SnailVis: a Paradigm to Visualize Complex Networks

Mariano G. Beiró, Jorge R. Busch, and J. Ignacio Alvarez-Hamelin*

Facultad de Ingeniería, Universidad de Buenos Aires,
Paseo Colón 850, C1063ACV - Buenos Aires - Argentina
{mbeiro,jbusch}@fi.uba.ar,ihameli@cnet.fi.uba.ar
http://cnet.fi.uba.ar

Abstract. We propose a new non-parametric and linear-complexity algorithm to visualize complex networks, which were previously decomposed in subsets according to some criteria. We show two representations: the first including all edges and vertices and the second, summarized, highlighting subsets and their relations. In this paper we use a community decomposition algorithm to generate the subsets; then we rank them by the number of inter-community connections. We also highlight the central core of each community, that is, the subset with the highest connectivity level, which is the $k_{\max}$-core of the k-core decomposition.

Key words: complex networks, visualization, communities detection

1 Introduction

Visualization is a useful tool to analyze *at a glance* prominent characteristics of Complex Networks, especially to compare between different samples or models. Several complex networks have millions of vertices and edges, which makes visualization rather difficult. A first solution consists in using an abstraction of the real network: given a set partition, we draw each set as a vertex and the relationships between them as a single edge. This solution does not allow to find or highlight particular vertices which may be important, hiding the internal sets structure.

In this paper we propose a linear time complexity algorithm to visualize all vertices and edges, and their corresponding abstraction. Our algorithm is based on a partition of the network, previously built in order to highlight a certain prominent characteristic. We draw the subsets in a certain order, placing them in a spiral. We give each vertex a ratio related to its degree, and then draw each subset in a disk whose surface is proportional to the sum of the surfaces of all its vertices; placing them randomly inside. To highlight each subset's *backbone* we place its $k_{\max}$-core [19, 5, 3] in the disk center, keeping some distance with the rest of the vertices.

* INTECIN (UBA-CONICET)

In order to make the visualization useful for big networks, it is important to choose a low complexity partition algorithm. In this paper we use an efficient implementation of community decomposition, based on the modularity function proposed by Newman [17].

On the other hand, more traditional graph drawing methods present a higher complexity. Among the ones based on spectral decomposition, the algorithm by Har'El [9] finds clusters in $\mathcal{O}(n^3)$. Force-directed methods like Kamada-Kawai [10] are $\mathcal{O}(n^2)$ and only draw particles, i.e. do not include a clustering strategy. In this work, the efficient partitioning algorithm plus a fast placing of subsets in a spiral results in a low complexity method.

To the best of our knowledge, the first work to introduce spirals to visualize data is the work of Lambert [11], who displayed the periodic variation of solar heating at different depths. More recently, Carlis and Konstan [7] proposed this curve to highlight serial periodic data, where serial attributes are shown on the spiral axis and periodic ones along the radii. Similar work in time series is presented in [21], improving the analysis. The main difference with our proposal is that we use the spiral to place the center of each subset of the partitioned graph, since we are interested in highlighting the relationships between each subset considering that the most central (the closest to the spiral origin) is the most connected. Another difference is that real networks present partitions whose size distributions have a long tail, so that we use the Fermat's spiral, where the radius difference decreases.

Finally, to cover other aspects of our proposal, we mention that LaNet-vi [12, 1] provides a low complexity visualization for large networks based on k-core decomposition [19, 5, 3]. In [1] and [2] transparency is used to display all edges, the same principle is applied in this work.

2 SnailVis

We visualize networks at 2 different levels: (i) The node level, at which all nodes and connections between them are rendered, and (ii) The partition level, which provides a high-level view of the structure.

2.1 The node level

At the node level we deploy the whole graph, and by this we mean that given $G = (V, E)$, every node $v \in V$ and every edge $e = (v_1, v_2) \in E$ is drawn.

Each node is represented by a circle, and its size is in some way related to the degree d_i. As degree distribution is typically *heavy-tailed* for many complex networks, we apply a logarithmic scale. So r_i (ratio for node $v_i \in V$) is computed as:

$$r_i = K \cdot log(d_i) \tag{1}$$

An edge $e = (v_1, v_2) \in E$ is represented as a line from v_1's center up to v_2's.

