Baltic J. Modern Computing, Vol. 4 (2016), No. 4, pp. 876–895
DOI: http://dx.doi.org/10.22364/bjmc.2016.4.4.18

Gene Duplication Models and Reconstruction of Gene Regulatory Network Evolution from Network Structure

Juris VIKSNA[1]*, David GILBERT[2]

[1] Institute of Mathematics and Computer Science, and Faculty of Computing, University of Latvia, Riga, Latvia
[2] College of Engineering, Design and Physical Sciences, Brunel University London, London, UK

juris.viksna@lumii.lv, david.gilbert@brunel.ac.uk

Abstract. In this paper we study evolution of gene regulatory networks from the graph-theoretic perspective. We consider two gene duplication models that are based on those studied before, but are more general and/or mathematically more precise than previously published schemes. Our aims are to assess the biological appropriateness of the proposed models and to study the possibilities of reconstruction of the evolution history of networks solely on the basis of network topology.

For one of the proposed models, which is fully deterministic, we provide an exact algorithm for reconstruction of evolutionary history of the network. The algorithm is applicable in real time to networks with up to 200 genes, which is comparable to sizes of real biological networks. The other proposed model involves random deletions of gene interactions. In this case a heuristic modification of the algorithm can be used to identify a large subset of genes that have been duplicated during the last duplication event.

The methods have been tested for analysis of yeast gene regulatory network and have been able to identify several biologically confirmed pairs of duplicated genes. Similarity between inferred pairs of gene duplicates is shown to be above average, thus indicating that traces from gene duplications, which have occurred long time ago, can still be detected from the network topology alone.

Keywords: Gene regulatory networks, Evolution of biological networks, Graph algorithms.

* The work was supported by Latvian Council of Science grant 258/2012 and Latvian State Research programme project NexIT (2014-2017).

1 Introduction

In the last decade bioinformatics has witnessed the emergence of high throughput experimental techniques allowing simultaneous measurements of activity of many genes, thus giving a possibility to reconstruct gene regulatory networks from experimental data. As a result, there has arisen interest in the properties of gene regulatory networks and in the ways how such networks might have evolved.

One of the first noticed properties of gene regulatory networks was that the distribution of vertex degrees tends to correspond to the so-called power law (Barabasi and Albert (1999), Sidow (1996), Watts and Strogatz (1998), Wolf *et al* (2002)), i.e. the number of genes with k connections is roughly equal to $ck^{-\beta}$ for some constants c and β. At the same time, gene networks also have properties that distinguish them from random networks with the power law vertex degree distribution (e.g. gene networks possess some modularity, which can be detected by distribution of clustering coefficients (Ravasz *et al* (2002)) being different from distribution in random networks). Consequently, several models for the possible evolution of gene regulatory networks leading to networks with similar properties as those obtained in experiments have been proposed (Chung *et al* (2003), Milo *et al* (2002), Ravasz *et al* (2002), Wagner (1994)). At the same time, without doubt there is an underlying biological process governing the evolution of networks, on basis of which several scenarios of network evolution have been proposed (Babu *et al* (2004), Friedman and Hughes (2003), Teichmann *et al* (2001)). Practically all of these scenarios are based on some process of gene duplication, preserving or duplicating some of the existing connections and losing some other.

In Ravasz *et al* (2002) a semi-formal model of ’hierarchical’ networks has been proposed, which generally involves replications of all network genes, preserving connections within each of replicas as well as adding connections to one particular ’central node’. It is shown (using simulation experiments) that such hierarchical networks have vertex degree and clustering coefficient distributions similar to those of biological networks.

A more formal approach is taken in Chung *et al* (2003). Here the authors consider two network evolution models: ’duplication model’ in which within each step exactly one network node is duplicated, duplicating also all of its connections, and ’partial duplication model’, with the difference from the duplication model being that each of the duplicated nodes connections is duplicated just with some probability p. It is proven that partial duplication model has the power law vertex degree distribution. Similarly as in the model considered in Ravasz *et al* (2002) the networks studied here are undirected graphs.

Besides that, already in 1994 a rather complicated duplication model has been analysed by Wagner (1994), before genome wide networks from high throughput experiment results even became available. This model is somewhat similar to the full and to the partial duplication models we are introducing here, the main difference is that in each step exactly one gene is duplicated together with all of its connections. One of the results obtained on the basis of this model is that the evolution of gene networks should preferentially occur either by duplication of single genes or by duplication of all genes in a network, the preference for such a behaviour being consistent with experimental observations in biology.

A number of authors have explored more biological approach to the problem, studying known biological networks and trying to infer from them potential scenarios of evolution. Usually in these cases a distinction is made between 'general' genes and the known transcription factors (as a consequence the networks considered are directed graphs). Such approach was used, for example, in Babu *et al* (2004), where the authors have analysed several known biological networks. Their results indicate that it is likely that duplications of single genes or gene and transcription factor pairs have occurred in the past together with duplication and random loss of their interactions.

More complicated network model (including also metabolic pathways) has been considered in Teichmann *et al* (2001) and several possible duplication events have been identified, although the results appear to be based on manual pathway analysis.

A scheme for single gene duplication has been also proposed in Sidow (1996), giving greater attention to duplication and loss of gene interactions. However, this is largely a qualitative proposal, not directly based on quantitative observations. More recently schemes of very similar type have been described by Thompson *et al* (2015), in this case also discussing possible biological mechanisms that could lead to gene regulatory changes. The paper also proposes several variants of pipelines of bioinformatics software packages that could be used for analysis of gene regulatory changes (however, without testing them either on real or on simulated data).

Comparatively recent review paper about underlining biological mechanisms for evolution of gene regulation has been published by Romero *et al* (2012). In context of next generation sequencing (NGS) data becoming available, Garfield and Wray (2010) discusses how the known bioinformatics techniques could be used to study gene regulation evolution from NGS data (however, also without presenting concrete user cases).

One of the first attempts to use real experimentally obtained data sets to study evolutionary changes in gene regulation is probably by Friedman and Hughes (2003). The authors have analysed several biological networks (*C.elegans*, *Drosophila* and *S.cerevisiae* (yeast)) and have found a confirmation of duplication events in two of them (*C.elegans* and *S.cerevisiae*). The results are based on analysing similarity between genes belonging to particular genome blocks. There is also an earlier study by Gu *et al* (2002)) that tries to estimate the number of duplicated genes in yeast, but does not directly attempt to confirm or reject the duplication hypothesis.

Yeast genome is particularly attractive for study of evolution of gene regulatory changes, since it is widely believed that during the evolution yeast genome has undergone a full duplication. The most extensive analysis of yeast genomes from such a perspective is probably by Thompson *et al* (2013), where evolutionary histories of 15 species of yeast are compared with changes in gene regulation (at the level of individual genes for a number of few well known regulatory motifs and also, more indirectly, at the genome wide level by comparing gene expression patterns).

In this paper we consider two gene duplication models that are based on those proposed before, but are more general than previously published mathematically well-defined models, and mathematically more precise compared to more complicated, but less formally described schemes. Our aims are to assess the biological appropriateness of the proposed models (this is done by computer simulations) as well as to study the possibilities of reconstruction of the evolution history of gene regulatory networks from the given final states. The reconstruction is attempted solely on the basis of network topology, not considering similarity between particular genes. (Thus, this is a kind of opposite approach to that used in Babu *et al* (2004).)

