OIKOS 110: 164–176, 2005

Ecological subsystems via graph theory: the role of strongly connected components

Stefano Allesina, Antonio Bodini and Cristina Bondavalli

Allesina, S., Bodini, A. and Bondavalli, C. 2005. Ecological subsystems via graph theory: the role of strongly connected components. – Oikos 110: 164–176.

In this paper we investigate ecological flow networks via graph theory in search of the real sequential chains through which energy passes from producers to consumers in complex food webs. We obtain such fundamental pathways by identifying strongly connected components (SCCs), subsystems that groups species that take part in cycling, and performing topological sorting on the acyclic graphs that are obtained. Topological sorting identifies preferential directions for energy to flow from sources to sinks, while recycling remains confined within each SCC. Resolving food web networks for SCC highlights the possibility that compartments can be found in ecosystems, but this does not seem a general rule. The four aquatic food webs described in detail show a rather clear subdivision between benthic and pelagic subcommunities, a result that is discussed in the light of other studies. Should further research confirm these results, new insight into the way ecosystems use energy will be provided, with implications on cycling, reciprocal dependency of variables and indirect effects.

S. Allesina, A. Bodini, and C. Bondavalli, Dept of Environmental Sciences, Univ. of Parma, Italy (sallesina@nemo.unipr.it). Present address for SA: Dept of Fisheries and Wildlife, Michigan State Univ., 13 Natural Resources Building, East Lansing, MI 8824, USA (allesina@msu.edu).

Food webs are extraordinarily complex. This complexity, no matter how inconvenient to ecological theory, must be investigated for a grasp of ecosystem behavior (Schmitz 1997, Bodini 2000). Understanding ecological systems means essentially unveiling patterns that are hidden in their complex architectures. One possibility to explore ecological dynamics is to use energy as a currency: in this case food webs take the form of flow networks (Ulanowicz 1986, Christian and Luczkovich 1999, Heymans et al. 2002).

As for energy transfers, every ecosystem can be reduced to a Lindeman's tropho-dynamic sequence of discrete trophic levels to which species can be apportioned in proportion to their diet (Burns et al. 1991, Ulanowicz 1995). However, this heuristically useful model does not represent a sequential chain of energy flows in the strict sense. On the other hand, because of the reticulate connections between a diversity of consumers and resources that make up the food web, it seems that, virtually, no constraints, but thermodynamic ones, regulate the energy flow in the food web (Polis and Strong 1996). According to this "multichannel" view of the ecosystem, a great deal of opportunity would arise for species to exchange energy with the other components, no matter where and how far apart they are in the web.

Understanding whether fundamental pathways for energy flow exist is of great importance, as it may shed light on movements of mass and energy in ecosystems, and, in turn, on the nature of cycles, reciprocal dependency of species and indirect effects, that are crucial for ecosystem functioning (De Angelis 1992,

Accepted 16 December 2004

ISSN 0030-1299

Wootton 2002). In search for these pathways, the object of interest in this study, we applied a simple analysis based on graph theory on a set of published and unpublished ecological networks. In particular, the method makes use of the concept of strongly connected component (Skiena 1990, Read and Wilson 1998). Applied to ecological food webs, this concept refers to subsets of species for which energy can flow from one another and back. Looking for strongly connected components in a series of ecological flow networks, we identified the trophic hierarchy from producers to consumers and found that ecosystems may be comprised of subsystems that are sequentially connected by linear chains of energy transfers. At the same time, however, the possibility that strongly connected components define subsystems within food webs depends on the range of magnitude of links (flows).

Methods

Network models

Understanding fundamental passages of energy within the ecosystem requires first that one knows who eats whom and by how much. In this way the amount of energy that flows from one species to another can be assessed. Once this task is accomplished all this information can be condensed in a graph-like model that provides an overall perspective of the trophic exchanges between species. In such pictorial representation species or aggregations of species (trophospecies), nutrient pools, and detrital compartments are depicted as nodes that are connected by arrows that represent energy exchanges. These latter are the outcome of the trophic interactions that are established between the trophospecies. In general, these flows are measured as grams of carbon per square meter per year (gC y^{-1} m^{-2}). We analyzed a series of ecosystems as network models (all the systems, with respective references, can be found in a single file for download at Robert Ulanowicz's home page http://cbl.umces.edu/ ~ulan/). Because in any ecosystem there are components that exchange matter and energy with the outside environment and also all the living species dissipate a fraction of energy, in the network models we analyzed we added three extra-nodes to take into account these functions. These three special nodes are: a) the input node, a node that stands for the outside environment as provider of mater and energy to the ecosystem and from which arrows reach those species that in the real system import matter and energy from the outside; b) the output node, that represents the outer environment as sink of all usable exports; and c) the dissipation node, that is the sink of the energy that is dissipated by all the living components.

Strongly connected components

In an ecosystem the energy flows from producers to the consumers of various level. Such directional flowing can be appreciated in the graph-like model by following the direction of the arrows that connect the nodes. By ideally moving along these arrows we can connect, in principle, any two nodes that are far apart from one another several trophic steps. Whenever one such connection exists, the combination of arrows that form it, is called "path". A "cycle" is a path that originates from one species and returns to it. Strongly connected components individuate the sets of trophospecies (vertices in the network) that are reachable from each other, that is to say the subset of species that are connected by cycles. In each strongly connected component, for any two nodes we can always find a path that goes from one to the other and the way back. In an ecosystem there may be one or more strongly connected components (SCCs). In the former case it is always possible to move from one species to any other in the whole system. When there are more than one strongly connected component, energy flows freely only within the SCC. Every strongly connected component does not share any node with the others, which means that SCCs form disjoint subsets of vertices.

Cycle search usually requires a large amount of time, as the number of operations to be performed (Mateti and Deo 1976, Ulanowicz 1983) increases more than exponentially with the number of vertices in the network. On the contrary searching for SCCs requires linear time (the number of operations grows linearly with the number of vertices and edges, Tarjan 1972). The simplest algorithm that performs SCCs search is the one introduced by Hopcroft and Tarjan (1973); it is explained in detail in Appendix 1.

The number of vertices S_i that belong to a certain component defines its size $|S_i|$. If the size is one (the node/species does not participate to any cycle), there are three possibilities: a) it is a source, that is a species that receives energy/matter from the outside and passes it to the next in the web without receiving anything from inside the system; b) the species/node is a sink, in which the matter/energy enters and from there it may be only exported/dissipated; c) the species is a cross-vertex that connects two SCCs without participating to any cycle.

