ADAC (2008) 2:3–16
DOI 10.1007/s11634-008-0018-3

REGULAR ARTICLE

Two local dissimilarity measures for weighted graphs with application to protein interaction networks

Jean-Baptiste Angelelli · Anaïs Baudot · Christine Brun · Alain Guénoche

Received: 25 October 2007 / Revised: 20 February 2008 / Accepted: 25 February 2008
Published online: 13 March 2008

Abstract We extend the Czekanowski-Dice dissimilarity measure, classically used to cluster the vertices of unweighted graphs, to weighted ones. The first proposed formula corresponds to edges weighted by a probability of existence. The second one is adapted to edges weighted by intensity or strength. We show on simulated graphs that the class identification process is improved by computing weighted compared to unweighted edges. Finally, an application to a drosophila protein network illustrates the fact that using these new formulas improves the 'biological accuracy' of partitioning.

Keywords Graph distance · Graph partitioning · Heuristic optimisation · Biological networks

Mathematics Subject Classification (2000) 05C12 · 90C35 · 90C59

J.-B. Angelelli · A. Guénoche (✉)
IML, CNRS-Université de la Méditerranée, case 907, Parc Scientifique de Luminy,
13288 Marseille cedex 9, France
e-mail: guenoche@iml.univ-mrs.fr

J.-B. Angelelli
e-mail: angele@iml.univ-mrs.fr

A. Baudot · C. Brun
IBDML, CNRS-Université de la Méditerranée, case 907, Parc Scientifique de Luminy,
13288 Marseille cedex 9, France
e-mail: abaudot@cnio.es

C. Brun
e-mail: brun@ibdm.univ-mrs.fr

1 Introduction

Biological protein–protein interaction networks are modelled by simple undirected graphs; nodes correspond to proteins and edges to physical contact between them. Since proteins interact specifically with each other to insure their function, highlighting groups of nodes in the graph corresponds to identifying classes of proteins involved in the same pathway(s) or cellular process(es). For this, we have previously proposed two clustering methods for the partitioning of large unweighted biological networks (*Prodistin*: Brun et al. 2003; Baudot et al. 2006 and *ClasDens*: Brun et al. 2004). First the Czekanowski-Dice dissimilarity measure (Dice 1945) between vertices is computed. Second, a clustering algorithm is applied to the resulting dissimilarity matrix to identify clusters having an edge density higher than in the whole graph. The principles of this algorithm are different according to the methods: hierarchical for *Prodistin* (BioNJ, Gascuel 1997), density based clustering for *ClasDens* (Guénoche 2004). Both methods have permitted the identification of classes of proteins, the prediction of the function of proteins with an unknown function, and the studies of the modular structure of interaction networks (Baudot et al. 2004; Zhong and Sternberg 2006).

In order to improve the biological meaning of the identified node groups, we propose two natural extensions of the Czekanowski-Dice dissimilarity measure for weighted graphs. The weight of an edge may represent either the probability for the existence of the interaction (determined experimentally), or the intensity of an existing interaction. The two extensions we present here are adapted to the semantic of the weights. They allow us to go beyond the study of the graph structure by integrating additional, secondary information to the graph. We may then expect to refine the relevance of our biological results.

Other computing methods proposed to identify protein clusters in protein-protein interaction networks have already been proposed. They are based on principles deriving from graph partitioning theory: the search for densely connected regions of a graph (Bader and Hogue 2003), the progressive disconnection of the graph using a evaluation of edge betweenness (Girvan and Newman 2002), random walks in the graph (van Dongen 2000; Pons and Latapy 2006) or spectral decomposition (Brandes et al. 2003; Nemwan 2006). Initially they were proposed to unweighted graphs and some of them have also been extended to weighted ones. None of them seems to prove that taking weights into account improves the partitioning process. Recently, Chen et al. (2006) and Chua et al. (2006) also consider a 'functional similarity weight' derived from the Czekanowski-Dice index. They use the product of weight values for edges in the common neighborhood of two vertices and come to a complicated formula, introducing a parameter to correct the situation where the proteins have too few neighbors. Moreover, they do not realize clusters using their formula, but only mention that known functional classes receive an index value larger than the Czekanowski-Dice index.

This article is organized as follows: in Sect. 2, we recall the basic definition of the Czekanowski-Dice dissimilarity, underlining the role and effect of an edge between two vertices. In Sect. 3, we extend this definition in two ways, corresponding to both interpretations of the weights. In Sect. 4, we describe some properties of these dissimilarities and show, by simulation on weighted random graphs, that taking the weights into account permits to recover eventually existing clusters more precisely. In Sect. 5,

two biological examples on a real fly protein interaction network are presented. They illustrate the resulting improvement, and help understanding the effect of weights for the partitioning of protein-protein interaction networks.