One of the proposed models (FDM) is fully deterministic and for this model we provide an exact algorithm for reconstruction of evolutionary history. Although the algorithm has exponential running time (the reconstruction problem itself is presumably **NP**-complete), it works well in practice on random networks with up to 200 genes (this is comparable to sizes of many gene regulatory networks that are analysed by biologists). The other proposed model (PDM) involves random deletions of gene interactions, and the prospects of unambiguous and/or computationally efficient reconstruction of full evolution history of such networks seems unlikely. However, a number of heuristics can be used that at least are able to identify a large subset of genes that have been duplicated during the last duplication event. Similar heuristic approaches could be applied also to noisy networks (i.e. networks involving random loss or emergence of gene interactions).

The methods have been applied to analysis of yeast regulatory network (Lee *et al* (2002)) and the results indicate that traces from gene duplications, which have occurred long time ago, can still be detected from the network topology alone. The methods also have been able to identify several biologically confirmed pairs of such duplicated genes.

2 Models of network evolution

We consider two gene duplication models for gene regulatory network evolution. Generally these models are similar to those considered before and briefly discussed in the previous section. Our motivation to select these particular modifications was partially influenced by our aim to study the reconstruction possibilities of network evolution. Therefore we tried to distinguish between the duplication process as such and the following insertions and deletions of regulatory relations, which under most models are largely treated as 'noise'. On the basis of the results in Ravasz *et al* (2002) and in Wagner (1994) it also seemed to be important to consider models that allow simultaneous duplications of several genes together with the regulatory relations involving these genes.

For the purpose of this paper gene regulatory network will be defined simply as a directed graph $N = (G, R)$, where $G = \{1, \ldots, n\}$ is a set of genes (graph vertices) and $R \subseteq G \times G$ is a set of regulatory relations (graph edges). The presence of edge $(g_1, g_2) \in R$ means that the gene g_2 is regulated by the gene g_1 – this could mean that the increased activity of g_1 either increases or decreases the activity of g_2, or, more generally, that g_1 is one of the arguments of some function that regulates the activity of g_2. We are focusing on the topology of networks and not on the properties of specific genes, thus the labels (indices) of individual genes in G are used only as a technical convenience and we will consider two networks N_1 and N_2 to be equal if they are isomorphic as *unlabelled graphs*.

2.1 Full duplication model (FDM)

In this model we assume that in each step a subset of network genes is being duplicated in such a way that all gene interactions within the duplicated part and their connections to non-duplicated part of the network are preserved. From the biological perspective this model reasonably well describes what happens immediately after the duplication of part of the genome. Note, however, that, in line with the proposals of several other authors, to preserve network 'modularity' duplicated parts themselves remain disconnected in FDM, which might or might not be the case for real biological networks, depending on specific biological events leading to a particular duplication.

Definition 1. Given network $N = (G, R)$ with $G = \{1, \ldots, n\}$, a *duplication event* D is a function that for some non-empty set $S(D) = \{g_1, \ldots, g_k\} \subseteq G$ maps network N to network $D(N) = (G \cup T(D), R')$, where $T(D) = D(g_1), \ldots, D(g_k)$, $D(g_i) = n + i$, and $R' = \{(x, y) \mid (G(x), G(y)) \in R \wedge \neg(x \in S(D) \wedge y \in T(D)) \wedge \neg(y \in S(D) \wedge x \in T(D))\}$ (where $G(x) = x$ for $x \leqslant n$ and $G(x) = D^{-1}(x)$ for $x > n$).

If we start with a network N, the evolution process involving duplications $D_1, \ldots, D_m$ will map network N to network $D_m(\ldots D_2(D_1(N)))$. Depending on the genes involved several different relationships between two duplication events D_i and D_j are possible. In discussing these we assume that D_i has occurred before D_j (i.e. $i < j$) and $N = (G, R)$ contains n genes.

If $S(D_i) \subseteq G$, $S(D_j) \subseteq G$, $S(D_i) \cap S(D_j) = \varnothing$, and there are no $(g_1, g_2) \in R$ with $g_1 \in S(D_i)$ and $g_2 \in S(D_j)$, or with $g_2 \in S(D_i)$ and $g_1 \in S(D_j)$, then events are *independent* and can be combined, i.e. there is a duplication event D such that networks $N_1 = D_i(D_j(N))$, $N_2 = D_j(D_i(N))$ and $N_3 = D(N)$ are isomorphic as unlabelled graphs.

If only the first three conditions $S(D_i) \subseteq G$, $S(D_j) \subseteq G$, $S(D_i) \cap S(D_j) = \varnothing$ hold, or alternatively, if $S(D_j) \subseteq S(D_i) \cup T(D_i)$ and $S(D_j)$ contains exactly one element from each pair from the set $\{\{x, D_i(x)\} \mid x \in S(D_i)\}$, then events are *interchangeable* and their order can be changed without the change of the topology of the resulting network, i.e. $N_1 = D_i(D_j(N)))$ and $N_2 = D_j(D_i(N))$ are isomorphic as unlabelled graphs. All independent events by definition are also interchangeable.

If events are not interchangeable we say that they are *dependent*. The motivation to distinguish between these categories is the following: to obtain the same result independent events can be applied in any order or both simultaneously; interchangeable events can be applied in any order (but not simultaneously); dependent events can be applied only in the order of their occurrence. On the basis of this we can define an evolution graph that describes duplication events that have occurred during the evolution of network.

Definition 2. Given network N and the sequence of duplication events $S = D_1, \ldots, D_m$ in the order in which they have occurred, we say that directed graph $E(S) = (\{0, \ldots, m+1\}, P \cup C \cup B)$ with three types of edges $P, C, B \subseteq \{0, \ldots, m+1\} \times \{0, \ldots, m+1\}$ is an *evolution graph* of N, where:

1. $(i, j) \in P \Leftrightarrow i < j$ and D_i and D_j are dependent and there is no k with $i < k < j$ such that both pairs D_i,D_k and D_k,D_j are dependent;

2. $(i, j) \in C \Leftrightarrow D_i$ and D_j are interchangeable;
3. $(i, j) \in B \Leftrightarrow$ either $i = 0$ and D_j is independent or interchangeable with D_k for all $k < j$, or $j = m + 1$ and D_i is independent or interchangeable with D_k for all $k > i$.

Given network N and a sequence of duplication events S, by $N' = D(N, E(S))$ we denote a network obtained from N by applying these duplication events in the order specified by S. Note that whilst the network N' is already uniquely defined by the initial network N and the sequence of duplication events S, the same (up to isomorphism) network N' could be obtained from different sequences S_1 and S_2 of duplication events from some given set and, correspondingly, by different evolution graphs $E(S_1)$ and $E(S_2)$.

