS (Supplementary Note 1).

Effective source augmentation. According to the definitions of node classification, it is always possible to augment a network by adding a single source node to an SI node, or a single sink node to a KI node without changing \(N_D\). However, the method is problematical when there is a need to augment more than one single node in parallel, since the classifications of many nodes in the network change every time a new node is added. In theory, nodes can be augmented serially one after the other but the node classification procedure would need to be performed for every node of the network for each step. When the size of a network is large, the method becomes extremely inefficient.
However, there is a way around this problem. We find that SI nodes are not isolated but are “correlated” and may be grouped into distinct clusters such that connecting a new node to an SI node affects all others in the cluster in the same way. As such, it is possible to identify a set of clusters in the network, each cluster consisting of a set of correlated SI nodes. The network shown in Fig. 3a contains 10 SI nodes (red and blue) and 8 SR nodes (yellow). The SI nodes are further grouped into three clusters, {14, 1, 5, 7}, {12, 3, 15}, {4, 6} (Fig. 3c shade areas). The unique features of these clusters are that: i) Each cluster contains only one driver node (blue). ii) The number of clusters is exactly the number of sources that can be added to the network in one time-step without increasing $N_D$. If there are $k$ clusters, it is possible to augment the network with $k$ source nodes in parallel. iii) Once a source node is connected to any node in an SI cluster, all nodes in the cluster become SR and they can be ignored from this point in the augmentation process. iv) A newly connected source node replaces the driver node in a cluster, and thereby result in the network having a new MDS. Figure 3d shows that when source nodes ($S_1$, $S_2$, $S_3$) are connected to the three different clusters, all nodes in the clusters change to SR but $N_D = 4$ remains unchanged, which verifies the properties of clusters. Since all nodes apart from the drivers have become source nodes, from this point no new augmentation is possible.

Identifying clusters. SI-network. To locate all clusters, it is necessary to first simplify the network by forming the so-called ‘SI-network’ which is easier to work with. The SI-network is obtained by first locating all SR nodes and then removing all of their incoming edges (i.e., setting $k_{SR} = 0$). This procedure fragments the original network. The connected network that remains is defined as the SI network, since it contains all SI-nodes, as shown in Fig. 3b. The following discussion concerning clusters only refers to the SI-network.

V-motif. The simplest and most elementary cluster of SI-nodes possible is referred to as a V-motif and is the basis for clusters of all types. Figure 4a, displays a typical V-motif in the SI-network for the nodes $R_1 \rightarrow \{L_1, L_2\}$, where root-node $R_1$ points to two and only two leaf nodes $L_1$ and $L_2$. The nodes $\{L_1, L_2\}$ form an elementary SI cluster as long as any one of the nodes is a driver. This two-node cluster has the fundamental property that its two constituent nodes $L_1$ and $L_2$ cannot both be drivers and thus can never participate in the same MDS. The property can be inferred from a simple maximum matching of the V-motif. (We see that if an independent source node ‘S’ is augmented to $L_2$, then every node of the SI cluster becomes matched (Fig. 4c) and thus all nodes in the cluster $\{L_1, L_2\}$ change to SR nodes (Fig. 4b). The
new node ‘S’ become unmatched and replaces $L_1$ as the new driver node. Hence $N_D$ remains conserved.) Moreover for other isolated motifs of the form, say, $R_1 \rightarrow \{L_1, L_2, L_3\}$ (Fig. 4d) which is not a V-motif, maximum matching shows that the nodes $L_1, L_2, L_3$ lack the above fundamental property.