Besides, the whole graph is partitioned into sets, as we explain later on. So that the vertices are gathered in their sets, and each set C_i is assigned a virtual

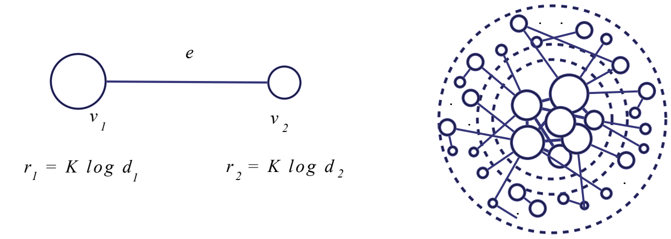

Fig. 1. Two connected vertices with different degrees (left). A set; the $k_{\max}$-core is place at the center (right).

circle in the image, where its nodes are deployed. The ratio R_i of this circle is proportional to the ratios of the belonging nodes, i.e.:

$$R_i = K_2 \cdot \sum_{v_j \in C_i} r_j = K_3 \cdot \sum_{v_j \in C_i} log(d_j) \tag{2}$$

K, K_2 and K_3 are geometric constants which were manually adjusted.

Each set contains a subset of *central nodes* (which usually have many connections between them) and *peripheral nodes* which tend to be connected to the central ones. The centrality measure we use is based on k-core decomposition, which is defined as following. A k-core is the maximum subgraph whose vertices have at least degree k in the induced subgraph; in other words each vertex should have at least k neighbors in the induced subgraph [19, 5]. The $k_{\max}$-core is the k-core with maximum k and not empty.

The circle is then divided into a central core and a peripheral ring, with another ring as margin between them. The areas of the circle and the ring are, again, set according to their nodes' size. Inside each region, nodes are placed at random. The right part of Figure 1 illustrates all these facts.

Finally, sets $C_1, C_2, ..., C_n$ are deployed into their virtual circles following a *spiral* curve (see Figure 2). This curve has several benefits:

- It takes advantage of the center of the picture, as the spiral starts in position ($\rho = 0, \theta = 0$)
- It may take a second round, or third, etc, if the amount of sets is big and one turn is not enough.
- Positions in the spiral may be computed assuring certain separation between sets.

  The spiral equation is:

$$\rho = A \cdot \theta^{\beta}, \beta \in R \ , \tag{3}$$

with $\beta = 0.5$, which is the *Fermat's spiral.*

Every set C_i will be centered in some point (ρ_i, θ_i) belonging to this locus. We sort them by their amount of external connections a_i, and we start with the one having the biggest value (C_1). The center is positioned at distance R_1 from the origin ($\rho = 0, \theta = 0$). This makes $\rho_1 = R_1$ so that θ may be immediately calculated.

Then C_2 is positioned on the spiral subject to the following condition: the distance from C_1's center should be equal to some value related to R_1 and R_2 (we will refer to it later). This restriction assures that C_1 and C_2 do not overlap in the visualization.

Distance between C_1 and C_2 depends on ρ_2 in the following way:

$$d(\rho_2) = {\rho_2}^2 + B_1 \cdot \rho_2 \cdot cos\left(\left(\frac{\rho_2}{K}\right)^{1/\beta} - \left(\frac{\rho_1}{K}\right)^{1/\beta}\right) + B_2 \ , \tag{4}$$

where $B_1 = -2 \cdot \rho_1$ and $B_2 = {\rho_1}^2$.

Solving $d(\rho_2) = f(R_1, R_2)$ gives raise to a non-linear equation on ρ_2, which we solved by the Newton-Raphson iterative method. After running the algorithm for many networks, we observed that an amount of 100 iterations for each set is enough to guarantee an error of less than 1%.