2 The Czekanowski-Dice dissimilarity

Let X be a set of n vertices, E a set of m edges (x, y) and $\Gamma = (X, E)$ the corresponding non-directed graph. We assume it is connected; if not, each component will be treated separately. For each subset Y of X, let $\Gamma(Y)$ be the set of vertices outside of Y which are connected to Y:

$$\Gamma(Y) = \{x \in X \setminus Y \mid \exists y \in Y, (x, y) \in E\}$$

and $\overline{\Gamma}(Y) = Y \bigcup \Gamma(Y)$. The neighborhood of x is denoted by $\Gamma(x) := \Gamma(\{x\})$, the degree of x by $Dg(x) = |\Gamma(x)|$ and δ is the maximum degree in the graph. Let $E(Y)$ be the set of internal edges in $Y \subset X$:

$$E(Y) = \{(x, y) \in E \mid x \in Y \text{ and } y \in Y\}.$$

The Czekanowski-Dice dissimilarity of two vertices x and y takes into account the numbers of common adjacent vertices and those which are only connected to x or y. It is defined by

$$D(x, y) = \frac{|\Delta(\overline{\Gamma}(x), \overline{\Gamma}(y))|}{|\overline{\Gamma}(x)| + |\overline{\Gamma}(y)|} = \frac{|P_{spe}(x, y)|}{|P_{tot}(x, y)|} \tag{1}$$

where Δ is the symmetric difference between two sets. It is a dissimilarity measure but not a *true* distance (Fichet and Le Calvé 1984): two connected vertices having only common adjacent vertices have a dissimilarity equal to 0 and this is not consistent with the triangular inequality.

Note that $\overline{\Gamma}$ is used in formula (1) (and not Γ), which is equivalent to add a loop to each vertex. The value of $D(x, y)$ is the ratio of two quantities quantifying the specific part to x and y, denoted $P_{spe}(x, y)$, and the total part, denoted $P_{tot}(x, y)$. If x and y are connected, both vertices are only counted in $P_{tot}(x, y)$, but if not, they are also counted in $P_{spe}(x, y)$. Remark that $\Gamma(x) \bigcap \Gamma(y)$ is counted twice in the total part. Dissimilarity D provides different values if x and y are adjacent or not.

We retain this dissimilarity for several reasons:

- It is very effective in graph partitioning (Kuntz 1992; Guénoche 2005), far better than the shortest path metric;
- It is a *local* dissimilarity since $D(x, y)$ can be computed only from the vertices connected to x or y;
- Each pair of vertices separated by more than 2 edges gets value 1;
- Therefore, it can be computed in time $O(n\delta^3)$.

Example 1 All along this text, we consider the graphs $\Gamma_+(x, y)$ and $\Gamma_-(x, y)$ displayed in Fig. 1 which only differ by the presence/absence of the edge (x, y). In each example, we compute the two corresponding dissimilarity measures, D_+ and D_-, both printed in the same array (Table 1). The diagonal values, equal to 0.0, are not printed.

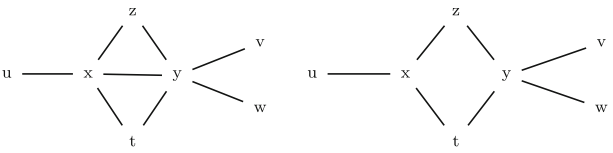

Fig. 1 Two simple graphs $\Gamma_+(x, y)$ and $\Gamma_-(x, y)$ differentiated by pair (x, y)

Table 1 The Czekanowski-Dice dissimilarities for both graphs in Fig. 1, D_+ (lower left part) and D_- (upper right part)

	t	u	v	w	x	y	z
t:		0.60	0.60	0.60	0.43	0.50	0.33
u:	0.60		1.00	1.00	0.33	1.00	0.60
v:	0.60	1.00		0.50	1.00	0.43	0.60
w:	0.60	1.00	0.50		1.00	0.43	0.60
x:	0.25	0.43	0.71	0.71		0.56	0.43
y:	0.33	0.75	0.50	0.50	0.27		0.50
z:	0.33	0.60	0.60	0.60	0.25	0.33	

When (x, y) is absent, $\overline{\Gamma}(x) = \{x, z, t, u\}$ and $\overline{\Gamma}(y) = \{y, z, t, v, w\}$. We have $P_{spe}(x, y) = \{x, y, u, v, w\}$ and $D_-(x, y) = \frac{5}{9} = 0.556$. When (x, y) is present, $\overline{\Gamma}(x) = \{x, y, z, t, u\}$ and $\overline{\Gamma}(y) = \{y, x, z, t, v, w\}$. We have $P_{spe}(x, y) = \{u, v, w\}$ and $D_+(x, y) = \frac{3}{11} = 0.273$. One can see that all the values linked to x or y are different.

3 Two weighted dissimilarities

As we wrote in the introduction, in protein-protein interaction graphs, an edge (x, y) means that there is a contact between the proteins x and y. These interactions are often revealed by experiments which provide for each edge a confidence score that can be interpreted as a probability of existence. Other types of data permit to quantify the intensity of the interaction. In both cases, the value is high when interaction is likely or strong. This information can be coded as a weight function $w : E \to [0, 1]$, admitting that value 0 corresponds to an absence of edge or a null intensity and value 1 to a maximum probability or intensity.