Building on the example of the V-motifs, it is possible to obtain three-node clusters and multiple-node clusters etc., each of which has the same fundamental property: there can only be one driver node within any cluster. Figure 4e gives an example of a three-node cluster (grey region) obtained from two intersecting V-motifs. Here root-node $R_1$ points to leaves $L_1$ and $L_2$ while root-node $R_2$ points to leaves $L_3$ and $L_4$. The set of SI nodes $L_1, L_2$ and $L_4$ form a cluster if any one of these nodes is a driver. Figure 4f gives an example of a four-node cluster obtained from three intersecting V-motifs. For networks with different types and varying degree distributions, V-motifs can become entangled in the SI network but may nevertheless be identified by these methods (Supplementary Fig. S2 and Supplementary Note 2).
Figure 4. Motifs in SI networks. (a-c) A demonstration of the simplest SI cluster and its features. (a) The prototypical V-motif in source augmentation consists of one root node (SI or SR) pointing to two SI leaves \( \{ L_1, L_2 \} \). Nodes \( L_1 \), a driver, and \( L_2 \) form the simplest SI cluster. For example, when an independent source node ‘S’ connects to \( L_1 \) (b), both \( L_1 \) and \( L_2 \) become SR. This can be well explained with maximum bipartite matching of the structure (c), where both leaves become always matched. Furthermore, node ‘S’ replace \( L_1 \) and becomes the new driver node (become unmatched in the -set). (d) A node configuration that has no cluster since it has no V-motif. (e) Here nodes \( L_1, L_2 \) and \( L_3 \) form a cluster being correlated via two intersecting V-motifs. (f), similar forms of four-nodes cluster via three intersecting V-motifs.

Effective sink augmentation. Similarly, the cluster concept also holds for sink augmentation. The ‘KI network’ (Supplementary Fig. S3) can be formed by removing outgoing edges of KR nodes (i.e., setting \( k_{\text{KR}} = 0 \)) instead of incoming edges. Now, the V-motif (Supplementary Figs S3 and S4) in the KI network is the same as that we have considered, but directionally reversed. Multiple KI nodes are collected in the same KI cluster if: i) KI nodes are correlated via V-motifs in the KI network; ii) there is one KI node that belongs to the MTS.

A network can be effectively augmented up to the number of KI cluster \( (N_{\text{KIC}}) \) new sink nodes. For each cluster it is only possible to connect a single sink node without changing \( N_{\text{D}} \) (Supplementary Fig. S3) and after augmentation, all nodes in the cluster become KR. Furthermore, we find that sink augmentation has no effect on the current MDS (Supplementary Note 3). Thus sources and sinks augmentation can be implemented in parallel without interference. These features make network augmentation very efficient and flexible. We have developed an algorithm to search all possible V-motifs in both SI and KI networks, and then determine all possible SI and KI clusters in linear time (see Methods).

Effective augmentation in synthetic networks. We have illustrated the importance of clusters in network augmentation. As the minimum number of new source (or sink) nodes that can be augmented by effective augmentation method depends on the numbers of SI (or KI) clusters existing in the network, there is a need to explore the latter empirically. Denote the number of SI clusters in a network as \( N_{\text{SIC}} \) and the number of KI clusters as \( N_{\text{KIC}} \). Based on their properties, we understand that SI clusters never contain source nodes and each cluster has exactly one non-source driver node. Thus for an arbitrary network \( G(A) \) the upper limit of \( N_{\text{SIC}} \) must be \( N_{\text{MAS}} = N_{\text{D}} - S \), where \( N_{\text{MAS}} \) is the maximum number of augmentable source nodes, and \( S \) is the number of source nodes in \( A \). Similarly, \( N_{\text{MAS}} \) is limited with \( N_{\text{MAS}} = N_{\text{D}} - K \), where \( K \) is the number of sink nodes in \( A \).