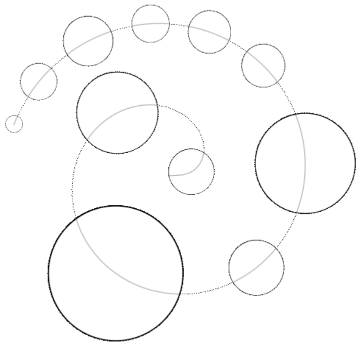

Fig. 2. Sets distributed in a spiral.

2.2 The partition level

At this level we only deploy each set as an abstract entity, without showing its internal structure. Nevertheless, the object's sizes will tell something about internal qualities. While the previous level stressed the sets' size, understood as the product of degrees of its nodes, in this level we intend to separate edges in internal and external: this will give us information about connectivity.

In this level each set C_i is represented by a circle, whose ratio R_i equals the number of internal edges e_{ii}.

A line between sets C_i and C_j represents all the connections between them; the line width L_{ij} being equal to e_{ij}, the number of edges between C_i and C_j.

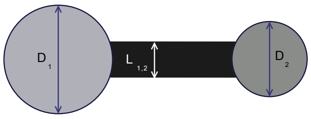

Fig. 3. High-level partitions deployment.

Figure 3 illustrates the situation: C_1 has 30 internal edges, C_2 has 20 of them, and there are 10 connections between C_1 and C_2. It may also be deduced that $\sum_{v_i \in C_1} d(v_i) = 70$ (sum of degrees in C_1) and $\sum_{v_i \in C_2} d(v_i) = 50$.

Finally, we shall mention that the two representations (node level and partition level) may be overlapped, as the sets have the same coordinates and order in both. To achieve this, the sets sizes have been normalized, and the distance between two neighboring sets in formula 4 has been computed as the maximum between their radii in both representations, to assure that they will not touch each other.

3 On partitions inducing community structure

3.1 Introduction

A community structure is induced in a network by a partition of the set of nodes in subsets, called communities, such that most connections are between nodes in a same community. A quantitative measure of these informal notion is given by modularity, which was introduced by Newman [17] and was extensively used in many works to evaluate the goodness of a community decomposition. Proofs of this may be found in the belgian mobile phone network studied by Blondel *et al.* [4] and the social networks analyzed in [17]. In this section we formalize a notion of weak optimality, and we describe an algorithm to obtain weakly optimal partitions. Numerical experience with real world graphs shows that our low complexity algorithm gives results that are very good when compared with another algorithms, using modularity as a measure of "goodness".

3.2 Some notations

Let $G = (V, E)$ be a non directed graph, and let $m : V \times V \to \mathbb{Z}$ be a function that satisfies $\forall (u, v) : m(u, v) = m(v, u) \geq 0$ and $\forall u : k(u) \doteq \sum_{v \in V} m(u, v) > 0$. Then $K \doteq \sum_{u \in V} k(u) > 0$. Then $m'(u, v) \doteq m(u, v)/K$ induces a probability measure in $V \times V$, and $k'(u) \doteq k(u)/K$ induces a probability measure in V, that is the marginal probability of m' (there is only one marginal probability because m is assumed symmetric). We shall denote also $k'(u, v) = k'(u)k'(v)$, which induces another probability measure in $V \times V$.

3.3 On partitions of V

Let $\mathcal{C}$ be a family of not empty pairwise disjoint subsets of V. We call $\mathcal{C}$ a partition of V when $\cup\mathcal{C} = V$. We shall consider the usual (lattice) partial order between partitions of V, $\mathcal{C} \preceq \mathcal{D}$ if $\mathcal{D}$ is a refinement of $\mathcal{C}$, or, which is the same, for any $C \in \mathcal{C}$ it holds

$$C = \cup\mathcal{D}_C$$

where $\mathcal{D}_C \doteq \{D \in \mathcal{D} : D \subset C\}$. Notice that with this partial order, there is always a minimal partition $\mathcal{C}_0 \doteq \{V\}$ and a maximal partition $\mathcal{C}_1 \doteq \{\{v\} : v \in V\}$.