From now on, we consider graphs weighted by a function w and we propose two methods for evaluating the dissimilarity on X. The first one relates to weights that can be interpreted as probabilities, the second one considers that $w(x, y)$ quantifies the intensity of the interaction between x and y. Both are not distance functions since they extend the Czekanowski-Dice formula.

3.1 Weights are probabilities

If weights are probabilities, the dissimilarity, denoted D_p, between each pair (x, y) is linked to the weight of the edge (x, y). It should be a weighted sum of the dissimilarity

Table 2 Dissimilarities D_{p+} (lower left part) and D_{p-} (upper right part) corresponding to the graphs in Fig. 2

	t	u	v	w	x	y	z
t:		0.60	0.78	0.78	0.35	0.40	0.63
u:	0.60		1.00	1.00	0.23	1.00	0.90
v:	0.78	1.00		0.67	1.00	0.68	0.71
w:	0.78	1.00	0.67		1.00	0.68	0.71
x:	0.24	0.31	0.81	0.81		0.65	0.88
y:	0.29	0.81	0.71	0.71	0.43		0.55
z:	0.63	0.90	0.71	0.71	0.68	0.52	

D^*_- corresponding to the graph $\Gamma^*_-(x, y) = \Gamma_-(x, y)$ where there is no edge between x and y, and the dissimilarity D^*_+ corresponding to the graph $\Gamma^*_+(x, y)$ where edge (x, y) has weight 1.

$$D_p(x, y) = (1 - w(x, y)) \times D^*_-(x, y) + w(x, y) \times D^*_+(x, y)$$

But the graphs $\Gamma^*_+(x, y)$ and $\Gamma^*_-(x, y)$ are themselves depending on probabilities of edges in $E(Y)$, with $Y = \overline{\Gamma(x)} \cup \overline{\Gamma(y)}$, and so are $D^*_+(x, y)$ and $D^*_-(x, y)$. Consequently, the calculation should be done by summing over all the subsets of $E(Y)$, each combination being weighted by the product of edge weights in the subset. For computational complexity reasons (enumerating subsets of $E(Y)$), we consider a simpler formula.

1. For $\Gamma^*_-(x, y)$, the specific part contains:
   (a) weights of edges corresponding to vertices exclusively connected to x or y,
   (b) for a vertex s connected to x and y, the difference $|w(x, s) - w(y, s)|$, which corresponds to cases where s is connected to only one vertex,
   (c) both loops on x and y, whose weights are implicitly equal to 1, in order to comply with the Czekanowski-Dice formula.

   Therefore, the weight of the specific part is $|P_{spe}(x, y)| = 2 + \sum_{s \in Y} |w(x, s) - w(y, s)|$. For the total part, we must count $|P_{tot}(x, y)| = 2 - 2 \times w(x, y) + \sum_{s \in Y} w(x, s) + w(y, s)$, where the value 2 corresponds to loops and $2 \times w(x, y)$ corresponds to the fact that edge (x, y) is counted twice in the sum even if absent in this graph (Table 2).
2. For $\Gamma^*_+(x, y)$, the specific part to x and y yields $|P_{spe}(x, y)| = \sum_{s \in Y} |w(x, s) - w(y, s)|$ because loops are not counted since x and y are connected. For the total part $|P_{tot}(x, y)| = \sum_{s \in Y} w(x, s) + w(y, s) + 2 + 2 \times (1 - w(x, y))$, the last term corresponds to the edge (x, y).

Denoting $S(x, y) = \sum_{s \in Y} |w(x, s) - w(y, s)|$ and $T(x, y) = \sum_{s \in Y} |w(x, s) + w(y, s)|$ we obtain a probability-weighted dissimilarity:

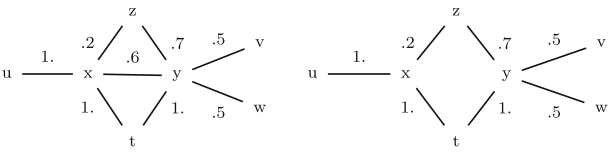

Fig. 2 Weighted graphs $\Gamma_+(x, y)$ and $\Gamma_-(x, y)$

$$D_p(x, y) = (1 - w(x, y)) \times \frac{S(x, y) + 2}{T(x, y) + 2 - 2w(x, y)} + w(x, y) \times \frac{S(x, y)}{T(x, y) + 4 - 2w(x, y)}. \tag{2}$$

Example 2 We use the same graphs as in Fig. 1, weighted as indicated in Fig. 2. We evaluate the D_{p+} and D_{p-} values for both graphs.