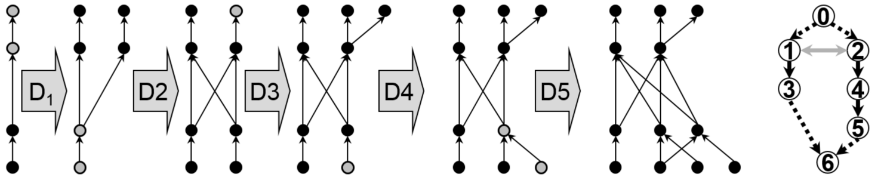

Fig. 1. A sequence of 5 duplication events and the corresponding evolution graph. Graph vertices that are duplicated by the next duplication event are denoted with grey dots. In evolution graph edges from P are shown with black, edges from C with grey and edges from B with dashed lines. By definition edges from C are bidirectional.

Informally an evolution graph shows all duplication events that have occurred for a given network and relations between those events. In addition it contains two extra vertices 0 (source) and $m + 1$ (sink) that could be characterized as "all events independent of $D_1, \ldots, D_m$ and occurring before them" and "all events independent of $D_1, \ldots, D_m$ and occurring after them"; the source vertex is connected by B-edges to all the vertices without incoming P edges and the sink vertex is connected by B-edges from all the vertices without outgoing P edges. In this way evolution graphs describe all the intermediate networks obtainable from N by subsequences of events from $S = D_1, \ldots, D_m$. An example of evolution graph is shown in Figure 1.

Two evolution graphs $E(S_1)$ and $E(S_2)$ involving sequences S_1 and S_2 of duplication events (not necessarily with elements from the same set) are *equivalent*, if $D(N, E(S_1))$ and $D(N, E(S_2))$ are isomorphic as unlabelled graphs. N is *irreducible* if there do not exist N', S and E, such that $N = D(N', E(S))$, i.e. network N is irreducible if it can not be obtained by a sequence of duplication events from any other network.

Evolution graphs raise a number of interesting questions about their properties. Firstly, an evolution graph equivalent to $E(S)$ can be obtained from $E(S)$ by changing the order of duplication events in the sequence S (the allowed changes depend on which events are independent and which events are interchangeable). It is not yet known whether (and how) two equivalent evolution graphs $E(S_1)$ and $E(S_2)$ can be conveniently characterized simply in terms of 'manipulation' of the elements of sequences S_1 and S_2. Apart from the allowed changes of the order of events, the equivalence will also be preserved by combining two successive independent events into a single one. However, the problem starts to get complicated already by the fact that for some networks there exist sets of several (more than 2) dependent events that can be combined into a single duplication event.

Secondly, an interesting question is whether there exist two different (non-isomorphic) irreducible networks N_1 and N_2 and evolution graphs $E(S_1)$ and $E(S_2)$, such that the resulting networks $D(N_1, E(S_1))$ and $D(N_2, E(S_2))$ are isomorphic. Although we do not have an example of such a pair of non-isomorphic networks, there are reasons to suspect that such pairs might exist. However, our experiments show that in practice such pairs of networks are at least quite rare (none were detected during the computational experiments of reconstruction of 200 random evolution graphs, each of them involving around 50 duplication events).

2.2 Partial duplication model (PDM)

Whilst there is a good biological motivation behind the FDM, it has some problems. Firstly, in networks evolving according to this model vertex degree distribution corresponds quite badly to the power law (see Section 4). Besides that, since edges in both directions are treated in the same way, the evolution leads to networks having vertices with large number of incoming edges, which does not correspond well to biological reality (the number of gene regulators are usually assumed to be comparatively small, at the same time the number of genes affected by a particular known gene regulator often tend to be large).

In order to deal with this we can treat the incoming and outgoing edges differently – after a duplication event all connections within the duplicated part as well as all incoming connections are preserved, however, all outgoing connections (to the set of non-duplicated genes) compete between the initial connections and the duplicated ones, as a result only one connection from the each pair is preserved. Apart from giving 'nicer' results (the model corresponds well to the power law distribution and the maximal number of incoming edges never increases), there seems to be a biological justification behind this: since gene regulators are often involved in regulation of several genes, to preserve 'useful connections' binding sites have to be more flexible to adapt to mutations. As a result, it is less likely that they will interact too long with two competing and independently mutating regulators. Thus, PDM might be adequate for description of the situation that could be observed in a comparatively short term after the duplication of part of the genome. Formally duplication events for PDM are defined as follows.

Definition 3. Given network $N = (G, R)$ with $G = \{1, \ldots, n\}$, a *partial duplication event* D is a function that for some non-empty set $S(D) = \{g_1, \ldots, g_k\} \subseteq G$ maps network N to network $D(N) = (G \cup T(D), R')$, where $T(D) = D(g_1), \ldots, D(g_k)$, $D(g_i) = n + i$, and R' is obtained by randomly, with probability $p = 1/2$, removing from the set $\{(x, y) \mid (G(x), G(y)) \in R \neg(x \in S(D) \wedge y \in T(D)) \wedge \neg(y \in S(D) \wedge x \in$

$T(D))\}$ (where $G(x) = x$ for $x \leqslant n$ and $G(x) = D^{-1}(x)$ for $x > n$) exactly one edge from each pair $\{(x,z),(y,z)\}$ with $x \in S(D)$, $y \in T(D)$ and $z \notin S(D)$, $z \notin T(D)$.

Evolution graphs for this model can be defined in a similar way as for FDM. An example of duplication of two genes according to both FDM and PDM models is shown in Figure 2.

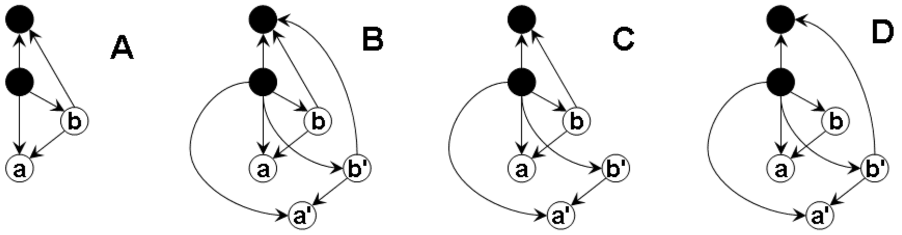

Fig. 2. Initial network A and the resulting networks after genes a and b are duplicated according to FDM (B) and PDM (C or D).

3 Reconstruction of network evolution

We are interested in reconstruction of duplication events that have transformed an initial network N into network N'. Several problems can be considered in this context for both FDM and PDM models.

Reconstruction of the full Evolution Graph (EGP). For a given network N' find an irreducible network N, the sequence of duplication events $S = D_1, \ldots, D_m$, and the corresponding evolution graph $E(S)$, such that $N' = D(N, E(S))$. From the biological perspective this would mean reconstruction of the complete history of evolution of gene regulation process. Due to noise such reconstruction clearly seems infeasible for real biological networks, thus the problem is largely of theoretical interest – how unambiguously and how efficiently such a reconstruction can be done for the described gene duplication models? Nevertheless, the experiments with yeast network (see Section 4) seem to suggest that even after a long evolution process some topological properties of N are still weakly preserved in the evolved network N'.

Reconstruction of a Duplication Event (DEP). For a given network N' find a network N and a duplication event D, such that $N' = D(N)$. Largely this can be regarded as an intermediate technical problem for solving LDEP described below.