Given a partition of V $\mathcal{C}$, we shall consider

$$D(\mathcal{C}) \doteq \cup_{C\in\mathcal{C}} C \times C$$

and its complement in $V \times V$

$$\bar{D}(\mathcal{C}) \doteq \cup_{C_1,C_2\in\mathcal{C},C_1\neq C_2} C_1 \times C_2$$

The Newman-Girvan modularity [17] of a partition $\mathcal{C}$ is then

$$Q(\mathcal{C}) = m'(D(\mathcal{C})) - k'(D(\mathcal{C}))$$

We introduce also

$$\bar{Q}(\mathcal{C}) = m'(\bar{D}(\mathcal{C})) - k'(\bar{D}(\mathcal{C}))$$

Then of course we have $Q(\mathcal{C}) + \bar{Q}(\mathcal{C}) = 0$.

3.4 Absolute and relative resolution

Given a partition $\mathcal{C}$ of V, consider

$$t : \mathcal{C} \times \mathcal{C} \to \mathbb{R}, t(C_1, C_2) \doteq \frac{m'(C_1 \times C_2)}{k'(C_1 \times C_2)}$$

We define the *absolute resolution* of $\mathcal{C}$,

$$t(\mathcal{C}) \doteq \max_{C_1,C_2\in\mathcal{C},C_1\neq C_2} t(C_1, C_2)$$

(if $|\mathcal{C}| = 1$, we set $t(\mathcal{C}) = 0$).

Then it holds that $\mathcal{C} \preceq \mathcal{D} \Rightarrow t(\mathcal{C}) \leq t(\mathcal{D})$. and as a consequence $t(\mathcal{C}) \leq t(\mathcal{C}_1)$ for any partition $\mathcal{C}$ of V, and we may define the *relative resolution* of $\mathcal{C}$ as

$$t'(\mathcal{C}) \doteq \frac{t(\mathcal{C})}{t(\mathcal{C}_1)}$$

3.5 An extension of modularity

Let us introduce the modularity at resolution t [18],

$$Q(t, \mathcal{C}) = m'(D(\mathcal{C})) - tk'(D(\mathcal{C}))$$

and also

$$\bar{Q}(t, \mathcal{C}) = m'(\bar{D}(\mathcal{C})) - tk'(\bar{D}(\mathcal{C}))$$

Then of course we have $Q(t, \mathcal{C}) + \bar{Q}(t, \mathcal{C}) = 1 - t$.

Notice then that

$$\bar{Q}(t, \mathcal{C}) = \sum_{C_1, C_2 \in \mathcal{C}, C_1 \neq C_2} (t(C_1, C_2) - t)k'(C_1 \times C_2)$$

3.6 Optimization

The Newman-Girvan modularity Q is considered as a good measure of the "goodness" of the community structure induced by $\mathcal{C}$. The problem of its maximization has been shown to be NP-complete [6], and several papers have dealed with it, using diverse techniques. Here we show a low complexity approach, based on a very simple idea that gives raise to a simple algorithm.

For any partition we have $Q(t, \mathcal{C}) + \bar{Q}(t, \mathcal{C}) = 1 - t$, thus the problem of maximizing $Q(t, \mathcal{C})$ with fixed resolution t is equivalent to the problem of minimizing $\bar{Q}(t, \mathcal{C})$. For this, we have the expression

$$\bar{Q}(t, \mathcal{C}) = \sum_{C_1, C_2 \in \mathcal{C}, C_1 \neq C_2} (t(C_1, C_2) - t)k'(C_1 \times C_2)$$

Let us call the partition $\mathcal{C}$ *submodular* at resolution t when $t(\mathcal{C}) \leq t$. In this case, all the terms in the sum above are negative.