If the size of a strongly connected component is equal to the total number of nodes, the graph is said to be strongly connected (Bang-Jensen and Gutin 2000). In this case cycling is possible between any pair of species.

If we imagine to represent any strongly connected component that exists in a certain network by a unique node, we obtain a simplified version of the ecosystem network that describes the linear flow of energy from producers to consumers, while cycling remains confined within each SCC. This simplified network is called directed acyclic graph (DAG). Because a DAG com-

prises only one-way flows, it is a pictorial representation of a hierarchy of dependencies involving groups of species.

Topological sorting

According to the hierarchical nature of DAGs, a procedure called topological sorting can be applied so that one can visually order its components (that is the SCCs) according to precedence constraints. In practice, doing topological sorting one orders the various SCCs from left to right. So that in this linear flowing any SCC is preceded by the component that provides energy to it and followed by the component that receives energy from it. Topological sorting can be achieved in linear time (Knuth 1997). In the framework of ecological flow networks, it makes evident the fundamental linear pathways through which energy and matter travel in the ecosystem. In other words, topological sort produces a chain of donors (such as resources) and consumers that makes up the sequential chain of energy flow. Accordingly, a topological sort must include at least a node with no incoming edges (source) and one with no outgoing edges (sink). This may provide a possible taxonomy of functional subsystems that govern energy transfer in the ecosystem.

The methods presented above provide an easy and fast way to a) individuate group of species that are strongly linked with each other, sharing cycles that augment the residence time of the matter inside the subsystem; b) produce a simple structure that accounts of the linear passage of energy matter between those subsystem. Appendix 1 provides a more rigorous description of the procedures.

Results

We examined 17 systems and found that each of them was composed by at least 4 SCCs, namely {input}{output}{dissipation}{species and nutrient pools}. The "special compartments" (Methods) are in fact SCCs because {input} is a source and {output} and {dissipation} are two sinks for carbon. The Ythan Estuary network (Baird and Milne 1981) matched with this elementary scheme, with all the species grouped in one SCC. All the other models included at least one further SCC of size 1. This means that at least one compartment is not strongly connected with the rest of the system. In fact, for the other 16 networks, the bulk of {species and nutrient pools} appears made of several "internal" SCCs. In most networks there are various SCCs of size 1 plus one very large SCC that comprises all the other compartments. In four aquatic ecosystems, however, we found two internal SCCs of size >1. They are the "Chesapeake Bay" (Baird and Ulanowicz 1989), the "Baltic Sea aggregated" (Wulff and Ulanowicz 1989), the "Charca de Maspalomas" (Alumnia et al. 1999) and the "Upper Chesapeake" (Hagy 2002). Table 1 describes the composition of each SCC in these four networks.

Using the standard aggregation technique for networks (Ulanowicz and Kemp 1979) and removing self-loops created during this process, every SCC was transformed into a single compartment. In this way, a DAG for every network was obtained. Performing topological sort on each DAG yielded a sequential chain of energy flow. Using the Baltic Sea network as an example, the entire process is pictorially represented in Fig. 1. In it, one sees the network as it was originally constructed (Fig. 1, upper) and its corresponding DAG ordered according to topological sort (Fig. 1, lower). Figure 2 shows the structure of the DAGs obtained from the other networks listed in Table 1.

None of the 17 networks considered in this study comprised only SCCs of size 1. In many of them at least one very large SCC was identified. This is obvious because a network made by only SCCs of size 1 does not have cycles (acyclic network), and matter, once entered in the system, would exit without being circulated, an ecologically unrealistic situation. In the Ythan Esturay the whole internal system participates to cycling, as there is a unique SCC grouping all living and non-living components. In the four networks listed in Table 1, internal SCCs of size greater than 1 separate benthic from pelagic species. That is cycling does not involve at the same time benthic and pelagic species. In three out of the four cases, fish that do not act as single SCC are associated with the benthic community. Only in the Upper Chesapeake network fish species contribute to cycling in the pelagic compartment.

In all networks the producers appear as SCC of size 1; this depends on the fact that carbon is the currency used for the networks. In the Baltic Sea (Table 1, Fig. 1) there are pelagic producers and benthic producers. The other SCCs of size 1 are compartment #7, benthic suspension feeders and #13, dissolved organic carbon. The former receives energy from the pelagic SCC and cedes it to the benthic SCC, while the latter receives energy from pelagic producers (compartment #1) and transfers it to the pelagic SCC. Both these variables are cross-vertices for the system. The number of these SCCs in the four networks changes depending on the level of resolution adopted by the authors who parsed the networks. In the Charca and Upper Chesapeake ecosystems most of these SCCs are producers; on the contrary in the Chesapeake Bay most of the SCC of size 1 are cross-vertex components.

In all these networks pelagic and benthic SCCs include at least a non-living compartment; in the Baltic Sea it is suspended POC for the pelagic subsystem and sediment POC for the benthic subsystem. Most part of the matter

Table 1. Species composition for the internal SCCs identified in the four aquatic networks with two internal SCC of size greater than 1. The right column embeds all the single-species SCCs. Names of compartments are those used by the authors who built up the networks.