We first study scale-free networks\(^{31,32} \) (Supplementary Note 4) of different average degree but having identical input and output degree exponents \( (\gamma_{\text{in}} = \gamma_{\text{out}} = 3) \). Without loss of generality, we consider here networks with no disconnected nodes. For the purposes of exploring the cluster distributions over different network sizes, we normalise the parameters based on the network’s total number of nodes, \( N \), respectively, i.e., \( (n_{\text{SIC}}, n_{\text{KIC}}, n_{\text{D}}) = (N_{\text{SIC}}, N_{\text{KIC}}, N_{\text{D}})/N \). Figure 5a demonstrates that \( n_{\text{SIC}} \) (blue dots) and \( n_{\text{KIC}} \) (red squares) follow the same trend: both decrease as the mean degree of the network increases. For networks with low average degree, e.g., \( (k) = 4 \) the minimum fraction of new nodes that can be augmented in parallel is approximately 9% of the total number of nodes \( (n_{\text{SIC}} = 9\%) \). For networks with high mean degree, augmentation is limited independent of the method. For example, in a fully connected network with \( N_{\text{D}} = 1 \), only one new source node can be augmented and \( n_{\text{SIC}} = 0 \) since no V-motifs exist.

To investigate the capability of the method in capturing a high proportion of augmenting solutions, the number of clusters \( N_{\text{SIC}} \) (blue dots) and \( N_{\text{MAS}} \) (yellow triangles; the upper limit of augmentation) are plotted as a function of average degree \( (k) \) in Supplementary Fig. S5a. At \( (k) = 4 \), approximately 65% \( (N_{\text{SIC}}/N_{\text{MAS}}) \) of the total possible source nodes \( N_{\text{MAS}} \) can be augmented simultaneously via our cluster based method in SF network. Again, Fig. S5a shows that the higher the mean degree of the network, the less the number of new nodes that can be augmented non-trivially.
Effective augmentation of synthetic networks. Distribution of the minimum fraction of new source nodes \(n_{\text{SIC}}\) and sink nodes \(n_{\text{KIC}}\) that can be effectively augmented in scale-free networks (identical degree exponents \(\gamma_{\text{in}} = \gamma_{\text{out}} = 3\)) and Erdős–Rényi networks respectively. (a) \(n_{\text{SIC}}\) and \(n_{\text{KIC}}\) versus \(\langle k \rangle\) are shown in blue dots and red squares respectively. These networks have a power law degree distribution. Insert figure shows the degree distribution of the network with \(\langle k \rangle = 6\). The solid line shows the curve fitting of the presented data using non-linear polynomial fitting functions. (b) \(n_{\text{KIC}}\) and \(n_{\text{KIC}}\) in Erdős–Rényi networks with varying \(\langle k \rangle\). The degree distribution of ER graphs converges to a Poisson distribution (insert).

To explore \(n_{\text{SIC}}\) and \(n_{\text{KIC}}\) over different network types, we further investigate the parameters in Erdős–Rényi networks33 (Fig. 5b, Fig. 5b, Supplementary Note 4). We find that ER networks are less augmentable \(\left(\frac{N_{\text{SIC}}}{N_{\text{MAS}}} \approx 90\% \text{ at } \langle k \rangle = 4\right)\) compared to SF networks \(\left(65\% \text{ at same } \langle k \rangle\right)\). This is due to the fact that in general ER networks require less driver nodes to achieve full control when compared to SF networks having identical network size and mean degree12. Intriguingly, we find that in ER networks, the decrease of \(n_{\text{SIC}}\) and \(n_{\text{KIC}}\) as a function of \(\langle k \rangle\) is sharper compared with SF networks. When the average degree exceeds a critical value \(k_c \approx 8\), the networks become almost non-augmentable. This is because the maximum number of source nodes that can be augmented, \(N_{\text{MAS}}\) peaks approximately zero at \(k_c \approx 8\) (Supplementary Fig. 5b). On the other hand, \(N_{\text{MAS}}\) approaches zero when \(k_c \approx 13\) in SF-networks. Thus SF networks can be augmented within broader average degree regions than ER networks. Furthermore, we find the effective augmentation method can be applied more efficiently in ER networks as compared to SF networks. This can be observed clearly in Supplementary Fig. S5 which shows the difference between \(N_{\text{SIC}}\) and \(N_{\text{MAS}}\) for ER networks is much smaller than their difference in SF networks (compare Fig. S5a,S5b).