Reconstruction of one of the Largest Duplication Events (LDEP). For a given network N' find a network N with the smallest possible number of genes, and a duplication event D, such that $N' = D(N)$ (in general such N might not be unique). This is likely the most practical and interesting problem from the perspective of biological applications – finding the latest duplication event (or, probably, by repeated solving of LDEP, finding several most recent events) of the genome and the corresponding gene regulation changes that have occurred as a result of such event(s). In the real biological networks the anticipated amount of noise will still be large (due to both, gene regulation changes that have occurred for other reasons than genome duplications, and (almost certain) biological inaccuracies of the proposed duplication models). However, the solving of LDEP for such real networks could be made much more feasible with the use of other additional information (notably about gene similarity), which is easily available for biological networks, but which we are not considering here. The larger issue for analysis of real biological networks very likely could be the validation of the predicted genome duplications – with observational data at the time-scale of biological evolution simply not being available, it can be done only indirectly. At the same time, solving LDEP could be applied to studies of statistical significance of a number of different proposed changes of regulatory patterns (e.g. such as described in Teichmann *et al* (2001) or Thompson *et al* (2015)).

3.1 Network reconstruction using full duplication model

Regarding the theoretical complexity of all three problems, it is easy to see that all of them are in class **NP**. Our approach for solving DEP involves using graph isomorphism solver for partitioning graph vertices into automorphism orbits as well as some steps of exhaustive search. Solving LDEP additionally involves solving a problem similar to INDEPENDENT SET, and solving EGP can be reduced to repeated solving of LDEP until we get an irreducible network. However, neither the reduction to SUBGRAPH AUTOMORPHISM nor to INDEPENDENT SET is currently known, thus we can only conjecture that DEP is at least as hard as SUBGRAPH AUTOMORPHISM and that LDEP and EGP are **NP**-complete.

Below we describe *FindLargestDuplicationEvent* algorithm for solving LDEP, which is likely the problem that is the most relevant for real biological applications. With simple modifications the algorithm can be adapted also for generating all the possible solutions for DEP. EGP can be solved by repeated solving of LDEP until we get an irreducible network.

The reconstruction process is based on finding pairs of genes that can be mapped one to another by graph automorphisms.

Definition 4. Given network $N = (G, R)$ and vertex $g \in G$, the orbit $o(g)$ is the largest subset of G, such that for all $g' \in o(g)$ there is an automorphism of graph N mapping g to g'.

By definition $o(g) = o(g')$ for all $g' \in o(g)$ and we can partition G into a set of disjoint orbits $o(G) = \{O_1, \ldots, O_s\}$, such that $G = O_1 \cup \ldots \cup O_s$. Duplicated pairs of vertices clearly should belong to non-singleton orbits, thus when looking for duplication candidates we are interested in considering only the subset of orbits $O(G) = \{O \in o(G) \mid |O| > 1\}$ and the subset of vertices $G_O = \{g \in G \mid \exists O \in O(G) : g \in O\}$.

The fact that two vertices v_1 and v_2 belong to the same orbit O does not, however, guarantee that there is a duplication event producing this pair of vertices. To decide whether this really is the case it is useful to construct a refinement of partition $O(G)$.

For $N = (G, R)$ and set of vertices $A \subseteq G$ by N_A we denote the subgraph of N induced by the set of vertices A.

Consider the subgraph N_{G_O} and let $C(G_O) = \{C_1, \ldots, C_t\}$ be the partition of N_{G_O} into connected components (some connected components may contain only a single vertex). We say that a connected component C *spans* a set of orbits $A \subseteq O(G)$ if $C \cap O \neq \emptyset$ for all $O \in A$. Due to the properties of automorphisms, if a component C_1 spans set of orbits $A \subseteq O(G)$ and a component C_2 spans a subset $B \subseteq A$, then C_2 must also span the whole set A. Moreover, the connected components C_1 and C_2 must be isomorphic.

Therefore we can partition the set of orbits $O(G)$ into subsets, such that two orbits are spanned by a connected component C if and only if they belong to the same set of this partition. Let this partition be $S(G_O) = \{S_1, \ldots, S_p\}$. For $O \in O(G)$ by $S(O) \in S(G_O)$ we denote the set with $O \in S(O)$ and define $G_{S(O)} = \{g \in G \mid \exists O' \in S(O) : g \in O'\}$.

For network $N = (G, R)$, $G' \subseteq G$ and vertex $v \in G$ by $E(v, G')$ we denote the set of all vertices from G' connected with v, i.e. $E(v, G') = \{w \in G' \mid (v, w) \in R \vee (w, v) \in R\}$.

Finally we construct refinement $P(G_O) = \{P_1, \ldots, P_q\}$ of the initial partition into orbits $O(G)$ in the following way. Each set of partition $P \in P(G_O)$ is a subset of some orbit $O \in O(G)$ denoted by O_P. Two vertices $v_1, v_2 \in O$ are placed in the same subset P if and only if $E(v_1, G - G_{S(O)}) = E(v_2, G - G_{S(O)})$. That is, we put two vertices of orbit O in the same subpartition of the refinement if and only if their neighbour vertices are the same, apart from the vertices from the orbits spanned by the same connected component C.

Refinement $P(G_O)$ places additional restrictions on pairs of vertices $v_1 \in S(D)$ and $v_2 \in T(D)$ that can be involved in the same duplication event D. Since, by definition of D, v_1 and v_2 must be connected to the same set of vertices from $G - (S(D) \cup T(D))$, they both must belong to the same subpartition $P \in P(G_O)$. The converse, however, is still not necessarily true.

At this stage it is useful to identify a subset of vertices that contains the entire connected components, which could be created by a single duplication event D without affecting any other vertices that might be involved in D. Let $\hat{C} \subseteq C(G_O)$ be a maximal set of connected components such that all $C \in \hat{C}$ span the same set of partitions $\hat{P} \subseteq P(G_O)$. Then for any two components $C_1, C_2 \in \hat{C}$ we can put all the vertices of C_1 in $S(D)$ and all the vertices of C_2 in $T(D)$ without affecting involvement of any other vertices in D. Thus, altogether we can put in each of the sets $S(D)$ and $T(D)$ all the vertices from $\lfloor |\hat{C}|/2 \rfloor$ components from $\hat{C}$ (the assignment of all vertices of a particular component to either $S(D)$ or $T(D)$ can be random).

For any two maximal sets of connected components $\hat{C_1}, \hat{C_2} \subseteq C(G_O)$ with the property described above we can include vertices from components $\hat{C_1}$ and $\hat{C_2}$ in $S(D)$ and $T(D)$ independently. Let D_C be a duplication event obtained by inclusion in each of the sets $S(D_C)$ and $T(D_C)$ all the vertices from $\lfloor |\hat{C}|/2 \rfloor$ components from all the maximal sets $\hat{C} \subseteq C(G_O)$ with the property that all $C \in \hat{C}$ span the same set of partitions $\hat{P} \subseteq P(G_O)$.