For $\Gamma_+(x, y)$ weighted by probabilities:

1. If (x, y) is absent, $|P_{spe}(x, y)| = 2 + w(u, x) + w(y, v) + w(y, w) + |w(x, z) - w(y, z)| + |w(x, t) - w(y, t)|$, $|P_{tot}(x, y)| = 2 + w(x, u) + w(x, z) + w(x, t) + w(y, v) + w(y, w) + w(y, z) + w(y, t)$ and $D^*_-(x, y) = \frac{4.5}{6.9} = 0.652$.
2. If (x, y) is present, $D^*_-(x, y) = \frac{4.5}{6.9}$ as before and for $D^*_+(x, y)$, we have $|P_{spe}(x, y)| = w(u, x) + w(y, v) + w(y, w) + |w(x, z) - w(y, z)| + |w(x, t) - w(y, t)|$, $|P_{tot}(x, y)| = 4 + w(x, u) + w(x, z) + w(x, t) + w(y, v) + w(y, w) + w(y, z) + w(y, t)$; $D^*_+(x, y) = \frac{2.5}{8.9} = 0.281$.

Consequently $D_{p+}(x, y) = 0.4 \times \frac{4.5}{6.9} + 0.6 \times \frac{2.5}{8.9} = 0.429$ and $D_{p-}(x, y)$ as been calculated as $D^*_-(x, y) = \frac{4.5}{6.9} = 0.652$.

3.2 Weights are intensities

If weights are considered to be intensities, the interactions are interpreted as strengths. The specific interactions to x or y move them apart and common interactors bring them closer. Therefore, connections between x and y are divided into two categories:

- attractive strength corresponding to the possible edge (x, y) and, for each vertex s connected to both x and y, the sum of weights $w(x, s) + w(y, s)$;
- repulsive strength corresponding to vertices only connected to x or y, and also loops when x and y are not connected (Table 3).

Let $R(x, y)$ be the repulsive part and $A(x, y)$ the attractive part of pair (x, y). We have

$$R(x, y) = \sum_{s \in \Gamma(x) \setminus \Gamma(y)} w(x, s) + \sum_{s \in \Gamma(y) \setminus \Gamma(x)} w(y, s),$$

$$A(x, y) = \sum_{s \in \Gamma(x) \cap \Gamma(y)} w(x, s) + \sum_{s \in \Gamma(x) \cap \Gamma(y)} w(y, s).$$

$R(x, y)$ corresponds to the symmetric difference $\Delta(\Gamma(x), \Gamma(y))$, as in the numerator in formula (1) and $A(x, y)$ to the union $\overline{\Gamma(x)} \bigcup \overline{\Gamma(y)}$. We note that when $(x, y) \in E$,

Table 3 Dissimilarities D_{i+} (lower left part) and D_{i-} (upper right part) corresponding to the graphs in Fig. 2

	t	u	v	w	x	y	z
t:		0.60	0.67	0.67	0.36	0.40	0.41
u:	0.60		1.00	1.00	0.23	1.00	0.69
v:	0.67	1.00		0.67	1.00	0.42	0.65
w:	0.67	1.00	0.67		1.00	0.42	0.65
x:	0.18	0.31	0.79	0.79		0.58	0.53
y:	0.23	0.75	0.48	0.48	0.25		0.39
z:	0.41	0.69	0.65	0.65	0.35	0.32	

$w(x, y)$ is counted twice in $A(x, y)$. Hence we define a dissimilarity D_i as a simple quantified version of the Czekanowski-Dice dissimilarity:

$$D_i(x, y) = \frac{R(x, y) + 2}{R(x, y) + A(x, y) + 2} \quad \text{if } (x, y) \notin E \tag{3}$$

$$D_i(x, y) = \frac{R(x, y)}{R(x, y) + A(x, y) + 2} \quad \text{if } (x, y) \in E \tag{4}$$

We note that if vertices x and y are more than two edges away from each other, formula (3) is used and the dissimilarity value is always equal to 1, whatever the path length may be.

Example 3 We evaluate the D_i values for both weighted graphs in Fig. 2.

For the graph weighted by intensities:

1. In case $(x, y) \notin E$, $R(x, y) = w(u, x)+w(v, y)+w(w, y)$, $A(x, y) = w(x, z)+ w(x, t) + w(y, z) + w(y, t)$ and $D_{i-}(x, y) = \frac{4}{6.9} = 0.580$.
2. In case $(x, y) \in E$, $R(x, y)$ in unchanged and $A(x, y) = w(x, z) + w(x, t) + w(y, z) + w(y, t) + 2w(x, y)$ and $D_{i+}(x, y) = \frac{2}{8.1} = 0.247$.

4 Properties and efficiency

Since the dissimilarities D_p and D_i are expected to extend the Czekanowski-Dice dissimilarity to weighted graphs, we have to check if both dissimilarity measures return the classical Czekanowski-Dice values when applied to unweighted networks.

Proposition 1 *When each edge has a weight equal to* 1*, values provided by D_i and D_p coincide with the Czekanowski-Dice dissimilarity.*

Proof Two cases:

– When weights are probabilities, D_p formula becomes:

$$\frac{S(x, y) + 2}{T(x, y) + 2} \text{ if } (x, y) \notin E \text{ and } \frac{S(x, y)}{T(x, y) + 2} \quad \text{if } (x, y) \in E,$$

where $S(x, y)$ is the number of edges specific to x or y and $T(x, y)$ the number of edges in $\Gamma(x)$ plus those in $\Gamma(y)$, which corresponds to Czekanowski-Dice formula since the edge (x, y) is counted twice;

– When weights are intensities, in the computation of D_i,

$$R(x, y) = S(x, y) \text{ and } R(x, y) + A(x, y) = T(x, y),$$

providing the same formulas as in the probability case.