Effective augmentation in real networks. To demonstrate the feasibility of the augmentation algorithm, we apply the tools developed above to several real networks from a variety of different applications (Table 1, Supplementary Tables S1 and S2). Figure 6a shows the fraction of clusters (both SI and KI clusters) determined from real networks versus mean degree. Overall, both \(n_{\text{SIC}}\) and \(n_{\text{KIC}}\) decrease as \(\langle k \rangle\) increases which is in agreement with what was found from simulations of ER and scale-free networks. For example, networks as the cellular, electronic circuit and power grid with relatively low average degree have many more clusters and are thus much more augmentable than networks having large average degree, such as social networks.

It is also important to note that some networks exhibit different proportions of SI and KI clusters. We find that electronic networks can be readily augmented with sink nodes \(n_{\text{SIC}}\) \(\approx 20\%\) but are limited with source nodes \(n_{\text{KIC}} \approx 3\%\). To further investigate this property, we plot \(n_{\text{SIC}}\) versus \(n_{\text{KIC}}\) in Fig. 6b and \(n_{\text{SIC}}, n_{\text{KIC}}\) versus \(n_p\) in Supplementary Fig. S6. We find that some networks, such as cellular, food web, power grid, social networks have approximately the same fraction of SI and KI clusters, while other networks, like electronic circuit (more augmentable with sink nodes than source nodes) and transcription networks (Yeast networks are more augmentable with source nodes than sink nodes) are away from the diagonal line. This may be caused by different in- and out-degree distributions and degree asymmetries14,34,35. Furthermore, large numbers of SI clusters and KI clusters are found in cellular networks, power grid and cortical networks indicating that these sorts of networks can be augmented efficiently with the clustering method used in our effective augmentation. Finally, we also observed that networks from the same categories have similar \(n_{\text{SIC}}\) and \(n_{\text{KIC}}\), which means they have similar augmentation properties.

Discussion
The cluster method proposed here allows us to generate a relatively large number of workable solutions in one time-step. For our method, the number of ways of adding \(m\) new nodes to the network non-trivially with \(n\) clusters (assuming each cluster has \(k\) nodes) is \((k - 1)^m\) where \(m \leq n\), which is usually quite considerable. Furthermore, the solutions allows connecting multiple source and sink nodes in parallel simultaneously without conflict, which increases the effectiveness of the method. It might seem a limitation that the newly connected source and sink nodes are required to be of degree one (i.e., have one edge). However, by taking advantage of the clusters present, it is a straightforward extension of the method to augment new nodes having higher degree. Each new edge can be attached to a different cluster.
Our approach is based on the presence of V-motifs in both the SI and KI network. When the method is applied to random networks with varying average degree, we find that the ability to augment a network is inversely proportional to its average degree. This is attributed to the fact that the network clusters reduce in number as network connectivity and average degree increase. Networks having large average degree tend to have few clusters, and thus are hard to be augmented in practice, as was found for social networks. For real networks, networks from the same categories tend to have similar fractions of clusters. In summary, we have formulated a successful and efficient procedures for augmenting arbitrary directed networks while keeping the minimum number of driver nodes required to fully control the network.

**Methods**

**Identifying minimum driver node set and minimum terminal set.** The MDS and MTS of a directed network $G(V, E)$ can be identified by the following steps. 1) Bipartite representation (Supplementary Fig. S1): divide node set $V$ into two disjoint node sets $+$ and $-$, such that any directed edge $(V_i, V_j) \in E$ can be represented as $(V_i^+, V_j^-)$ in the bipartite graph $G(V^+ \cup V^-, E)$. 2) Maximum bipartite matching: determine the maximum