Pelagic	Benthic+fish	Size 1 SCCs
Chesapeake Bay		
Zooplankton; ctenophores; *Chrysaora fuscescens* (sea nettle); suspended POC; suspended bacteria; POC; ciliates; DOC	Other polychaetes; *Macoma* spp.; *Meiofauna*; crustacean deposit feeder; *Micropogonias* spp. (croaker); *Trinectes maculatus* (hogchoker); spot; *Morone americana* (white perch); *Arius felis* (hardhead catfish); *Pomatomus saltatrix* (bluefish); *Paralichthys dentatus* (summer flounder), *Morone saxatilis* (striped bass); sediment POC; bacteria in sediment POC; *Nereis succinea* (polichaetes); *Callinectes sapidus* (blue crab)	1. Phytoplankton; 4. Benthic diatoms; 5. Free bacteria; 6. Heterotrophic microflagell.; 7. Other suspended feeders 8. *Mya arenaria*; 9. *Crassostrea* spp. (oysters); 12. *Anchoa mitchilli* (bay anchovy); 13. *Brevoortia tyrannus* (menhaden); 15. *Cynoscion regalis* (weakfish); 14. *Alosa sapidissima* (American shad); 11. *Alosa pseudoharengus* (aalewife) & *Coregonus artedi* (blue herring); 10. Fish larvae
Baltic Sea aggregated		
3. Pelagic bacteria; 4. Microzooplankton 5. Invertebrate carnivores; 6. Mesozooplankton; 14. Suspended POC;	8. Meiofauna; 9. Deposit feeders; 10. Benthic invertebrates carnivores; 11. Planktivorous fish; 12. Carnivorous fish; 15. Sediment POC	1. Pelagic producers; 2. Benthic producers; 7. Benthic suspension feeders; 13. DOC
Charca de Maspalomas		
Mesozooplankton; suspended POC; DOC; pelagic bacteria; microzooplankton	Benthic invertebrate carnivores; *Liza aurata*; *Dicentratus punctatus*; sediment POC; benthic microfauna; benthic deposit feeders; *Diplodus sargus*	1. Cyanobacteria; 2. Eukaryotic phytoplankton; 3. *Chara globularis*; 4. *Ruppia maritima*; 5. *Cladophora* spp.; 6. Periphyton; 8. Macrozooplankton; 10 Benthic suspension feeder; 11. *Gallinula chloropus*
Upper Chesapeake		
Pelagic+Fish	Benthic	Size 1 SCCs
Heteroflagellates; rotifers; meroplankton; mesozooplankton; *Anchoa mitchilli* (bay anchovy); ctenophores; cuspension feeding benthos; *Callinectes sapidus* (blue crab); *Brevoortia tyrannus* (enhaden); *Alosa aeastivalis* (herring) and *Alosa sapidissima* (shad); *Morone americana* (white perch); spot; *Micropogonias* spp. (croaker); *Trinectes maculatus* (hogchoker); *Anguilla rostrata* (American eel); *Ictalurus melas* (catfish); *Morone saxatilis* (striped bass); *Pomatomus saltatrix* (bluefish); *Cynoscion regalis* (weakfish); suspended POC; particle attached bacteria; DOC; free bacteria; ciliates; *Crassostrea* spp. (oysters)	Benthic bacteria; meiofauna; deposit feeding benthos; sediment POC	1. Net phytoplankton; 4. *Chrysaora* spp. (sea nettle); 5. Microphytobenthos; 6. SAV; 7. Picoplankton

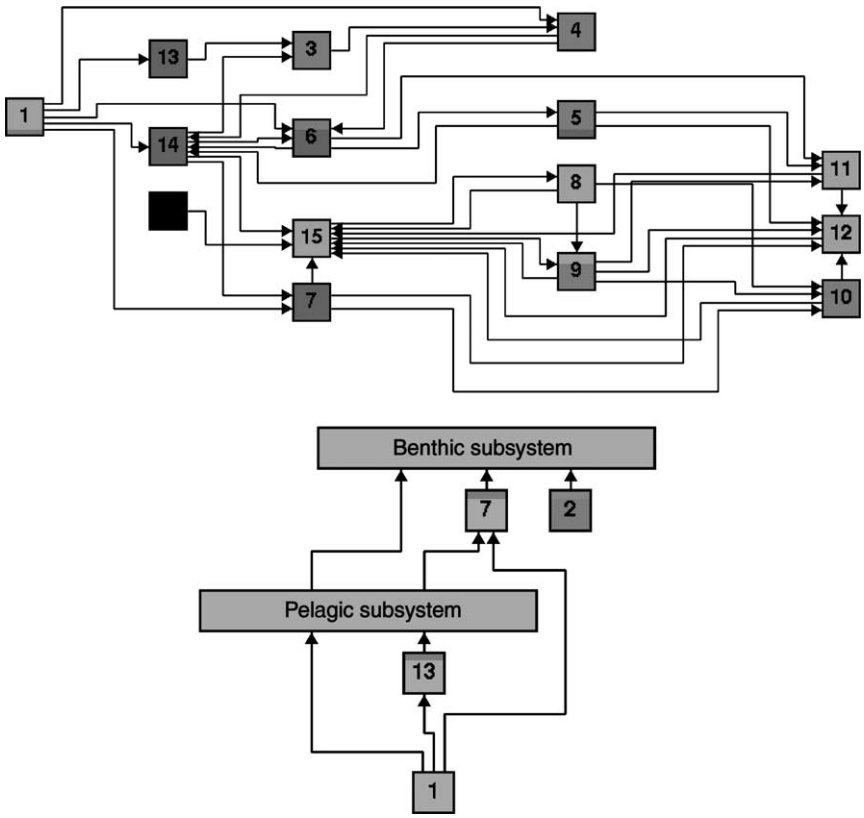

Fig. 1. Original network for the Baltic Sea ecosystem (upper graph) and its topologically sorted DAG (bottom-up perspective, lower graph). Correspondence between numbers and names of compartments is given in Table 1.

that is recycled in the subsystem passes through these compartments.

Once a DAG is obtained one has a perspective of the linear flow of energy that characterizes the ecosystem. Figure 1 and 2 help to explain this concept as they show the topologically sorted DAG obtained from the four networks. In particular, Fig. 1 compares the original network of the Baltic Ecosystem and its corresponding DAG (Fig. 1, lower graph). What is striking in this graph is that the pelagic subsystem supplies energy to the benthic subsystem, which acts as internal sink for the system. Energy flows from the pelagic compartments to the benthic compartments but not the other way around. This ordered sequence of energy transfer occurs also in the Chesapeake Bay and the Charca de Maspalomas ecosystem (Fig. 2 upper and middle graphs).

In the Upper Chesapeake (Fig. 2, lower graph) it is the pelagic compartment that receives energy from the benthic subsystem, but it does not act as a sink. It behaves like a cross-vertex for *Chrysaora* spp., the unique sink for this system. The Upper Chesapeake may seem anomalous if compared with the other three networks also because fish, usually at the top of the food chain, belong to the pelagic SCC, that is they recycle matter within the pelagic compartment. In all the other cases top predators recycle within the benthic compartment.

These results yield for a qualitative (presence/absence) description of trophic relations. Introducing quantities would ensure the prospect of a robustness test. In fact, at every passage between compartments, the energy/matter is diminished according to thermodynamics, meaning that recycling is based on weak connections, while linear transfers could involve strong relations. We tested the robustness of our results by considering the magnitude of links and setting up a filter to determine the presence/absence of edges. In particular, the filter was based on the total amount of matter exiting every compartment: dividing the weights of every outgoing edge by the sum of all outgoing edges, we obtain their fractional importance.

We sequentially increased the threshold from e^{-20} to e^{0} and counted the number of SCCs produced considering in each run all the edges that had a fractional importance higher than the threshold and pruning away the others. The number of SCCs is upper bounded by the number of compartments, as in an acyclic network the number of compartments equals the number of SCCs. We reported the fraction of edges removed together with the number of SCCs. Only results for the Chesapeake Bay are given here as the main conclusions can be applied to any of the system investigated. Figure 3 shows the number of SCC and the percentage of remaining edges as a function of the threshold of magnitude for the links.