Another important fact is the efficiency of this computation. Often interaction networks have several thousands of vertices but $O(n)$ edges, and it is essential to maintain a time complexity linear in n. The main parameter is the maximum degree (for the worst case) or the average degree in practice. The maximum degree δ can be pretty large ($\delta \approx 100$) but it is very small compared to n and the average degree is always ≤ 10.

Proposition 2 *The D_p and D_i dissimilarities can be calculated in time $O(n\delta^3)$.*

Proof To calculate the Czekanowski-Dice dissimilarity in time $O(n\delta^3)$, the graph must be coded by its adjacency lists. For each vertex x, we have to evaluate a maximum of δ^2 values since for a path longer than 2 we have seen that the dissimilarity values are always equal to 1. To determine $D_p(x, y)$ or $D_i(x, y)$, we first establish their specific and common sets of vertices examining at most 2δ vertices connected to x or y. So, the dissimilarity value can be established in $O(\delta)$ and all the values strictly lower than 1 are calculated in time $O(n\delta^3)$. In the case of weighted graphs, we add the corresponding weight lists to the adjacency lists and the same operations (sums and ratios) are performed with the weights. Therefore, the time complexity is identical.

A computer program in C can be requested from the corresponding author. The file graph is just a list of edges, given by a pair of labels with a weight value; the generated distance file is in the standard Phylip format. Vertices can be labelled with any ascii string.

4.1 A simulation process

The classes are built by optimizing the clustering criterion (5) below that is based on dissimilarity values. We want to show that by using weights the existing classes are more accurately reproduced. For generating data, we use a simulation process which has been developed to compare partitioning algorithms in graphs (Guénoche 2005). A precise comparison protocol have been defined. It comprises mainly three tools:

1. A generator of Erdös–Reyni random graphs. Let P be a random partition on X in p classes. Linking randomly and independently some pairs of vertices with larger probabilities within the classes than between them, makes natural classes in the graph. The respective probabilities are denoted p_i and p_e and the partition is more or less evident according to their difference $|p_i - p_e|$.

2. A graph partitioning algorithm optimizing a criterion, denoted *PartOpt* in the following. It belongs to the centroid methods in clustering, but it does not require the definition of centers for classes. Let $P = (P_1, \ldots, P_p)$ be a partition of X in p classes and $P(x)$ denotes the class containing x. Given a partition, the $\Im$ inertia function is the sum, for all the elements x in X, of the squares of the average dissimilarity values $D(x, y)$ to the other elements y belonging to $P(x)$:

$$\Im(P) = \sum_{x \in X} \left(\frac{\sum_{y \in P(x)} D(x, y)}{|P(x)| - 1} \right)^2 . \tag{5}$$

   Our algorithm for minimizing $\Im(P)$ over the set of all the partitions P of X in p classes, is a simple tabu search heuristic. A very recent article (Guénoche 2008) shows that this simple algorithm gives better average results than many methods cited in the introduction. Starting with an initial random partition, it returns a partition Q.
3. Four criteria to measure the closeness of Q to P. The three first ones are evaluated comparing the classes of Q to those of P. A correspondence $\sigma : Q \to P$ is first established and $P_{\sigma(j)}$ denotes the class in P corresponding to class Q_j in Q. The last criterion is based on an editing distance between partitions defined as the minimum number of transfers of an element from one class to another, to turn Q into P.
   (a) τ_a: the percentage of internal edges in P that remain internal in Q:

$$\tau_a = \frac{\sum_{j=1,\ldots,p} |(x, y) \in E(P_j) \text{ such that } Q(x) = Q(y)|}{\sum_{j=1,\ldots,p} |(x, y) \in E(P_j)|}$$

   (b) τ_e: the percentage of elements in Q which also belong to their corresponding class in P:

$$\tau_e = \sum_{j=1,\ldots,p} \frac{|Q_j \bigcap P_{\sigma(j)}|}{|Q_j|}$$

   (c) τ_p: the percentage of joined pairs in Q which are also joined together in P:

$$\tau_p = \frac{1}{p} \sum_{j=1,\ldots,p} \frac{|(x, y) \in Q_j \text{ such that } P(x) = P(y)|}{|Q_j|}$$

   (d) τ_t is the *transfer distance* value $\theta(P, Q)$ divided by n. The transfer distance counts the minimum number of single-element transfers from one class to another one that are necessary to transform P into Q. This distance between partitions has been first proposed by Régnier (1965) then by Day (1981). Recently it has been studied (and bounded according to class cardinality) by Charon et al. (2006). It is well adapted to very close partitions because a small number of transfers reveals partitions that are practically identical.