| Type            | Name                | N   | L   | $n_D$ | $n_{SIC}$ | $n_{KIC}$ |
|-----------------|---------------------|-----|-----|-------|-----------|-----------|
| Electronic Circuits | s208                | 122 | 189 | 0.24  | 3.28      | 19.67     |
|                 | s420                | 252 | 399 | 0.23  | 2.38      | 19.84     |
|                 | s838                | 512 | 819 | 0.23  | 1.95      | 19.92     |
| Food Web        | Everglades          | 69  | 916 | 0.30  | 2.90      | 2.90      |
|                 | Baywet              | 128 | 2137| 0.23  | 0         | 0         |
|                 | Gramdry             | 69  | 915 | 0.30  | 2.90      | 2.90      |
| Social          | Cons-frequency-rev  | 46  | 879 | 0.04  | 0         | 0         |
|                 | Manuf-frequency-rev | 77  | 2228| 0.01  | 0         | 0         |
| Transcription   | Ecoli               | 419 | 519 | 0.75  | 0.48      | 3.10      |
|                 | Yeast               | 688 | 1079| 0.82  | 2.76      | 0.87      |
|                 | ColdInterFullVec    | 424 | 577 | 0.73  | 0.24      | 6.13      |
| Power Grid      | Dallas              | 4941| 13188| 0.12   | 5.75      | 6.64      |
| Cortical        | Macaque cortical    | 1168| 2486| 0.04  | 3.42      | 3.42      |
| Neuronal        | C. elegans-1        | 131 | 764 | 0.09  | 0.76      | 0         |
|                 | C. elegans-2        | 277 | 2105| 0.12  | 1.08      | 0.72      |
| Cellular        | AA                  | 1485| 3400| 0.29  | 10.71     | 9.90      |
|                 | BB                  | 804 | 1674| 0.27  | 8.33      | 7.71      |
|                 | EF                  | 1407| 3290| 0.31  | 10.95     | 10.73     |
|                 | PA                  | 2554| 6080| 0.32  | 11.51     | 10.14     |

Table 1. The properties of the real network analysed. For each network, we show its types, name, number of nodes (N), edges (L), fraction of driver nodes ($n_D$), minimum proportion of new source and sink nodes can be augmented in parallel ($n_{SIC}$ and $n_{KIC}$). Note that $n_{SIC}$ and $n_{KIC}$ are shown in percentage.
matching of the bipartite graph with Hopcroft-Karp algorithm\cite{13,34}. MDS and MTS are unmatched nodes in the \( - \) set and \( + \) set respectively. The number of computational steps needed to find one maximum matching can be as small as \( O(N^{3/2}) \) with the Hopcroft-Karp algorithm, where \( N \) and \( L \) represent the number of nodes and edges in the network respectively.

**Identifying SI and KI clusters.** Nodes in a directed network \( G(V, E) \) should be classified into two sets: SI and SR. To expedite finding SI clusters it is helpful to first form the corresponding SI network defined as the remaining connected sub-graphs found after removing all incoming edges to all SR nodes. The identification of SI clusters is then achieved through a sequential process of searching for V-motifs. The search begins with low out-degree nodes and ends with nodes having highest out-degree. The process makes use of cluster merging. Two clusters (sets of nodes) can be merged into one if: i) the two clusters appear as two distinct leaves of a V-motif; ii) the two clusters contain one or more common nodes. The detailed procedure is as follows:

1. Search the SI network for all nodes having out-degree equal two \( (k_{\text{out}} = 2) \). If there are no nodes with \( k_{\text{out}} = 2 \) in the SI network there will be no clusters. Each identified node is the root node of a prototypical V-motif, and the two nodes connected to the root are leaves. Being part of the SI network, they must both be SI. From here on, treat both of these SI nodes as a merged cluster. Multiple clusters are merged if they meet the above merging conditions i) or ii). For example, from the V-motifs in Fig. 3c, the sets of SI nodes \{14, 1\}, \{1, 5\} and \{5, 7\} are considered as three clusters (merge condition i). The clusters can be further merged together as \{1, 5, 7, 14\} (merge condition ii).

2. Search for all nodes with out-degree equal to three. First, each of these root nodes has three leaves needs to be merged into clusters where possible. A V-motif is formed if each root node points to two and only two distinct clusters, e.g. In Supplementary Fig. S2a, \( R \rightarrow \{L_1, L_2, L_3\} \) is a V-motif, since \( L_1 \) and \( L_2 \) belongs to the same cluster. Finally, all clusters identified should be merged where possible before move to the next step.