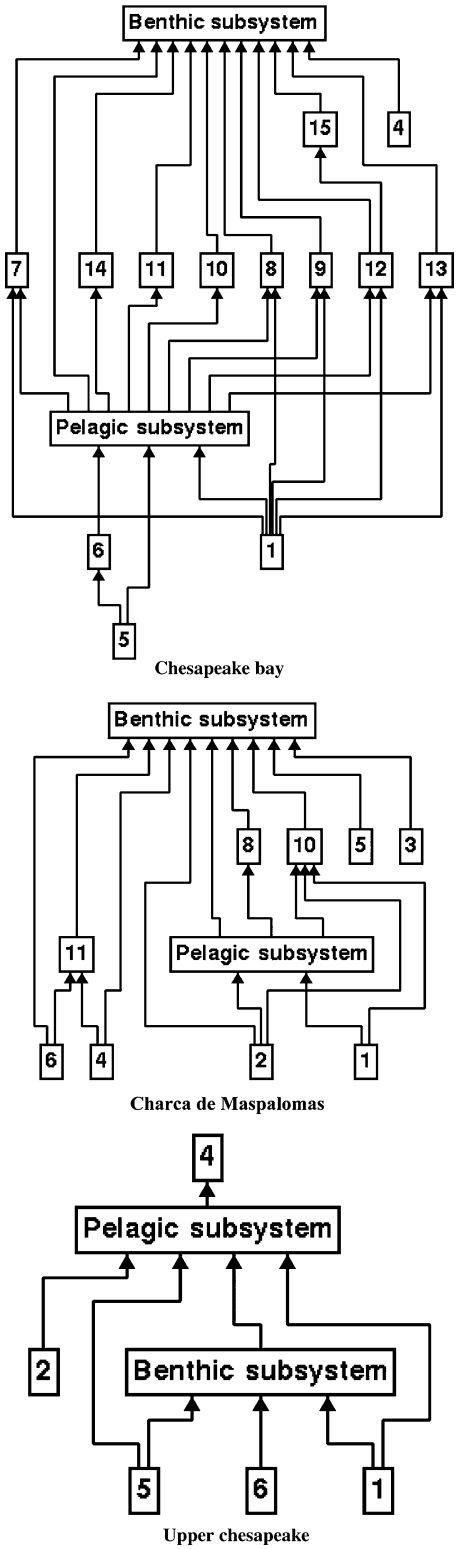

Fig. 2. Topologically sorted DAGs (bottom-up perspective, lower graph) of three aquatic networks. Correspondence between numbers and names of compartments is given in Table 1.

Our findings show how cycles are very sensitive to weak edges removal. In Chesapeake Bay the edges removal starts with the threshold $e^{-14.1}=7.52E-07$, when just an edge is eliminated. The number of SCCs starts increasing with threshold $e^{-11.9}=6.79E-06$ (7 edges out of 177 removed), rising up exponentially until $e^{-1.6}=2.02E-01$ when only 31.6% of the edges are still present. Starting from this threshold all the compartments become SCCs, making the network acyclic.

The fact that number of cycles/SCCs increases exponentially with the number of edges removed is not surprising, but points out, once more, how food web modeling is prone to sampling effort (we may expect that weak edges individuation would require more sampling effort than greater ones, Martinez et al. 1999), and how the elimination of a small amount of trophic exchanges, that may occur, for example, when the niche of a species is exploited by another one with similar diet, could completely reshape the cycling behavior of the ecosystem.

Defining a threshold of magnitude to decide which links appear in a food web is crucial. In the 4 networks that showed a benthic/pelagic division, such a compartmentalization vanishes when resuspension is included and the two subgroups melt down into a unique big group. However the magnitude of these links is so small that their functional role is certainly less crucial than re-suspension in shallow waters. For a more thorough discussion on the importance of trophic links in food webs see Raffaelli and Hall (1995).

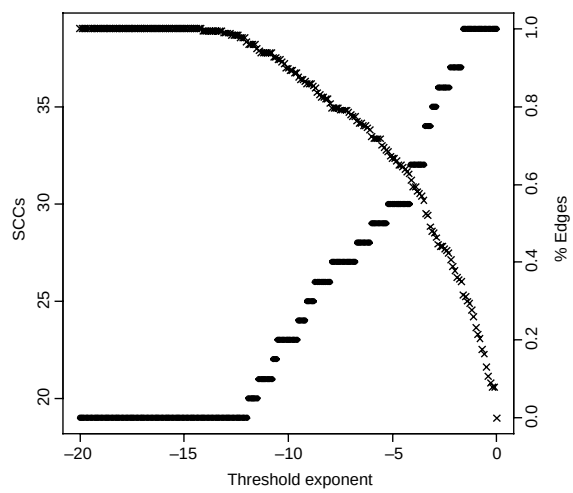

Fig. 3. Results of threshold based edges removal. Horizontal axis represents the threshold's exponent λ (threshold $=e^{\lambda}$); left vertical axis accounts for the number of SCCs produced once the weak edges have been pruned, right vertical axis for the percentage of edges remaining.

Discussion

Resolving a number of ecological networks for their strongly connected components and performing topological sort on the associated direct acyclic graph has shown that, although ecosystems comprise myriad interaction links, they can form subsystems that are sequentially connected by few linear chains of energy transfers. This outcome immediately recalls to the problem of whether or not food webs are divided into compartments. May's (1972) conjecture that food webs are compartmented because blocking would enhance stability has been challenged on the empirical ground. Pimm and Lawton (1980), examining some real cases, found no evidence that webs are arranged into blocks based on dynamical constraints. Considering trophic similarity as the degree to which species share predators and prey, they found that probabilities of compartmentalization, averaged across all the food webs, were not statistically significant. Raffaelli and Hall (1992) focused on the same webs separately and found, without the possibility to generalize their outcomes in a statistical sense, that in several of those webs probability of compartmentalization is high. In both these works food webs were represented as undirected graphs in which links described of feeding (predator–prey) relationships.

Using directed graphs (food web graphs, sensu Cohen 1978), Yodzis (1982) was able to decompose food webs into compartments using the "clique" concept, that is grouping sets of species in which every pair had some food resource in common. Dominant cliques are formal structural objects that are uniquely identified by a rigorous procedure, and help understanding how food webs are organized in relation to the niche concept (Cohen 1978). Strongly connected components, that also are formal structural objects, on the other hand reveal how food webs are organized around essential flows of energy. To this end using cliques would not be possible because they share species between one another.