Construction of D_C can be regarded as the first (and computationally the easiest) step of solving LDEP, in which we identify all the vertices that can be included in the largest duplication event without affecting inclusion of any others. This will include all the pairs of vertices that might be created by duplication events involving just a single vertex. The benefit of this step is that it reduces the number of vertices for which deciding whether these could or should be included in the same duplication event is a more difficult problem.

After the construction of D_C the remaining set of vertices that might be involved in duplication event is $G_C = G_{S(O)} - (S(D_C) \cup T(D_C))$ with partitioning $Q(G_C) = \{Q_1, \ldots, Q_r\}$, where for each i: $|Q_i| > 1$ and there is $P_j \in P(G_O)$, such that $Q_i = P_j \cap G_C$. It is easy to observe that: 1) for no connected component C of the subgraph N_{G_C} all the vertices of C can be included in the same duplication event D, and 2) if $v_1 \in S(D)$ and $v_2 \in T(D)$ for some event D, then v_1 and v_2 must belong to the same connected component C of the subgraph N_{G_C}. For further analysis of the pairs of vertices that can be included in the largest duplication event it is convenient to use *reconstruction graphs* defined as follows.

Definition 5. Given network $N = (G, R)$ with G_C and $Q(G_C) = \{Q_1, \ldots, Q_r\}$ constructed as described above, we say that an undirected graph $R(N, G_C) = (\{1, \ldots, r\}, X \cup Y)$ with two types of edges $X, Y \subseteq \{\{a, b\} \mid a, b \in \{1, \ldots, r\}\}$ is a *reconstruction graph* of N, where:

1. $\{i, j\} \in X \Leftrightarrow$ there is and edge in N between Q_i and Q_j and $Q_i \cup Q_j$ consists of at least two connected components in $N_{Q_i \cup Q_j}$;
2. $\{i, j\} \in Y \Leftrightarrow$ there is and edge in N between Q_i and Q_j and $Q_i \cup Q_j$ consists of a single connected component in $N_{Q_i \cup Q_j}$.

Reconstruction graphs provide useful information that characterizes all the possible vertices (duplication events) and all the possible edges between interchangeable events that might appear at the 'lowest level' of evolution graph of N (i.e. for each possible evolution graph of N the set of vertices that are connected by B-edges to sink vertex in this particular graph). A sample of reconstruction graph is shown in Figure 3.

It is easy to show that:

1. If two sets $Q_1, Q_2 \in Q(G_C)$ are connected by edge from X, then there is a duplication event D involving vertices from both of these sets, i.e. $S(D) \cap Q_1 \neq \emptyset$ and $S(D) \cap Q_2 \neq \emptyset$. Since, by definition of X edge, there should be at least two isomorphic connected components C_1 an C_2 in $N_{Q_1 \cup Q_2}$ that span both Q_1 and Q_2, then we can take $S(D) = C_1$ and $T(D) = C_2$.
2. If two sets $Q_1, Q_2 \in Q(G_C)$ are connected by edge from Y, then there is no duplication event involving vertices from both of these events, i.e. for all D either $S(D) \cap Q_1 = \emptyset$ or $S(D) \cap Q_2 = \emptyset$. If, to the contrary, we assume that there is such an event D, then $S(D)$ and $T(D)$ must be isomorphic and disconnected in $N_{Q_1 \cup Q_2}$, which contradicts the definition of Y edge.

Thus, to find the largest number of vertices of G_C that can be involved in a single duplication event we can initially construct reconstruction graph $R(N, G_C)$ and then partition it into connected components $R_1, \ldots, R_s$ taking into account both X and Y edges. For R_i let D_{R_i} be a duplication event with the largest possible number of vertices in $S(D_{R_i})$ from the partitions that correspond to the vertices of R_i. Since inclusion

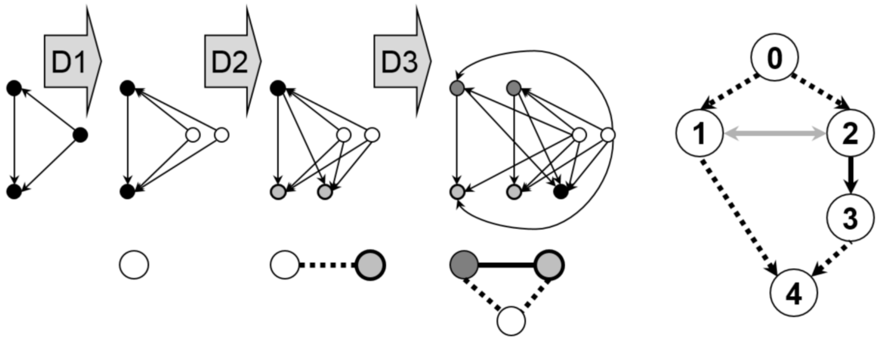

Fig. 3. A sequence of 3 duplication events (left, top), the corresponding evolution graph (right) and reconstruction graphs (left, bottom) for the network obtained after each of 3 duplication events. In this simple case the partitioning $Q(G_C)$ is the same as $O(G)$. In networks vertices belonging to different orbits are shown in different shades, one-vertex orbits are black. In reconstruction graphs edges from X are black and edges from Y are dotted, the shades of vertices match the shades of the corresponding orbits.

in duplication event of vertices from different components R are independent, we can construct the largest duplication event D_R involving vertices of G_C by taking $S(D_R) = \bigcup_i S(D_{R_i})$. The largest duplication event D_{max} for the whole network N then can be obtained by taking $S(D_{max}) = S(D_C) \cup S(D_R)$.

However, there does not seem to be a computationally efficient way for finding sets $S(D_{R_i})$. The problem of finding these sets has some similarities to INDEPENDENT SET problem and some specialised algorithms for the latter probably can be adapted for this purpose. At the same time, in practical computational experiments that we have performed on reconstruction of network evolution (see Section 4) the sizes of components R_i were very small and there was no need for more elaborated approaches for finding $S(D_{R_i})$ than exhaustive search (involving few simple heuristic rules). The algorithm *FindLargestDuplicationEvent* is summarised in pseudo-code form as Algorithm 1.

Algorithm 1 contains two computationally non-trivial steps: *Step 2* involving computation of automorphism orbits and *Step 11* (repeated r times for each R_i separately) involving exhaustive search. All the other steps can be computed in polynomial time (at most $O(|G|^3)$ for *Step 5*, the majority of steps requiring just linear time).

The algorithm has been implemented by adapting a well-known program package **nauty** (McKay (2013)) for partitioning vertices in orbits. In simulation experiments the computation of automorphism orbits turned out to be computationally the most expensive part of the algorithm and limited its application to networks with sizes of around 200 vertices. At the same time, the computations of D_{R_i} were comparatively fast in practice (largely due to small sizes and simple structure of components R_i) and never posed a computational efficiency issue for networks used in simulation experiments.