We have shown that $\mathcal{C}$ is submodular at resolution t if and only it is *weakly optimal* at resolution t, which means that $Q(t, \mathcal{D}) \leq Q(t, \mathcal{C})$ whenever $\mathcal{D} \preceq \mathcal{C}$

Suppose that we choose communities $C_1, C_2 \in \mathcal{C}$ such that $t(C_1, C_2) = t(\mathcal{C})$, and consider the new partition $\mathcal{D}$ of V obtained from $\mathcal{C}$ by replacing C_1 and C_2 by $C_1 \cup C_2$. Then $t(\mathcal{D}) \leq t(\mathcal{C})$, with strict inequality if (C_1, C_2) were the only pair realizing $t(\mathcal{C})$. This gives an algorithm to obtain, starting at $\mathcal{C}_1$, a sequence of partitions with decreasing resolutions, and as a consequence weakly optimal for increasing ranges of t. We stop the algorithm when we arrive at resolution 1, thus our actual partition is weakly optimal at resolution 1, that is, it is weakly optimal for the Newman-Girvan modularity.

4 Case studies

We have applied SnailVis to visualize community structures, as introduced in section 3. The source code is publicly available at [20].

In the following case studies we will show how our representations at both levels may be used to decide at a glance if a certain network has achieved a community structure, to compare different networks and to extract conclusions.

4.1 An *e*-mail network (Small)

The *e*-mail network is formed by 1,100 university members sending 10,000 *e*-mails between them [8]. Applying the optimization algorithm based on submodularity, we get 11 communities, and a modularity $Q = 0.522$.

Our visualization provides the following pictures:

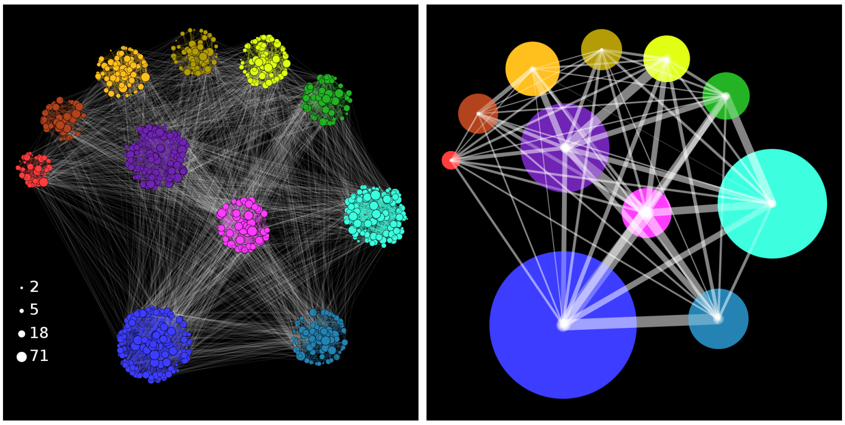

Fig. 4. E-mail network visualization. (*i*) Node level. (*ii*) Partition level.

From the node-level picture we observe that nodes are rather similar in degree, and communities are also quite similar in size. This is verified in the statistics in figure 5.(*i*)(*ii*), where the distribution is fitted with an exponential, fast-decreasing function. Besides, the pink community, being the one with more external connections, is not the biggest one in terms of degree. This implies a possibly poor community, as in shown at the partition level: it has several thick edges compared to its ratio. The blue community is quite good instead, as well as the light blue one. Both have many internal connections and few links outside.

4.2 An *e*-vote network (Medium)

This network was extracted from Wikipedia [13, 14]. It is formed by 7,000 users voting for several administrator elections. The modularity is $Q = 0.356$.

Our visualization shows a poorer community partition with respect to the previous one, i.e. the first two sets are not good partitions considering their size and the size of their outgoing edges. Figures 4.2.(*ii*) and (*iii*) confirm this. On the other hand the node-level exhibits a correlation between centrality and degree: nodes in the central core with degree in the order of 500, and nodes in the periphery with about ten connections. This happens because the degree distribution obeys a power-law: there are few members who were much voted,