Not always strongly connected components divide the food web into compartments that group more than one species. In the Ythan Estuary, for example, species belong to a unique SCC, that is this ecosystem is not compartmented, a result that matches with Raffaelli and Hall's (1992) conclusion based on trophic similarities. Most of the 17 food webs we examined did not show clear compartmentalization. This suggests that answering the original question, posed in terms of a dichotomous choice, "are food webs divided into compartments?" does not add that much because, likely, food webs behave differently from one another in this respect. More interesting questions would be "in what circumstances we observe compartmentalization in a system or a certain class of systems?" and "is there any pattern that describe compartmentalization, when it exists, in food webs?" We reiterate, however, that compartments emerged as a result of searching for SCC, a procedure necessary to identify the trophic hierarchy in the food webs.

The four ecosystems described in this paper show a clear subdivision between pelagic and benthic subcommunities. For the Chesapeake Bay the same result was obtained by Girvan and Newman (2002), who used new approach to clustering based on the concept of "edge betweeness" applied to the network graph and by Krause et al. (2003), who searched for cohesive subgroups within the food web structure. In the former case, however, the study was conducted using weighted networks, that is networks in which links are quantified, while the latter utilized both quantitative both qualitative data. Krause et al. (2003) did not find compartments when they applied their method to the unweighted network, and they drew the conclusion that the method seems unable to identify compartments in unweighted networks.

Basically in both these works the main focus is on density or concentration of interaction. In our study cycling is the main feature that eventually determines the appearance of subgroups. Also, and more important, the two approaches lead to different results. In fact Krause and colleagues identify two compartments, whereas we have two rather large SCC and a series of single species SCC. Girvan and Newman (2002) too, identify "undetermined" species, but they were less than the number of single species-SCC we found. Second and most important there is no exact correspondence of species between our groups and theirs. This suggests that the "social" position that species or guilds occupy in ecosystem and their functional role as distributors of medium are linked in a complicated way that is largely unknown and not easy to understand.

SCCs delineate how multiple reticulate connections typical of food webs aggregate in essential linear pathways. When there is a unique SCC, as in the case of Ythan Estuary, energy is free to cycle in the ecosystem. Whenever there are SCCs of size greater than 1, only exchanges between the SCCs are not totally free. They are constrained by the flow patterns from sources to sinks. So in the Baltic Sea, Chesapeake Bay and Charca de Maspalomas benthic components do not supply energy to pelagic ones, whereas the opposite occurs in the Upper Chesapeake. Also many of the size 1-SCC do not interact each other and determine parallel flow of energy from sources to sinks. These results suggest that impacts that can variously affect energy flows (e.g. collapse of a prey population, variations in the primary production) likely spread in the ecosystem according to the flow pattern identified by the structure of SCC.

While experimental evidences may confirm or discard this hypothesis, it must be reiterated that this study cannot say anything about the distance up to which

impacts propagate. This question has been explored both empirically (Abrams et al. 1995, Menge 1995) and theoretically (Yodzis 2000, Williams et al. 2002). Williams et al., in particular, using the mean distance between all nodes in a set of seven food webs (among which the Chesapeake Bay and the Ythan Estuary discussed here) found that the vast majority of species are not isolated so that impacts may affects more species than is often appreciated. In their research Williams et al. used undirected networks and in so doing they could not explore the problem of directions of impacts. On the other hand our results suggest that impacts cannot spread in any direction but only according to the flow patterns established by the SCCs and their sorting. Rather than seeing the two approaches in contrast with one another we perceive the potential for their integration.

The fact that SCCs impose constraints to energy flows does not discard the intrinsic complexity of food webs in favor of a trophic level-based view of nature (Polis 1991, Polis and Strong 1996). However, the results obtained by focusing on SCCs need to be accommodated in the framework of food web theory, which is not an easy task because SCCs neither contribute to define trophic levels (within the same SCC we can find primary producers, omnivores and predators as well) not their topological sorting separate the grazing and the alimentary chains (e.g. detrital components are present both in the pelagic and benthic SCC in the 4 selected ecosystems).

The presence of many SCCs of size equal to 1 implies that a high number of compartments do not take part in cycling. Since cycling activity defines ecosystem maturity (Odum 1969, Ulanowicz 1996) and may be used to identify conditions of stress for the ecosystem (Odum 1985, Ulanowicz 1996) these SCCs could be used as an additional feature to describe ecosystem health. To this end, however, a more robust relationship linking SCCs structure with ecosystem properties must be found. SCCs acting as cross-vertices (in the Chesapeake Bay, they are the filter feeding communities) may also play an important role in the distribution of energy. This depends on the amount of currency that is channeled through them and their position in relation to the other SCCs; a quantitative study based on flow values and energy budget could shed light on this point.

SCCs analysis could be a useful approach to integrate the standard procedures of network analysis. Strongly connected components are "de facto" linearly coupled subsystems. In this respect they might be treated as independent networks for which the rest of the system can be apportioned to the input and output environment. Accordingly, one could perform network analysis on every SCC, unveiling, with respect to the whole ecosystem, its role in the circulation of matter and energy, its contribution to dependencies between compartments, and its contribution to create causal chains of indirect effects.

As a final point we stress that the results presented here are currency-sensitive; that is, nutrient flow networks (i.e. N and P) likely would have different SCCs, leading to different architecture for the DAGs. In particular, we expect that the conclusion that recycling remains confined within each SCC would be challenged in nutrient flow networks. These issues that will be discussed in detail elsewhere.

Conclusion

In this work we analyzed ecological flow networks searching for strongly connected components and used a series of models described in the literature. Being a network a model of an ecosystem there is a risk that results of SCCs analysis are an artifact of the particular structure of the data (i.e. criteria to construct networks). By selecting several networks built by different authors we tried to minimize the bias due to each author's perspective in building networks as for the level of resolution (i.e. single species vs guilds) and structure of flows.

By this analysis we show that trophic hierarchies can be identified so that one has a clue about the sequential chain of energy flow in ecosystems; also, studying SCCs we found that compartmentalization in ecosystems is possible but it is not a general rule. Empirical evidences coupled with further modeling applications must be produced to corroborate these findings. In this respect what presented here should not be intended as a final statement about ecosystems functioning; rather it is only a trajectory for further research.

Acknowledgements – The authors were supported by the European Commission (Project DITTY contract No. EVK3-2001-00226).