3. Repeat procedure (2) iteratively on nodes after increasing \( k_{\text{out}} \) by one each step. The process terminates when either no V-motifs can be found, or in the worst case terminated on nodes with highest possible \( k_{\text{out}} \) value. Finally, the desired SI clusters are those clusters which contain more than one nodes, one of which must be a driver node. Distinct SI clusters do not share common nodes.

4. Due to the complexity of the network, the first implementation of the above procedure will not identify all possible SI clusters in the network. However, the remaining SI clusters can be identified by eliminating the existing SI clusters from the SI network (delete all nodes in the SI clusters along with their edges) and repeat procedures (1)–(3) until no further V-motifs can be found. This will eventually reveal all possible SI clusters.

The identification of KI clusters follows the same procedures except that it is necessary to begin with the KI network which is determined by removing incoming edges to KR nodes. A standard V-motif in the KI network is defined as one leaf node pointed to by two KI root nodes (Supplementary Fig. S4). Similarly, KI cluster identification follows the same procedures except searching begins with low in-degree \( (k_{\text{in}} = 2) \) nodes. The conditions of two clusters merge together in sources augmentation holds for sinks augmentation as well. SI (or KI) clusters determination is based on the degree of individual nodes in the SI (or KI) network. Thus, the complexity is proportional to number of nodes and edges in the SI (or KI) network. Therefore, the complexity of the algorithm is linearly proportional to \( N \) and \( L \) in the network.