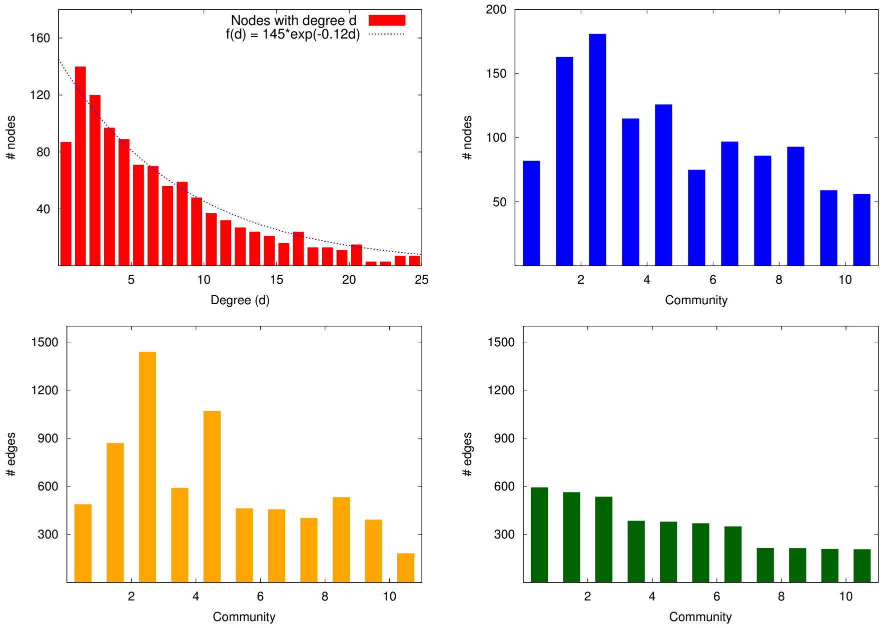

Fig. 5. Statistics for the email network. Communities are sorted as in the spiral. (*i*) Degree distribution. (*ii*) Community sizes. (*iii*) Internal edges for each community. (*iv*) External edges for each community.

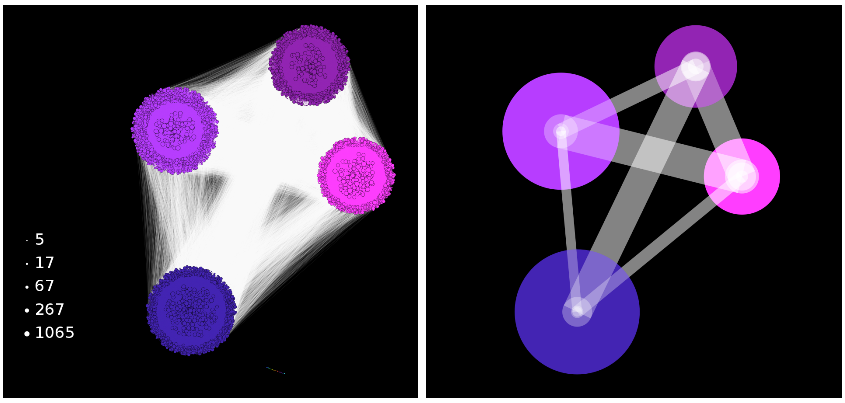

Fig. 6. Wikipedia voting network. (*i*) Node level. (*ii*) Partition level.

and many members with 1 or 2 votes. (see figure 4.2.(*i*) for more details). For this reason degrees vary on a long range, unlike the e-mail network where the distribution is exponential.

Aside from the four big communities in the graph, we also found a dozen more with just 2 o 3 members, remaining disconnected from the rest of the graph: i.e.

one of them voted for the other, but no one else voted for any of them. This little communities may be seen at the node-level picture.

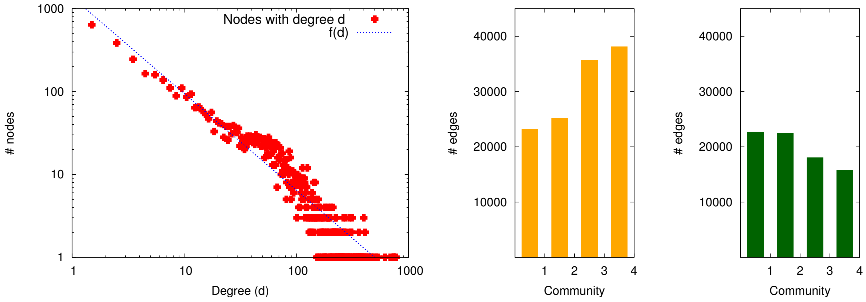

Fig. 7. Statistics for the Wikipedia voting network. Communities are sorted as in the spiral. (*i*) Degree distribution and least squares fit $f(d) = 2282 \cdot d^{-1.58}$. (*ii*) Internal edges for each of the 4 biggest communities. (*iii*) External edges for each of the 4 biggest communities.