References

Abrams, P., Menge, B. A., Mittelbach, G. G. et al. 1995. The role of indirect effects in food webs. – In: Polis, G. A. and Winemiller, K. O. (eds), Food webs: integration of pattern and dynamics. Chapman and Hall, pp. 371–396.

Almunia, J., Basterretxea, G., Aristegui, J. et al. 1999. Benthic-pelagic switching in a coastal subtropical lagoon. – Est. Coast. Shelf Sci. 49: 363–384.

Baird, D. and Milne, H. 1981. Energy flow in the Ythan Estuary, Aberdeenshire, Scotland. – Est. Coast. Shelf Sci. 13: 455–472.

Baird, D. and Ulanowicz, R. E. 1989. The seasonal dynamics of the Chesapeake Bay ecosystem. – Ecol. Monogr. 59: 329–364.

Bang-Jensen, J. and Gutin, G. 2000. Digraphs: theory, algorithms and applications. Springer Monographs in Mathematics. – Springer-Verlag.

Bodini, A. 2000. Reconstructing trophic interactions as a tool for understanding and managing ecosystems: application to

a shallow eutrophic lake. – Can. J. Fish. Aquat. Sci. 57: 1999–2009.
Burns, T. P., Higashi, M., Wainwright, S. et al. 1991. Trophic unfolding of a continental shelf energy-flow network. – Ecol. Modell. 55: 5–26.
Christian, R. R. and Luczkovich, J. J. 1999. Organizing and understanding a winter's seagrass foodweb network through effective trophic levels. – Ecol. Modell. 117: 99–124.
Cohen, J. E. 1978. Food webs and niche space. – Princeton Univ. Press.
De Angelis, D. L. 1992. Dynamics of nutrient cycling and food webs. – Chapman and Hall.
Girvan, M. and Newman, M. E. J. 2002. Community structure in social and biological networks. – Proc. Natl Acad. Sci. USA 99: 7821–7826.
Hagy, J. D. 2002. Eutrophication, hypoxia and trophic transfer efficiency in Chesapeake Bay. – PhD thesis, Univ. of Maryland, USA.
Heymans, J. J., Ulanowicz, R. E. and Bondavalli, C. 2002. Network analysis of the South Florida Everglades graminoid marshes and comparison with nearby cypress ecosystems. – Ecol. Modell. 149: 5–23.
Hopcroft, J. and Tarjan, R. 1973. Algorithm 447: efficient algorithms for graph manipulation. – Comm. ACM 16: 372–378.
Knuth, D. E. 1997. The art of computer programming Vol. 1: Fundamental algorithms. – Addison-Wesley.
Krause, A. E., Frank, K. E., Mason, D. M. et al. 2003. Compartments revealed in food-web structure. – Nature 426: 282–285.
Martinez, N. D., Hawkins, B. A., Dawah, H. A. et al. 1999. Effects of sample effort on characterization of food-web structure. – Ecology 80: 1044–1055.
Mateti, P. and Deo, N. 1976. On algorithms for enumerating all the circuits of a graph. – SIAM J. Comput. 5: 90–99.
May, R. M. 1972. Will a large complex system be stable? – Nature 238: 413–414.
Menge, B. A. 1995. Indirect effects in marine rocky intertidal interaction webs: paterns and importance. – Ecol. Monogr. 65: 21–74.
Odum, E. P. 1969. The strategy of ecosystem development. – Science 164: 262–270.
Odum, E. P. 1985. Trends expected in stressed ecosystems. – BioSci. 164: 419–422.
Pimm, S. L. and Lawton, J. H. 1980. Are food webs divided into compartments? – J. Anim. Ecol. 49: 879–898.
Polis, G. A. 1991. Complex trophic interactions in deserts: an empirical critique of food-web theory. – Am. Nat. 138: 123–155.
Polis, G. A. and Strong, D. R. 1996. Food web complexity and community dynamics. – Am. Nat. 147: 811–846.
Raffaelli, D. and Hall, S. J. 1992. Compartments and predation in an estuarine food web. – J. Anim. Ecol. 61: 551–560.
Raffaelli, D. and Hall, S. J. 1995. Assessing the relative importance of trophic links in food webs. – In: Polis, G. A. and Winemiller, K. O. (eds), Food webs: integration of pattern and dynamics. Chapman and Hall, pp. 185–191.
Read, R. C. and Wilson, R. J. 1998. An atlas of graphs. – Oxford Univ. Press
Schmitz, O. J. 1997. Press perturbations and the predictability of ecological interactions in a food web. – Ecology 78: 55–69.
Skiena, S. 1990. Strong and weak connectivity. – In: Skiena, S. (ed.), Implementing discrete mathematics: combinatorics and graph theory with mathematica. Addison-Wesley, pp. 172–174.
Tarjan, R. 1972. Depth first search and linear graph algorithms. – SIAM J. Comp. 1: 146–160.
Ulanowicz, R. E. 1983. Identifying the structure of cycling in ecosystems. – Math. Biosci. 65: 219–237.
Ulanowicz, R. E. 1986. Growth and development: ecosystem phenomenology. – Springer-Verlag.
Ulanowicz, R. E. 1995. Ecosystem trophic foundations: Lindeman exonerata. – In: Patten, B. C and Jorgensen, S. E. (eds), Complex ecology: the part-whole relation in ecosystems. Prentice-Hall, pp. 549–560.
Ulanowicz, R. E. 1996. Ecology. The ascendent perspective. – Columbia Univ. Press
Ulanowicz, R. E. and Kemp, W. M. 1979. Toward canonical trophic aggregation. – Am. Nat. 114: 871–883.
Williams, R. J., Berlow, E. L., Dunne, J. A. et al. 2002. Two degrees of separation in complex food webs. – Proc. Natl Acad. Sci. USA 99: 12913–12916.
Wootton, T. J. 2002. Indirect effects in complex ecosystems: recent progress and future challenges. – J. Sea Res. 48: 157–172.
Wulff, F. and Ulanowicz, R. E. 1989. A comparative anatomy of the Baltic Sea and Chesapeake Bay. – In: Wulff, F., Field, J. G and Mann, K. H. (eds), Network analysis in marine ecology. Springer-Verlag, pp. 232–256.
Yodzis, P. 1982. The compartmentation of real and assembled ecosystems. – Am. Nat. 120: 551–570.
Yodzis, P. 2000. Diffuse effects in food webs. – Ecology 81: 261–266.

Subject Editor: Per Lundbers

Appendix 1

1) Notation and concepts

We define an ecological network as a directed graph (digraph) G(V,E) with V vertices, the species/trophospecies or nutrient pools, and E oriented, weighted edges, the trophic exchanges, usually measured as grams of C for hectare for year (or other currencies, according to the aim of the research).