Table 4 The quality criteria τ_a, τ_e, τ_p, and τ_t for the computed partitions according to the weight average values of the internal and external edges

w_i	w_e	Probability-based D_p				Intensity-based D_i			
		τ_a	τ_e	τ_p	τ_t	τ_a	τ_e	τ_p	τ_t
1.0	0.75	0.99	0.99	0.98	0.02	0.95	0.95	0.91	0.07
1.0	0.90	0.96	0.96	0.93	0.06	0.92	0.93	0.87	0.11
1.0	1.0	0.88	0.90	0.81	0.16	0.88	0.90	0.81	0.16
0.90	1.0	0.86	0.89	0.80	0.18	0.85	0.88	0.78	0.18
0.75	1.0	0.72	0.75	0.63	0.37	0.72	0.75	0.62	0.37

The transfer distance value is obtained realizing a matching of the classes of P onto those of Q which minimizes the sum of symmetrical differences between matched classes.

4.2 Results

To evaluate the influence of weights in our clustering process, we consider graphs with 100 vertices randomly spread among 3 classes with an internal edge probability $p_i = 0.3$ and an external one $p_e = 0.15$. Initially, all the edges get the same weight value $w_i = w_e = 1$. In agreement with our previous results, the four criteria τ_a, τ_e, τ_p, and τ_t yield satisfactory average values over a set of 100 trials, that can be seen in the median row of Table 4.

To measure the weight effect, we let the weights vary successively in the internal and external edges. Let w_i and w_e be the weight average values of the two kinds of edges. When external edges are given a weight $w_e < 1 = w_i$, we hope it will be easier to recover initial clusters of the initial random partition P than with $w_e = 1$. Conversely, if we give to the internal edges the weights $w_i < 1 = w_e$, the cohesion of the classes is weakened and we expect it will be more difficult to recover P.

We first considered the values $w_e = 0.9$ and $w_e = 0.75$, leaving $w_i = 1$ unchanged. In Table 4 we observe that the computed clusters get closer to the clusters of the initial partition. Then we fix $w_e = 1$, and decrease w_i; the predicted effect is that the computed partitions move away from the initial ones.

For weights with an interpretation as probabilities (left part of Table 4) or as intensities (right part), we observed that the difference between initial and computed partitions decreases and the transfer distance increases. Thus, this simulation study tends to prove that clusters established by using weights are more accurate when weights are larger within the classes than between them, and also that the neighborhoods are strengthened in that case.

5 Applications to the drosophila interactome

The edges are weighted with probabilities

Here, we used a Drosophila protein-protein interaction network of 1906 interactions (Formstecher et al. 2005) provided with a confidence score based which is on

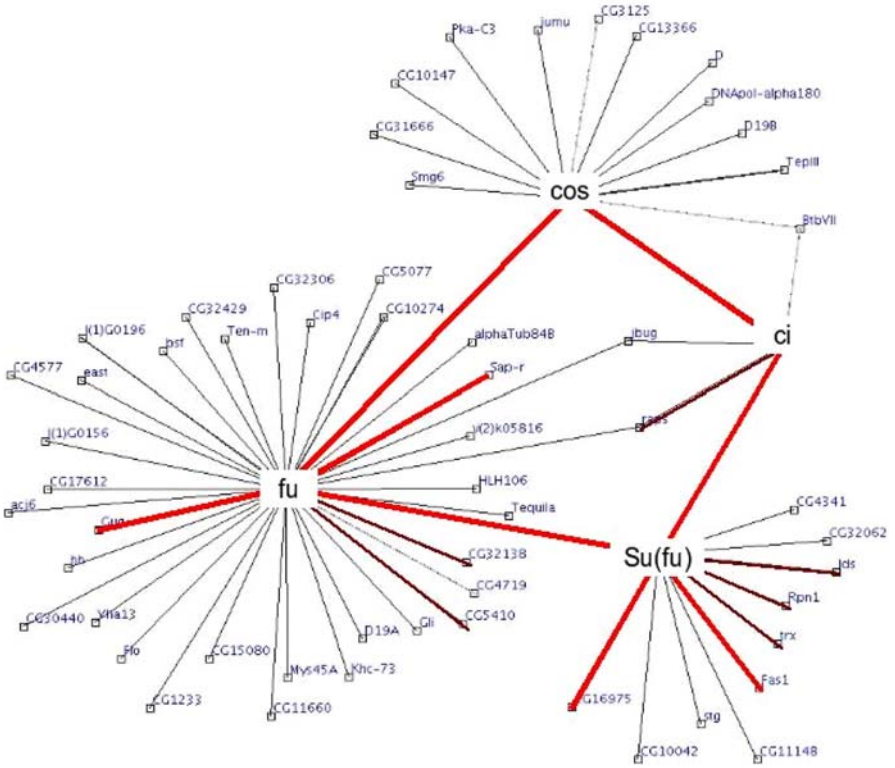

Fig. 3 Interactions around the *fu*, *Su(fu)* and *cos* proteins. Edges are enlarged according to their weight