4.3 A web graph (Very Large)

This network represents 5 million links between 875,000 web pages [15]. It was extracted from the 2002 Google Programming Contest. Modularity is $Q = 0.968$.

4.4 Evolution of the algorithm on a dolphins interaction network

This network represents associations of 62 dolphins in a community at New Zealand [16]. Visualization in figure 9 makes clear how SnailVis works. When t is big (left) it allows for small communities, whose internal connections are comparable to the external ones. This happens because a big t penalizes the existence of big communities (see section 3.5). For $t = 1$ (medium), Q is the Newman modularity, which balances the amount external and internal links. For $t < 1$ (right) big communities are allowed to appear, until at some point, the whole graph forms a single community.

5 Conclusions

We proposed a new visualization paradigm useful to analyze network properties like connectivity and degree from the perspective of graph partitions. Visualizing a good network partition in a high-level of abstraction may easily reveal something about the behavior and evolution of the system. The algorithm's low complexity makes it proper for large complex networks.

In the particular case of using modularity to get communities, our visualization was useful to compare networks. We could establish a rank based on

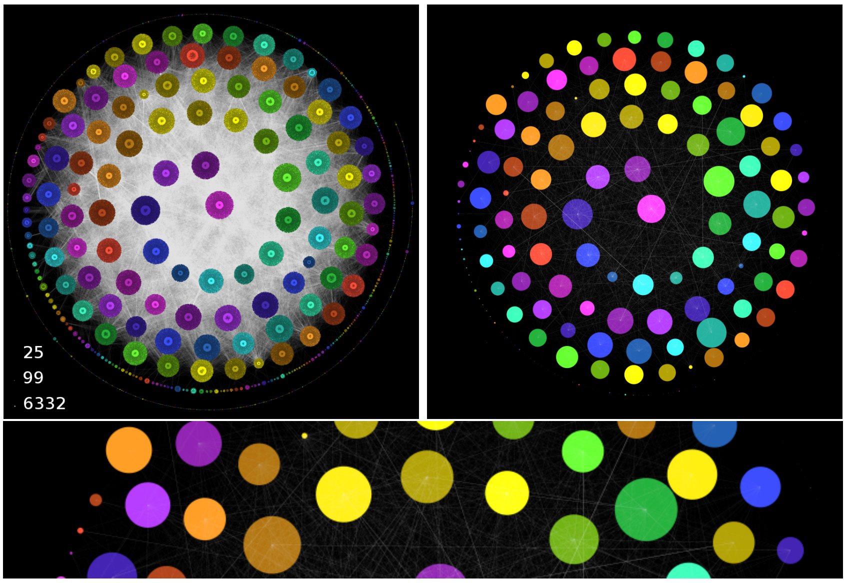

Fig. 8. Web graph. (*i*) Node level. (*ii*) Partition level. (*iii*) Partition level zoom.

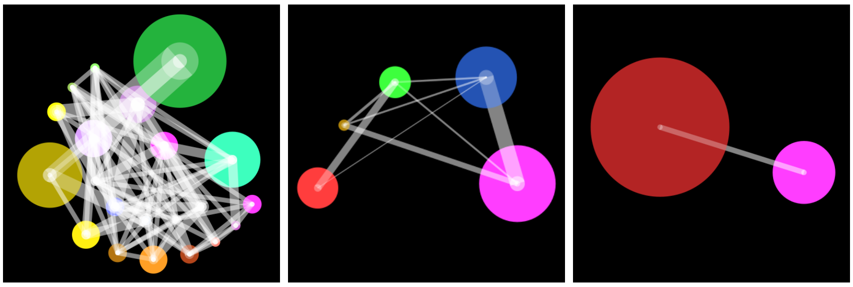

Fig. 9. Evolution of a dolphins interaction network. We represent communities evolution for different values of t. (*i*) $t = 5$, small communities, with many external connections, (*ii*) $t = 1$, maximal modularity, (*iii*) $t = 0.3$, very big communities.

community structure, as there is a high correlation between the edges/circles ratio in the pictures and the modularity measure Q of the associated partition. I.e., the following sequence with increasing Q: (e-vote, e-mail, web graph) may also be deduced by visual comparison of the aforementioned ratio.