We represent the digraph as an adjacency matrix, that is to say a squared $V \times V$ matrix where every coefficient represents the flux of matter from row-vertex to column one. If the edge does not exist the coefficient will be 0.

A strongly connected component (SCC) of the graph is a maximal subset of vertices set $U \subseteq V$ where for any u_1, $u_2 \in U$, there is at least a directed path (sequence of vertices and edges) from u_1 to u_2 and one from u_2 to u_1, so that u_1 and u_2 are reachable from each other. The vertices of any directed graph can be partitioned into n strongly connected components $S_1, \ldots, S_n$ where $V \geq n$. Strongly connected components do not share vertices with each other (disjoint sets).

The size of a strongly connected component $|S_i|$ is the number of vertices belonging to the ith of these components. A SCC with $|S_i| = 1$ can be only a source (a vertex with just outgoing edges), a sink (just incoming edges) or a cross-vertex, that is a node connecting two or more SCCs. If the size of a strongly connected component is equal to the total number of nodes, $|S_i| = V$, the graph is said to be strongly connected. If a digraph encompasses more than 1 strongly connected component, it will have no Hamiltonian cycles, that is cycles that visit all the vertices in the graph; this is because cycles involving vertices in different SCC are not possible (Bang-Jensen and Gutin 2000); in fact all the cycles in

the graph remain confined into its strongly connected components.

Aggregating vertices inside every SCC yields a directed acyclic graph (DAG). Because a DAG comprises only one-way edges, it is a pictorial representation of a hierarchy of dependencies involving graph components. For ecological flow networks such hierarchy pertains the transfer of mass/energy throughout the system.

According to the hierarchical nature of DAGs, a procedure called topological sorting can be applied so that one can order graph elements according to precedence constraints. Thus, given a directed acyclic graph G'(V',E'), a topological sort is a linear ordering of all its vertices from left to right, such as for every edge ($e_{v1,v2} \in E'$), v_1 precedes v_2 in the order of vertices. A topological sort must include at least a source and a sink.

2) Finding SCCs

In order to find SCCs in a digraph we should perform two depth first search-DFS visits: the first one on the digraph, the second one on its transposed form, that is to say with all the edges with inverted directions. Exploring a graph in search of SCCs requires a calculation effort that is linearly proportional to the graph size, considering vertices and edges ($\Theta(V+E)$) (Tarjan 1972). The simplest algorithm for SCCs (Hopcroft and Tarjan 1973) has been used here. See below for a ready to use code.

3) Aggregating SCCs

Once obtained the SCCs, we should lump together all the vertices belonging to the same component. We utilized the algorithm proposed by Hirata (1978) that has been extended in Ulanowicz and Kemp (1979) work. Because of in our analysis we need just topological values (presence/absence of edges), the aggregation procedure is simplified. We may describe this process as an homomorphic mapping ϕ from digraph G(V,E) to G'(V',E') (Hirata and Ulanowicz, 1985), that is to say a surjective map of the graph. The aggregation process causes self-loops to be created: these self-loops (diagonal coefficients on the adjacency matrix) have been eliminated before running the topological sorting.

4) Topological Sorting

Topological sorting orders, in linear time (Knuth 1997), the vertices and edges of a DAG in a simple and consistent way; hence, in the framework of ecological flow networks, it makes evident the fundamental linear pathways that energy and matter use to travel in the ecosystem. In other words, topological sort produces a chain of donors (such as resources) and consumers in a succession of steps that highlights the linear fundamental flux of matter, which enters the system and leaves it, after a certain time that is affected by cycling.