Algorithm 1 Algorithm for finding the largest duplication event in given network

```
procedure FindLargestDuplicationEvent($N = (G, R)$)
    Compute the orbits $O(G) = \{O_1, \ldots, O_s\}$ and set of vertices $G_O$
    Compute the set of connected components $C(G_O)$
    Compute the partition of orbits $S(G_O)$
    Compute the refinement $P(G_O)$ of partition $O(G)$
    Compute duplication event $D_C$ and the sets $S(D_C)$ and $T(D_C)$
    Compute the set of vertices $G_C$ and the partition $Q(G_C)$
    Build the reconstruction graph $R(N, G_C) = (\{1, \ldots, r\}, X \cup Y)$
    Partition vertices of $R(N, G_C)$ into connected components $R_1, \ldots, R_s$
    for $i = 1, \ldots, r$ do
        find the largest duplication event $D_{R_i}$ and the sets $S(D_{R_i})$ and $T(D_{R_i})$
    end for
    Compute $S(D_R) = \bigcup_i S(D_{R_i})$
    Compute $D'_{max}$ by selecting $S(D'_{max}) = S(D_C) \cup S(D_R)$ and $T(D'_{max}) = T(D_C) \cup T(D_R)$
    Construct network $N'$ by removing from $N$ all the vertices in $T(D'_{max})$ and renumbering remaining ones with $1, ..., n'$; assume renumbering transforms $S(D'_{max})$ to $S$
    Construct $D_{max}$ by selecting $S(D_{max}) = S$
    return $N'$ and $D_{max}$
end procedure
```

3.2 Network reconstruction using partial duplication model

Partial duplication model involves random deletions of gene interactions and the prospects of unambiguous and/or computationally efficient reconstruction of full evolution history of such networks seems unlikely. Also there is no obvious way to obtain the exact solution of LDEP without checking all the possible edge combinations as candidates for random deletions that have occurred during the duplication process. Nevertheless PDM still has some deterministic features from treating the incoming and the outgoing gene regulation edges differently (the latter being preserved by duplications) and these features can be exploited by heuristic approaches for finding approximate solutions to LDEP.

A straightforward adaptation of *FindLargestDuplicationEvent* to PDM involves checking all the candidates of connected components with k vertices that might be involved in duplications and all modifications of these components obtained by removing incoming edges. Unfortunately, such approach is feasible only for $k \leqslant 2$ (were $k = 2$ is the non-trivial case), allowing analysis of networks with up to 100 vertices. And since the possibility of a k vertex component to be involved in a duplication is dependent on which incoming edges have or have not been removed, we can use only a greedy approach that selects any suitable pair of components with $k \leqslant 2$ vertices whenever one is found. All duplicated components involving single vertex (the case $k = 1$) are independent however, and according to simulation experiments (see Section 4) around 85% of genes in randomly evolved networks using PDM are duplicated as singletons. Thus, a modified version of the algorithm is able to detect at least around 85% of genes from the largest possible duplication event.

3.3 Reconstruction of large or noisy networks

There are two challenges, if we wish to apply the reconstruction procedure to real networks. Firstly, despite the fact that graph automorphism problem is considered to be comparatively easy, it is doubtful whether we will be able to use it for graphs with several thousands of vertices. A potential solution here could be to use some heuristic for computation of automorphism orbits. With this, of course, we are loosing the possibility to reconstruct the complete evolution, however, we still might be able to recover large part of the most recent duplications.

Secondly, even if assuming that our duplication models well correspond to biological reality, the evolution of real networks involves other processes that govern the appearance of new or disappearance of existing regulatory relations. Thus, the real data are certain to be extremely 'noisy'. One consequence is that instead of checking for automorphisms we should be looking for 'approximate automorphisms', which is a much harder problem. The only practical solution here again could be use of a heuristic approach, heuristic computations of approximate solutions often computationally being not much harder than heuristic computations that attempt to find exact solutions.

Currently we have implemented a simple heuristic of such a type that is based on breadth-first searches from each network vertex followed by comparison of vertex and edge properties at different depth levels of search. Experiments with networks having few hundreds of vertices show that usually there are just few automorphism orbits that are either split or merged when this heuristic is used, however, such split or merged orbits almost always exist. Nevertheless for random networks the difference is not very large and the use of such type of heuristic can be regarded as a direct computation of partition $P(G_O)$ instead of initial G_O. The exact neighbourhood comparison rules used for vertex partitioning can also be easily modified to deal with different notions of 'approximate automorphisms' and making the approach adaptable for finding approximate solutions of LDEP for both FDM and PDM models.

4 Simulation and reconstruction experiments

Properties of networks. To study how similar are the properties of networks evolved under FDM and PDM models to the properties of real biological networks we have performed a number of tests for both duplication models as well as for FDM with 'noise' (a random deletion of some edges after each duplication step). In each test for a chosen duplication model and for some probability p (ranging between 0.001 and 0.2) 100 networks have been generated. This was done by starting from small random networks (15 to 30 genes) and by performing random duplication steps (a vertex was chosen for duplication with probability p) until networks' sizes reached around 5000 genes (FDM) or 20000 (PDM). The following conclusions were obtained from these tests:

- Probability p (within the used range) with which a vertex is selected for duplication in each step has little qualitative impact on the results.
- PDM produces networks corresponding very well to the power law. The ratio of number of edges to number of vertices is also similar to what we usually have in biological networks.

- FDM produces networks that very badly correspond to the power law. The number of edges is also much larger than in biological networks.
- FDM with 'noise' produces networks with vertex degree distribution that is close, but still deviates from the power law. Since it is actually not that clear how close to the power law should be the properties of real biological networks, the obtained networks might still be considered as biologically believable.
- For all models clustering coefficients are slightly increasing with network size (similarly as for hierarchical networks in Ravasz *et al* (2002)).
- For PDM around 85% of duplications involved just a duplication of a single gene.

The vertex degree distributions obtained for both FDM and PDM models as well as for 'noisy' FDM are shown in Figure 4.

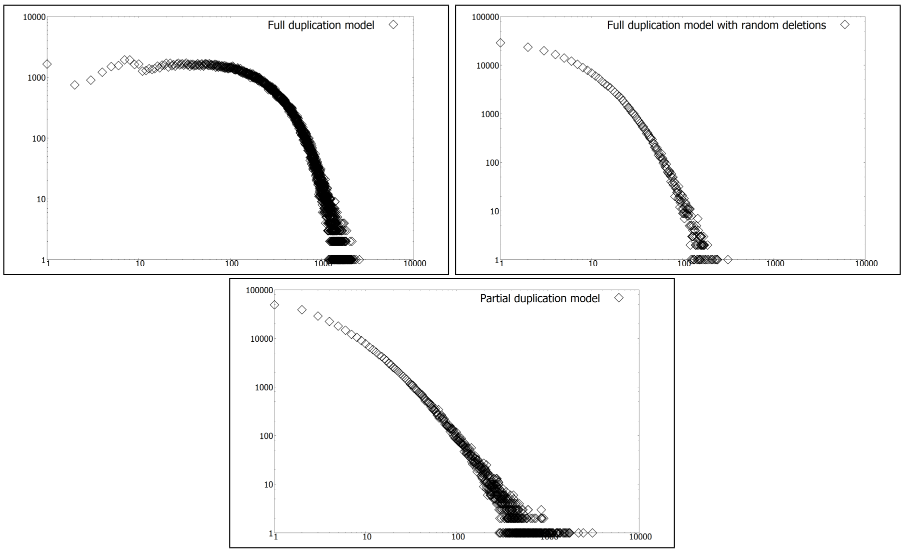

Fig. 4. Vertex degree distribution for duplication models (y axis shows vertex degree and x axis the number of vertices having this degree). To satisfy the power law, distribution graphs should be linear when drawn in logarithmic scale. A probability $p = 0.05$ for duplication of each vertex within duplication step was used. In model with random deletions the probability for edge deletion was 0.2. The exact values of these probabilities have a limited impact on degree distributions.