different statistical parameters. These interactions are subdivided into five categories of decreasing confidence, namely A, B, C, D and E. For our experiment, we weighted the edges of the graph that correspond to interactions scored A with a (maximum) probability of 1, the edges corresponding to an interaction scored B with a probability of 0.8, C with 0.6, D with 0.4 and E with 0.2. These first scores have been empirically chosen for sake of illustration. Then we computed the D_p dissimilarity for a graph weighted with probabilities and obtained a partition using the *PartOpt* method. For comparison purposes, this method has also been applied to the classical CD dissimilarity ignoring the weights. In both cases, we only classified proteins having more than three interactions. From this experiment, we chose an example showing that considering the interaction probabilities may increase the biological relevance of the partition: the proteins *cos*, *fu* and *Su(fu)* are actors of the cytoplasmic part of the Hedgehog signalling pathway. Altogether, they regulate the nuclear translocation and activity of *ci* protein. Interestingly, we noticed that the *fu*, *Su(fu)* and *cos* proteins were not classified together when the non-weighted graph was analyzed, whereas they were joined together, as expected from their biological role, when the weighted graph was studied.

Looking in detail to the network around these three proteins (Fig. 3), it appears that *cos* has 14 interactors (neighbours), among which one is *fu* and only one (*ci*) is shared

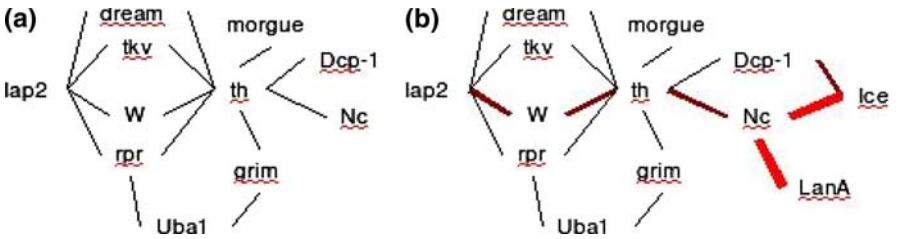

Fig. 4 Interactions between proteins belonging to the cell death class (**a**) without or (**b**) with weights. In **b**, thickness of the edges corresponds to their weight

with *Su(fu)*. When the classical CD dissimilarity is computed from the unweighted graph, the large number of specific interactors of *fu*, *Su(fu)* and *cos* explains the fact that they are separated. When the probabilistic version (with confidence scores) is adopted, the influence of the 12 edges which were previously scattering *cos*, *fu* and *Su(fu)* in different functional clusters, is decreasing because they are poorly weighted (a majority of them have a weight equal to 0.4). The 3 proteins are then classified together according to their biological role. Interestingly, when the edges are weighted with only two values (1 for A, B and C categories, 0.2 for D and E) then the *ci* protein is also found in the cluster. Finally, when the weights are randomly distributed the 4 proteins are, as expected, not clustered together anymore. Altogether, these results suggest that the use of edge weights increases the reliability of the classification result.

The edges are weighted with intensities

Here, a drosophila protein-protein interaction network containing 2849 interactions, has been compiled. The edges were weighted with the results of a transcriptome experiment (kindly provided by L. Perrin, IBDML, Marseille, France). Since the design of a scoring scheme to weight network edges is a research subject on its own, we empirically choose weights reflecting the functional data, for sake of illustration. When the mRNA expression (taken here as intensity) of both interacting proteins is varying significantly in the transcriptome experiment, the edge is weighted with 1, when the mRNA expression of only one out of the two interacting proteins is varying, the edge is weighted with 0.2 and when none of them vary, the edge is weighted with 0.1. The D_i dissimilarity was computed and the *PartOpt* method was used to partition the graph. Here, with an average overlap of 73% between classes, classification changes are noticed compared to the non-weighted graph.

For instance, proteins involved in cell death such as the pro-apoptotic proteins *W*, *rpr* and *grim*, the caspase inhibitors *Iap2* and *th* and the initiator caspase *Nc* are all grouped in the same 'cell death' class (displayed in Fig. 4a) when the non-weighted graph is analyzed. However, the final effector caspase, *Ice*, does not belong to this cluster but is found in another class (not shown) that controls the establishment of cellular localization. When the transcriptome data are taken into account, *Ice* is added to the cell death class because the edge between *Nc* and *Ice* is weighted with 1 (they are co-expressed). So here again, the result suggests that the use of weighted edges has

increased the reliability of the cluster. Furthermore, another protein, *LanA*, is classified in the cell death class only when weights are taken into consideration. Whereas this protein is not known to be involved in cell death in drosophila, its human ortholog was shown to be involved in down-regulation of *IAPs* proteins (Andjilani et al. 2006). Therefore, this result suggests that as in humans, *LanA* might play the role of a cell death regulator in drosophila.

6 Conclusions

The new dissimilarity measures presented in this work allow to analyze weighted networks with distance methods. As shown by the simulations, using weights on graph edges improves the accuracy of the class recovery. Therefore, their use for the analysis of weighted biological networks should improve the quality and the accuracy of the identified functional classes, on the one hand by taking into account the confidence score of the provided interactions, and on the other hand by allowing to integrate additional functional information into the interaction graph. Finally, our proposed modifications may motivate the design of novel partitioning methods for weighted biological networks.