**References**

1. Boguná, M., Papadopoulos, F. & Krioukov, D. Sustaining the internet with hyperbolic mapping. Nat. Commun. 1, 62 (2010).
2. Scanfeld, D., Scanfeld, V. & Larson, E. L. Dissemination of health information through social networks: Twitter and antibiotics. *Am. J. Infect. Control* **38**, 182–188 (2010).
3. Enquist, P., Guerraoui, R., Kermarrec, A. M. & Massoulié, L. Epidemic information dissemination in distributed systems. *Computer* **37**, 60–67 (2004).
4. Galbiati, M., Delpini, D. & Battiston, S. The power to control. *Nat. Phys.* **9**, 126–128 (2013).
5. Benaroch, M. & Kaufman, R. J. Justifying electronic banking network expansion using real options analysis. *MIS Quart* **24**, 197–225 (2000).
6. Da Silva, E. L., Gil, H. A. & Areiza, J. M. Transmission network expansion planning under an improved genetic algorithm. *IEEE Trans. Power Syst.* **15**, 1168–1174 (2000).
7. Kitsak, M. et al. Identification of influential spreaders in complex networks. *Nat. Phys.* **6**, 888–893 (2010).
8. Van Mieghem, P., Oncim, I. & Kooij, R. Virus spread in networks. *IEEE/ACM Trans. Netw.* **17**, 1–14 (2009).
9. Newman, M. E., Forrest, S. & Balthrop, J. Email networks and the spread of computer viruses. *Phys. Rev. E* **66**, 035101 (2002).
10. Clarke, R. A. & Knake, R. K. Cyber war: The next threat to national security and what to do about it (Tantor Audio, 2014).
11. Sun, S., Zhang, X., Tough, D. F. & Sprent, J. Type I interferon-mediated stimulation of T cells by CpG DNA. *J. Exp. Med.* **188**, 2335–2342 (1998).
12. Liu, Y.Y., Slotine, J. J. & Barabási, A. L. Controllability of complex networks. *Nature* **473**, 167–173 (2011).
13. Yan, G., Ren, J., Lai, Y. C., Lai, C. H. & Li, B. Controlling complex networks: How much energy is needed? *Phys. Rev. Lett.* **108**, 218703 (2012).
14. Neperuz, T. & Viceche, T. Controlling edge dynamics in complex networks. *Nat. Phys.* **8**, 568–573 (2012).
15. Ifs, J. & Ruths, D. Control profiles of complex networks. *Science* **343**, 1373–1376 (2014).
16. Yuan, Z., Zhao, C, Di, Z., Wang, W. X. & Lai, Y. C. Exact controllability of complex networks. *Nat. Commun.* **4**, 2447 (2013).
17. Gao, J., Liu, Y. Y., D’Souza, R. M. & Barabási, A. L. Target control of complex networks. *Nat. Commun.* **5**, 3415 (2014).
18. Plosai, M., Liu, Y. Y., Slotine, J. J. & Barabási, A. L. Effect of correlations on network controllability. *Sci. Rep.* **3**, 1067 (2013).
19. Jia, T. et al. Emergence of bimodality in controlling complex networks. *Nat. Commun.* **4**, 2002 (2013).
20. Kalman, R. E. Mathematical description of linear dynamical systems. *J. Soc. Ind. Appl. Math. Ser. A* **1**, 152–192 (1963).
21. Slotine, J. J. E. & Li, W. Applied nonlinear control Vol. 199 (Prentice-hall, 1991).
22. Lin, C. T. Structural controllability. *IEEE Trans. Autom. Control* **19**, 201–208 (1974).
23. Shields, R. W. & Pearson, J. B. Structural controllability of multi-input linear systems. *Rice University ECE Technical Report* (1975).
24. Murota, K. Matrices and matroids for systems analysis Vol. 20 (Springer Science & Business Media, 2009).
25. Lovász, L. & Plummer, M. D. *Matching theory* Vol. 367 (American Mathematical Society, 2009).
26. Jia, T. & Barabási, A. L. Control capacity and a random sampling method in exploring controllability of complex networks. *Sci. Rep.* 3, 2354 (2013).
27. Wang, W. X., Ni, X., Lai, Y. C. & Grebogi, C. Optimizing controllability of complex networks by minimum structural perturbations. *Phys. Rev. E* 85, 026115 (2012).
28. Xiao, Y. D., Lao, S. Y., Hou, L. L. & Bai, L. Edge orientation for optimizing controllability of complex networks. *Phys. Rev. E* 90, 042804 (2014).
29. Hou, L., Lao, S., Small, M. & Xiao, Y. Enhancing complex network controllability by minimum link direction reversal. *Phys. Lett. A* 379, 1321–1325 (2015).
30. Kim, Y., Choi, T. Y., Yan, T. & Dooley, K. Structural investigation of supply networks: A social network analysis approach. *J Oper Manag* 29, 194–211 (2011).
31. Barabási, A. L. & Albert, R. Emergence of scaling in random networks. *Science* 286, 509–512 (1999).
32. Strogatz, S. H. Exploring complex networks. *Nature* 410, 268–276 (2001).
33. Erdös, P. & Rényi, A. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.* 5, 17–61 (1960).
34. Newman, M. E. Assortative mixing in networks. *Phys. Rev. Lett.* 89, 208701 (2002).
35. Pastor-Satorras, R., Vázquez, A. & Vespignani, A. Dynamical and correlation properties of the internet. *Phys. Rev. Lett.* 87, 258701 (2001).
36. Hopcroft, J. E. & Karp, R. M. An n5/2 algorithm for maximum matchings in bipartite graphs. *SIAM J Comput* 2, 225–231 (1973).

**Author Contributions**
J.W., X.Y.and L.S. designed the research. J.W. performed the numerical simulation and analysed empirical data. All authors reviewed the manuscript.

**Additional Information**
Supplementary information accompanies this paper at http://www.nature.com/srep

**Competing financial interests:** The authors declare no competing financial interests.

**How to cite this article:** Wang, J. *et al.* Effective Augmentation of Complex Networks. *Sci. Rep.* 6, 25627; doi: 10.1038/srep25627 (2016).

This work is licensed under a Creative Commons Attribution 4.0 International License. The images or other third party material in this article are included in the article’s Creative Commons license, unless indicated otherwise in the credit line; if the material is not included under the Creative Commons license, users will need to obtain permission from the license holder to reproduce the material. To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/