Reconstruction experiments on simulated networks. Experiments for reconstruction of evolution graph (solving of EGP) have been performed for FDM using *FindLargestDuplicationEvent* algorithm. The algorithm has been implemented by adapting a well-known program package **nauty** (McKay (2013)) for partitioning vertices in orbits. This partitioning turned out to be computationally the most expensive part of the algorithm and limited network sizes to around 200 vertices.

We generated 200 networks, starting from random networks and applying random duplication events until networks' sizes reached 150-200 genes. Then the procedure

FindLargestDuplicationEvent was repeatedly called until no further reconstruction was possible. In all 200 cases the reconstructed networks were identical to initial networks. The reconstruction process usually took few minutes on a standard workstation.

A modification of *FindLargestDuplicationEvent* for PDM limited to identification only of duplicated components consisting of a single vertex (the case $k = 1$) has been applied to PDM networks with up to 300 genes for finding the largest duplication event (PDM networks have fewer edges, thus the automorphism procedure can deal with graphs with more vertices, despite the fact that it has to be called many times). Experiments confirmed that in this way it is still possible to reconstruct a large part of the occurred duplications – as expected the number of vertices involved in reconstructed events were around 85% of the number of vertices that were involved in simulated duplications, although it was not possible to unambiguously identify all the vertices that were initially duplicated.

A heuristic version of algorithm has also been implemented and tested for FDM and PDM models. A simple breadth-first search based heuristic was used. The results indicate that for FDM networks with 150-200 vertices the largest duplication event can be reconstructed correctly by heuristic version in around $2/3$ of cases and on average around 90% of vertices from the largest duplication event are identified correctly. The program was also able to process in real time (few minutes) FDM networks with up to 5000 vertices and PDM networks with up to 20000 vertices. It is difficult to estimate the quality of these results due to the difficulty of computing the real largest duplication event for such networks.

Reconstruction of biological networks. It is a very tempting but a difficult challenge to try applying the reconstruction methods to real gene regulatory networks. One of the difficulties here is already outlined in the previous section, namely, the method is not particularly well suited for large and/or noisy networks. However, probably almost as serious difficulty is still the current limited availability of genome wide biologically confirmed gene regulation networks. Nevertheless, we thought it will be useful to apply the method to one of the most complete genome-wide networks currently available - *S.cerevisiae* (yeast) network (Lee *et al* (2002)). This network is particularly suited for such type of experiment, since it is widely believed that during the evolution yeast genome has undergone a full duplication.

The network is obtained using so called *ChIP-chip* experiments (the experimental technology was developed by Ren *et al* (2002)) and contains data about 6270 yeast genes, 106 of them being transcription factors. Gene regulations were inferred from affinities with which given transcription factors were bound to promoter regions of given genes. The network is given as 106 × 6270 matrix, containing probabilities (or, more exactly, p-values, where lower p-values correspond to higher probabilities) for given transcription factors to be involved in regulation of given genes. Using this matrix we constructed a deterministic network, connecting transcription factors with genes if the corresponding p-values were below a certain threshold. The resulting network contained 5962 genes (we discarded those for which protein sequences were unavailable). The total number of connections depended on the used threshold for p-values, however, small changes were observed using p-value thresholds from a wide range just below 0.0001, thus the corresponding network using this threshold and containing 15303 edges was selected for further experiments.

A heuristic version of algorithm for FDM model with breadth-first search based computation of the partition $P(G_O)$ was used. In total 277 non-singleton orbits (i.e. duplication candidates) were discovered. To evaluate whether the orbits really contain duplicated gene pairs, we made all pairwise comparisons between the protein sequences corresponding to 5962 genes (by using Smith-Waterman algorithm implemented in **ssearch** procedure from FASTA package (Pearson (1990))). A normalised comparison score computed as $ssearch_score(P_1, P_2)/min\{length(P_1), length(P_2)\}$then was assigned to each protein pair $\{P_1, P_2\}$.

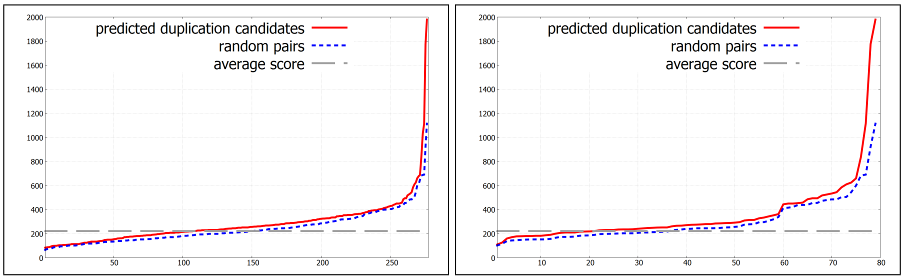

Fig. 5. Score distributions for detected duplication candidates, for randomly selected pairs, and the average scores. y axis shows scores, value on x axis is proportional to the number of pairs not exceeding given score. The first diagram shows all detected pairs, the second the remaining ones after the pairs of adjacent genes have been filtered out.

The distribution of normalized scores is shown in Figure 5. Here the scores for genes within orbits are compared with the average score and with scores for randomly selected gene pairs. At first the results may not seem too encouraging with similarity scores between the predicted duplicated gene pairs being only slightly above the scores between random pairs. Also the numbers of the predicted pairs with scores below and above the average score value are practically equal. Nevertheless the scores between the predicted duplicated gene pairs remain consistently higher over the whole range of scores, thus suggesting that some information about network evolution still can be recovered from network topology alone. Also the predicted duplicate genes include a number of pairs that are regarded as known duplicates by biologists – e.g. gene pairs COS5 and COS8, or YLR460C and YNL134L.

An additional feature observed from this experiment was that unusually high number of orbits contained genes that are adjacent (or, in few cases, almost adjacent) on the yeast genome. From 158 two gene orbits 120 contained such adjacent genes. In most cases the genes of these pairs were from the opposite strands of DNA, and it is known that such pairs tend to have common regulators. To filter out this effect we considered only the orbits not containing such adjacent genes. With such genes excluded the higher values of similarity scores between predicted duplicates compared to the scores between random pairs became more apparent. Also approximately $3/4$ of predicted duplicates had similarity scores above the average value.

5 Conclusions and discussion

The current stage of research on models of gene duplication and on possibilities of reconstruction of evolution of gene regulatory networks from their topology raises a number of interesting questions both, from the perspective of mathematical properties of the models and the design of efficient algorithms, and from the perspective of application of such type of evolution reconstruction methods to real biological data sets.

From the mathematical point of view, likely the most interesting are the properties of FDM networks. One of the open questions already discussed in this paper is the question about the existence of two non-isomorphic irreducible networks N_1 and N_2 and evolution graphs $E(S_1)$ and $E(S_2)$, such that the resulting evolved networks $D(N_1, E(S_1))$ and $D(N_2, E(S_2))$ are isomorphic.