Acknowledgments. This work was partially financed by UBACyT I413/2008 project, INTECIN, and a Peruilh grant of FIUBA.

We also thank Prof. Alex Arenas for his e-mail dataset, Prof. Jure Leskovec for his datasets from Wikipedia and the web graph, and Prof. Mark Newman for the dolphins interaction network.

References

1. J I Alvarez-Hamelin, M G Beiro, L Dall'Asta, A Barrat, and A Vespignani. **La**rge **Net**work **vi**sualization tool, `http://sourceforge.net/projects/lanet-vi`, 2005.
2. J I Alvarez-Hamelin, M Gaertler, R Görke, and D Wagner. Halfmoon - A new Paradigm for Complex Network Visualization. Technical Report TR-2005-29, Faculty of Informatics, Universität Karlsruhe (TH), 2005.
3. V Batagelj and M Zaversnik. Generalized Cores. *CoRR*, arXiv.org/cs.DS/0202039, 2002.
4. V. Blondel, J-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *J. Stat. Mech.*, 2008(1):10008, Oct. 2008.
5. B Bollobás. The evolution of sparse graphs. *Graph Theory and Combinatorics*, pages 35–57, 1984.
6. U Brandes, D Delling, M Gaertler, R Görke, M Hoefer, Z Nikoloski, and D Wagner. On modularity clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20:172–188, 2008.
7. J V Carlis and J A Konstan. Interactive visualization of serial periodic data. In *UIST '98: Proceedings of the 11th annual ACM symposium on User interface software and technology*, pages 29–38, New York, NY, USA, 1998. ACM.
8. R Guimerà, L Danon, A Díaz-Guilera, F Giralt, and A Arenas. Self-similar community structure in a network of human interactions. *Phys. Rev. E*, 68(6):065103, Dec 2003.
9. N. Har'El. Finding the largest eigenvalues of a real symmetric matrix, and corresponding eigenvectors, 1992. Research Report, Department of Mathematics, Israel Institute of Technology.
10. T Kamada and S Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31:7–15, 1989.
11. J H Lambert. *Johann Heinrich Lamberts Pyrometrie, oder, Vom Maasse des Feuers und der Warme [microform] : Mit acht Kupfertafeln.* Haude und Spener, 1779.
12. LArge NETwork VIsualization tool. `http://xavier.informatics.indiana.edu/lanet-vi/`.
13. J Leskovec, D Huttenlocher, and J Kleinberg. Signed networks in social media. In *28th ACM Conf. on Human Factors in Computing Systems*, pages 407–416, 2010.
14. J Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Predicting positive and negative links in online social networks. In *WWW 2010*, 2010.
15. J Leskovec, K J Lang, A Dasgupta, and M W Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *CoRR*, abs/0810.1355, 2008.
16. D Lusseau, K Schneider, O J Boisseau, P Haase, E Slooten, and S M Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. can geographic isolation explain this unique trait? *Behavioral Ecology and Sociobiology*, 54(4):396–405, 2003.
17. M E J Newman and M Girvan. Finding and evaluating community structure in networks. *Physical Review*, E 69(026113), 2004.
18. J Reichardt and S Bornholdt. Statistical mechanics of community detection. *Phys. Rev. E*, 74(1):016110, Jul 2006.
19. S B Seidman. Network structure and minimum degree. *Social Networks*, 5:269–287, 1983.
20. SnailVis. `http://cnet.fi.uba.ar/SnailVis/`.
21. M Weber, Marc A, and W Müller. Visualizing time-series on spirals. In *INFOVIS '01: Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*, page 7, Washington, DC, USA, 2001. IEEE Computer Society.