5) Programming SCCs, DAG and topological sort

Here below a complete software in C that accomplishes all the procedures performed in the paper is presented. The code requires GSL (GNU Scientific Library, www.gnu.org/software/gsl).

```
/*
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - Strongly Connected - - - - - - - - - - - - - - - - - - - - - - - -
- - this is a sample code - - - - - - - - - - - - - - - - - - - - - - -
- - allesina@msu.edu - - - - - - - - - - - - - - - - - - - - - - - - - -

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
INPUT: Matrix File
OUTPUT: SHELL or FILE
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
COMPILING:
gcc scc.c -o SCC -lgsl -lgslcblas -O2
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

RUNNING:
./SCC test.txt
test.txt contains
1st line: nubmer of compartments  - >N
2nd line . . . N+1th line Adjacency matrix
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

//Libraries must include GSL
//http://sources.redhat.com/gsl/
*/
#include  <stdio.h >
#include  <gsl/gsl_matrix.h >
#include  <gsl/gsl_vector.h >
#include  <gsl/gsl_blas.h >
#include  <gsl/gsl_permutation.h >
#include  <gsl/gsl_sort_vector_double.h >
//Global Variables
int N; //number of vertices
gsl_matrix * M; //the adjacency matrix
//- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
//VARS STRONGLY CONNECTED COMPONENTS
//- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
gsl_vector * DFSC; //colour
gsl_vector * SCCS; //belonging SCC
gsl_vector * DFSD; //Discovery
gsl_vector * DFSF; //Finishing
gsl_permutation * DFSOrder;
int DFS_T; //Time
int SCC_Num; //SCC count
```

```
// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
//VARS AGGREGATION
// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
gsl_matrix * Agg; //the aggregation matrix
gsl_matrix * DAG; //Resulting DAG once the Sccs have
been aggregated
// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
//INPUT READING
// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
int ReadFile(char * myfile)
{
  char *line = NULL;
  FILE *fp;
  int i,j,tmp;
  printf ("\nOpening:%s\n", myfile);
  if ((fp = fopen (myfile, "r")) = = NULL)
   {
     printf ("Cannot open the file!!!\n");
     return 1;
   }
  fscanf(fp,"%d\n",&N);
  printf("Number of vertices:%d\n",N);
  M = gsl_matrix_calloc(N,N);
  for (i = 0;i < N;i + +)
   {
     for (j = 0;j < N;j + +)
      {
       fscanf(fp,"%d",&tmp);
       gsl_matrix_set(M,i,j,tmp);
      }
       fscanf(fp,"\n");
   }
  //print the adjacency matrix
  printf("\nAdjacency matrix\n");
  PrintMat(M);
  return 0;
}
// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
//PRINT A MATRIX
// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
int PrintMat(gsl_matrix * C)
{
  int i,j;
    for (i = 0;i < N;i + +)
    {
      for (j = 0;j < N;j + +)
       {
         printf("%1.0f\t", gsl_matrix_get(C,i,j));
       }
      printf("\n");
    }
    return 0;
}
// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
//AGGREGATION INTO DAG
// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
int AggMat(void)
{
gsl_matrix * M1;
gsl_matrix * M2;
int i,j,k;
double z;
M1 = gsl_matrix_calloc(N,N);
M2 = gsl_matrix_calloc(N,N);
for(i = 0;i < N;i + +)
  {
    for (j = 0;j < N;j + +)
     {
       z = 0;
       for (k = 0;k < N;k + +)
          {
            z = z + gsl_matrix_get(Agg,i,k)*gsl_
            matrix_get(M,k,j);
          }
       if (z > 0)
          {
      z = 1;
          }
       gsl_matrix_set(M1,i,j,z);
     }
  }
  for(i = 0;i < N;i + +)
  {
    for (j = 0;j < N;j + +)
    {
    z = 0;
    for (k = 0;k < N;k + +)
      {
    z = z + gsl_matrix_get(M1,i,k)*gsl_matrix_
    get(Agg,j,k);
    }
    if (z > 0)
    {
    z = 1;
    }
gsl_matrix_set(M2,i,j,z);
    }
  }
//Remove Self Loops
for(i = 0;i < N;i + +)
  {
    if (gsl_matrix_get(M2,i,i) = = 1)
     {
       gsl_matrix_set(M2,i,i,0);
     }
  }
//Build Dag
DAG = gsl_matrix_calloc(SCC_Num,SCC_Num);
  for(i = 0;i < SCC_Num;i + +)
    {
      for(j = 0;j < SCC_Num;j + +)
        {
        gsl_matrix_set(DAG,i,j,gsl_matrix_
        get(M2,i,j));
```

```
            }
        }
    return 0;
}
// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
//STRONGLY CONNECTED COMPONENTS
// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
int DFS (int i,int tras)
{
  int k;
  DFS_T++;
  gsl_vector_set(DFSC,i,1);
  gsl_vector_set(DFSD,i,DFS_T);
  if (tras==1)
    {
      gsl_vector_set(SCCS,i,SCC_Num);
    }
  for (k=0;k<N;k++)
    {
      if(gsl_matrix_get(M,i,k)>0)
        {
         if (gsl_vector_get(DFSC,k)==0)
           {
             DFS(k,tras);
           }
        }
    }
  DFS_T++;
  if (tras==0)
   {
      gsl_vector_set(DFSF,i,-DFS_T); //reverse
   }
  gsl_vector_set(DFSC,i,2);
  return 0;
}
int SCC (void)
{
  int i,l;
  printf ("\nSTRONGLY CONNECTED
  COMPONENTS\n");
  SCC_Num=0;
  DFSC=gsl_vector_calloc(N);
  DFSD=gsl_vector_calloc(N);
  DFSF=gsl_vector_calloc(N);
  SCCS=gsl_vector_calloc(N);
  DFSOrder=gsl_permutation_calloc(N);
  DFS_T=0;
  for (i=0; i<(N); i++)
    {
      if (gsl_vector_get(DFSC,i)==0)
        {
          DFS(i,0);
        }
    }
  //order the nodes
  gsl_sort_vector_index(DFSOrder, DFSF);
  //repeat DFS search on the transpose ordering nodes
  by time
  gsl_matrix_transpose(M);
  //re initialize vectors
  DFSC=gsl_vector_calloc(N);
  DFSD=gsl_vector_calloc(N);
  DFS_T=0;
  //starts DFS
  for (i=0; i<(N); i++)
    {
      if(gsl_vector_get(DFSC,gsl_permutation_get
      (DFSOrder,i))==0)
        {
          DFS(gsl_permutation_get(DFSOrder,i),1);
          SCC_Num++;
        }
    }
  //Allocate Aggregation Matrix
  Agg=gsl_matrix_calloc(N,N);
  //lists the SCCS
  for (i=0; i<(SCC_Num); i++)
    {
      printf ("\nStrongly Connected Component
      #%d\n",i);
      for (l=0;l<(N);l++)
    {
      if (gsl_vector_get(SCCS,l)==i)
          {
              printf("%d\t",l);
              gsl_matrix_set(Agg,i,l,1);
          }
      }
    }
  //re-transpose the matrix
  gsl_matrix_transpose(M);
  printf ("\nNumber of Components:%d\n",
  SCC_Num);
  return 0;
}
//- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
//TOPOLOGICAL SORT
//- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
int Toposort(int i)
{
  int k,z;
  DFS_T++;
  gsl_vector_set(DFSC,i,1);
  gsl_vector_set(DFSD,i,DFS_T);
  for (k=0;k<SCC_Num;k++)
    {
      if(gsl_matrix_get(DAG,i,k)>0)
        {
          if (gsl_vector_get(DFSC,k)==0)
            {
              Toposort(k);
            }
```

```
        }
      }
    DFS_T++;
    gsl_vector_set(DFSC,i,2);
    //Print the component
    printf("[");
    for(z=0;z<N;z++)
      {
        if(gsl_vector_get(SCCS,z)==i)
          {
            printf("%d ",z);
          }
      }
    printf("]<-");
    return 0;
}
//-----------------------------------------
//MAIN
//-----------------------------------------
int main(int argc, char *argv[])
{
    int i,j;
    //reads the command line: filename.txt
    char *myfile;
    myfile=argv[1];
    printf("\nSelected file:%s\n", myfile);
    //reads the File
    j=ReadFile(myfile);
//-----------------------------------------
//start
//-----------------------------------------
    //Strongly Connected Components
    j=SCC();
    //Aggregation into DAG
    printf("\nAGGREGATION MATRIX\n");
    PrintMat(Agg);
    //aggregate the compoments
    j=AggMat();
    printf("\nResulting Directed Acyclic Graph\n");
    for (i=0;i<SCC_Num;i++)
      {
        for (j=0;j<SCC_Num;j++)
          {
            printf("%1.0f\t", gsl_matrix_get(DAG,i,j));
          }
       printf("\n");
      }
    //Topological Sort
    DFSC=gsl_vector_calloc(SCC_Num);
    DFSD=gsl_vector_calloc(SCC_Num);
    DFSF=gsl_vector_calloc(SCC_Num);
    DFS_T=0;
    printf ("\n TOPOLOGICAL SORTING\n");
      for (i=0; i<SCC_Num; i++)
      {
        if (gsl_vector_get(DFSC,i)=0)
          {
            Toposort(i);
          }
      }
      printf("\n\n");
    //-----------------------------------------
    //stop
    //-----------------------------------------
    return 0;
}
```