Another interesting question is related to computation of reconstruction graphs $R(N, G_C) = (\{1, \ldots, r\}, X \cup Y)$. Currently these graphs contain two types of edges: edge of type Y between partitions Q_1 and Q_2 corresponding to two vertices of $R(N, G_C)$ implies that these partitions can not be included in a single duplication event; edge of type X, however, implies only that Q_1 and Q_2 *could* (but not necessarily *have to*) be included in a single duplication event. Thus, the role of type X edges is mainly in defining of connected components R_i of $R(N, G_C)$, each of these then can be analysed independently.

Still, often it is possible to show that the two partitions Q_1 and Q_2 *must* be included in a single duplication event, and this property can be decided by analysing solely the vertices belonging to Q_1 or Q_2 and the edges between them. An interesting question therefore is whether we can define a simple property (similar to the properties we use in Definition 5) of graph $N_{Q_1 \cup Q_2}$ that is necessary and sufficient for such an 'enforced' X edge to exist.

From the computational perspective, however, it is not too difficult to decide whether two partitions Q_1 and Q_2 must be involved in a single duplication event and should be connected by an 'enforced' X edge or not. Actually this is already done in our implementation of the algorithm as the initial stage of exhaustive search used for computing duplication events D_{R_i}. Assuming that we have computed such 'enforced' X edges, there are number of questions about the structure of connected components R_i themselves. For example, is this the case that all the vertices connected by 'enforced' X edges must belong to a clique? This really have been the case in all our simulation experiments, however, the size of components R_i encountered in these experiments have been too small to assess the validity of such hypothesis. However, if the hypothesis is true, it will imply that LDEP can be reduced to INDEPENDENT SET problem.

For the applications to analysis of real biological networks it is very likely, however, that only heuristic approaches will be practically useful and the improvements of such approaches are certainly possible (currently we have used only a simple breadth-first search based heuristic for characterizing the nearest neighbourhood of each of the vertices). Also in the best case we likely will be able to reliably identify only candidate gene pairs from some of the latest duplication events that have occurred, and not a consistent evolution history.

Nevertheless, the analysis results on yeast genome suggest that information about gene duplications that have occurred long time ago can still be recovered from the network topology alone. Biological gene regulatory networks in almost all cases contain additional information (about gene similarity, relative positions of genes on genome etc.) that usually is more reliable than the topological structure of the network. Thus, it seems there is a good potential for practical applications integrating the information about the gene regulatory network structure with the information about the genes themselves. One study of such type that we are considering is identification of biologically well-known small gene regulatory patterns in data sets that are obtained by high-throughput technologies at genome wide level. Currently such genome wide networks are described only by probability matrices (in a similar way as for yeast network, which we have analysed here) and do not possess well-defined topological features (the latter being dependent on the probability thresholds applied). Identification of biologically well-established regulatory patterns could supplement such high-throughput data sets with additional biologically useful information. Apart from the yeast genome data set analysed here, several other data sets suitable for such studies and obtained by newer and more reliable NGS technologies are recently becoming available, e.g. data about gene regulation in *E.coli* obtained from *RNA-seq* experiments (Gama-Castro *et al* (2011)), and these data sets could merit analysis using techniques similar to those we have presented here.

References

Babu, M.M., Luscombe, N.M., Aravind, L., Gerstein M., Teichmann, S.A. (2004). Structure and evolution of transcriptional regulatory networks, *Current Opinions in Structural Biology* 14, 283-291.

Barabasi A.L., Albert, R. (1999). Emergence of scaling in random networks, *Science* 286, 509-512.

Chung, F., Lu, L., Dewey, T.G., Galas, D.J. (2003). Duplication models for biological networks, *Journal of Computational Biology* 10, 677-687.

Erwin, D.H., Davidson, E.C. (2009). The evolution of hierarchical gene regulatory networks, *Nature Reviews Genetics* 10, 141–148.

Friedman, R., Hughes, A. (2004). Gene duplication and the structure of eucaryotic genomes, *Genome Research* 11, 373-381.

Gama-Castro, S. et al. (2011) RegulonDB version 7.0: transcriptional regulation of Escherichia coli K-12 integrated within genetic sensory response units, *N*ucleic Acid Research 39, 98–105.

Garfield, D.A., Wray, G.A. (2010). The evolution of gene regulatory interactions, *BioScience* 60, 15-23.

Gu, Z., Cavalcanti, A., Chen, F., Bouman, P., Li, W. (2002). Extent of gene duplication in the genomes of drosophila, nematode and yeast, *M*olecular Biology and Evolution 19, 256-262.

Lee, T.I. et al (2002). Transcriptional regulatory networks in *Saccharomyces cerevisiae*, *Science* 298, 799-804.

McKay, B.D., Piperno A. (2013) Practical graph isomorphism II, *Journal of Symbolic Computation* 60, 94-112.

Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., Alon, U. (2002). Network motifs: simple building blocks of complex networks, *Science* 298, 824-827.

Pearson, W.R. (1990). Rapid and sensitive sequence comparison with FASTP and FASTA, *Methods in Enzymology* 183, 63-98.

Ren, N. et al (2002). Genome-wide location and function of DNA binding proteins, *Science* 290, 2306-2309.

Ravasz, E., Somera, A.L., Mongru, D.A., Oltvai, Z.N., Barabasi, A.L. (2002). Hierarchical organization of modularity in metabolic networks, *Science* 297, 1551-1555.

Romero I.G., Ruvinsky, I., Gilad, Y. (2012). Comparative studies of gene expression and the evolution of gene regulation, *Nature Reviews Genetics* 13, 505–516.

Sidow, A. (1996). Gen(om)e duplications in the evolution of early vertebrates, *Current Opinion in Genetics and Development* 6, 715-722.

Strogatz, S.H. (2001). Exploring complex networks, *Nature* 410, 268-276.

Teichmann, S.A., Rison, S.C.G., Thornton, J.M., Riley, M., Gough, J., Chothia, C. (2001). The evolution and structural anatomy of the small molecule metabolic pathways in *Escheria coli*, *Journal of Molecular Biology* 311, 693-708.

Thompson, D. et al (2013). Evolutionary principles of modular gene regulation in yeasts, *eLife* doi: 10.7554/eLife.00603.001.

Thompson, D., Regev, A., Roy, S. (2015). Comparative analysis of gene regulatory networks: from network reconstruction to evolution, *Annual Review of Cell and Developmental Biology* 31, 399–428.

Wagner, A. (1994). Evolution of gene networks by gene duplications: A mathematical model and its implications on genome organization, *Proceedings of the National Academy of Sciences of USA* 99, 4387-4391.

Watts D.J., Strogatz, S.H. (1998) Collective dynamics of 'small world' networks, *Nature* 393, 440-442.

Wolf, Y.I., Karev, G., Koonin, E.V. (2002) Scale-free networks in biology: new insights into the fundamentals of evolution?, *BioEssays* 24, 105-109.

Received October 28, 2016 , accepted November 11, 2016