Acknowledgments J.B. Angelelli and Anaïs Baudot are respectively supported by fellowships from the "Ministère de l'Enseignement Supérieur et de la Recherche" and "Association pour la Recherche contre le Cancer". This work is supported by the CNRS ACI IMP-Bio. The authors would like to thank the editors of this Journal for their great help in editing this article.

References

Andjilani M, Droz JP, Benahmed M, Tabone E (2006) Down-regulation of fak and iaps by laminin during cisplatin-induced apoptosis in testicular germ cell tumors. Int J Oncol 28(2):535–542

Bader GD, Hogue CW (2003) An automated method for finding molecular complexes in large protein interaction networks. BMC Bioinf 4:2

Baudot A, Martin O, Mouren P, Chevenet F, Guénoche A, Jacq B, Brun C (2006) Prodistin web site: a tool for the functional classification of proteins from interaction networks. BMC Bioinf 22(2):248–250

Baudot A, Jacq B, Brun C (2004) A scale of functional divergence for yeast duplicated genes revealed from the analysis of the protein–protein interaction network. Genome Biol 5:R76

Brandes U, Gaertler M, Wagner D (2003) Experiments on graph clustering algorithms. In: Di Battista G, Zwick U (eds) ESA 2003. LNCS, vol 2832. Springer, Heidelberg, pp 568–579

Brun C, Chevenet F, Martin D, Wojcik J, Guénoche A, Jacq B (2003) Functional classification of proteins for the prediction of cellular function from a protein–protein interaction network. Genome Biol 5:R6

Brun C, Herrmann C, Guénoche A (2004) Clustering proteins from interaction networks for the prediction of cellular functions. BMC Bioinf 5:95

Charon I, Denoed L, Guénoche A, Hudry O (2006) Maximum transfer distance between partitions. J Classif 23(1):103–121

Chen J, Chua HN, Hsu W, Lee ML, Ng SK, Saito R, Sung WK, Wong L (2006) Increasing confidence of protein–protein interactomes. Genome Inf 17(2):284–297

Chua HN, Sung WK, Wong L (2006) Exploiting indirect neighbours and topological weight to predict protein function from protein-protein interactions. Bioinformatics 22(13):1623–1630

Day W (1981) The complexity of computing metric distances between partitions. Math Soc Sci 1:269–287

Dice LR (1945) Measures of the amount of ecologic association between species. Ecology 26(3):297–302

van Dongen S (2000) Graph clustering by flow simulation. Ph.D. Thesis, University of Utrecht

Fichet B, Le Calvé G (1984) Structure géometrique des principaux indices de dissimilarité sur signes de présence–absence. Stat Anal Donnés 3:11–44

Formstecher E, Arresta S, Collura V, Hamberger A, Meil A, Trehin A, Reverdy C, Betin V, Maire S, Brun C, Jacq B, Arpin M, Bellaiche Y, Bellusci S, Benaroch P, Bornens M, Chanet R, Chavrier P, Delattre O, Doye V, Fehon R, Faye G, Galli T, Girault JA, Goud B, Gunzburg Jde , Johannes L, Junier MP, Mirouse V, Mukherjee A, Papadopoulo D, Perez F, Plessis A, Rosbach M, Ross C, Saule S, Stoppa-Lyonnet D, Vincent A, White M, Legrain P, Wojcik J, Camonis J, Daviet L (2005) Protein interaction mapping: a drosophila case study. Genome Res 15:376–384

Gascuel O (1997) BioNJ: an improved version of the nj algorithm based on a simple model of sequence data. Mol Biol Evol 14(7):685–695

Girvan M, Newman MEJ (2002) Community structure in social and biological networks. Proc Natl Acad Sci USA 99(12):7821–7826

Guénoche A (2004) Clustering by vertex density in a graph. In: Banks D et al Proceedings of IFCS congress classification, clustering and data mining applications. Springer, Berlin, pp 14–24

Guénoche A (2005) Comparing recent methods in graph partitioning. ICGT'05. Electronic notes in discrete mathematics, vol 22, pp 83–89

Guénoche A (2008) Comparison of algorithms in graph partitioning. RAIRO (to appear)

Kuntz P (1992) Représentation euclidienne d'un graphe abstrait en vue de sa segmentation. Ph.D. Thesis, Ecole des Hautes Etudes en Sciences Sociales, Paris

Nemwan MEJ (2006) Modularity and community structure in networks. PNAS 103(23):8577–8582

Pons P, Latapy M (2006) Computing communities in large networks using random walks. J Graph Algorithms Appl 10(2):191–218

Régnier S (1965) Quelques aspects mathématiques des problèmes de classification automatique. ICC Bull 4, reprint (1983) Math Sci Hum 82:13–29

Zhong W, Sternberg PW (2006) Genome-wide prediction of C. elegans genetic interactions. Science 318:1481–1484