Information and Computation 213 (2012) 23–32

Contents lists available at SciVerse ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

Mod/Resc Parsimony Inference: Theory and application

Igor Nor [a,b,*], Danny Hermelin [c], Sylvain Charlat [a], Jan Engelstadter [d], Max Reuter [e], Olivier Duron [f], Marie-France Sagot [a,b,*]

[a] Université de Lyon, F-69000, Lyon, Université Lyon 1, CNRS, UMR5558, France
[b] Bamboo Team, INRIA Grenoble Rhône-Alpes, France
[c] Max Planck Institute for Informatics, Saarbrücken, Germany
[d] Institute of Integrative Biology, ETH Zurich, Switzerland
[e] University College London, UK
[f] Institute of Evolutionary Sciences, CNRS – University of Montpellier II, France

ARTICLE INFO

Article history:
Available online 2 February 2012

Keywords:
Computational biology
Biclique edge covering
Bipartite graph
Boolean matrix
NP-completeness
Graph theory
Fixed-parameter tractability
Kernelisation

ABSTRACT

We address in this paper a new computational biology problem that aims at understanding a mechanism that could potentially be used to genetically manipulate natural insect populations infected by inherited, intra-cellular parasitic bacteria. In this problem, that we denote by Mod/Resc Parsimony Inference, we are given a boolean matrix and the goal is to find two other boolean matrices with a minimum number of columns such that an appropriately defined operation on these matrices gives back the input. We show that this is formally equivalent to the Biclique Edge Cover for Bipartite Graphs problem and derive some complexity results for our problem using this equivalence. We provide a new, fixed-parameter tractability approach for solving both problems that slightly improves upon a previously published algorithm for the Biclique Edge Cover for Bipartite Graphs. Finally, we present experimental results applying some of our techniques to a real-life dataset.

1. Introduction

Wolbachia is a genus of inherited, intra-cellular bacteria that infect many arthropod species, including a significant proportion of insects. The bacterium was first identified in 1924 by M. Hertig and S.B. Wolbach in *Culex pipiens*, a species of mosquito. *Wolbachia* spreads by altering the reproductive capabilities of its hosts [6]. One of these alterations consists in inducing so-called *cytoplasmic incompatibility* [7]. This phenomenon, in its simplest expression, results in the death of embryos produced in crosses between males carrying the infection and uninfected females. A more complex pattern is the death of embryos seen in crosses between males and females carrying different *Wolbachia* strains. The study of *Wolbachia* and cytoplasmic incompatibility is of interest due to the high incidence of such infections, amongst others in human disease vectors such as mosquitoes, where cytoplasmic incompatibility could potentially be used as a driver mechanism for the genetic manipulation of natural populations.

The molecular mechanisms underlying cytoplasmic incompatibility are currently unknown, but the observations are consistent with a "toxin/antitoxin" model [18]. According to this model, the bacteria present in males modify the sperm (the so-called modification, or mod factor) by depositing a "toxin" during its maturation. Bacteria present in females, on the other hand, deposit an antitoxin (rescue, or resc factor) in the eggs, so that offsprings of infected females can develop normally. The simple compatibility patterns seen in several insect hosts species [1–3] have led to the general view that cytoplasmic

* Corresponding author at: Université de Lyon, F-69000, Lyon, Université Lyon 1, CNRS, UMR5558, France.
E-mail addresses: norigor@gmail.com (I. Nor), Marie-France.Sagot@inria.fr (M.-F. Sagot).

0890-5401/$ – see front matter
doi:10.1016/j.ic.2011.03.008

C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	0	0	1	0	0	0	1	1	0	0	0	0	0	1	1	0	1	1	1
2	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
3	0	1	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0
4	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
7	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
8	1	1	1	1	1	0	1	0	0	1	0	1	1	1	1	1	1	1	0
9	0	0	1	0	0	0	1	1	0	0	0	0	0	1	1	1	1	1	0
10	1	0	1	0	0	0	1	0	1	0	1	0	0	1	0	0	1	1	0
11	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
12	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
16	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 1. The *Culex pipiens* dataset. Rows represent males and columns females.

incompatibility relies on a single pair of mod/resc genes. However, more complex patterns, such as those seen in Fig. 1 of the mosquito *Culex pipiens* [5], suggest that this conclusion cannot be generalised. Indeed, we show (see Section 5) that eight mod/resc gene pairs are necessary to explain this data.

The aim of this paper is to provide a first model and algorithm to determine the minimum number of mod and resc genes required to explain a compatibility dataset for a given insect host. Such an algorithm will have an important impact on the understanding of the genetic architecture of cytoplasmic incompatibility. Beyond *Wolbachia*, the method proposed here can be applied to any parasitic bacterium inducing cytoplasmic incompatibility.

Let us now propose a formal description of this problem. Let the *compatibility matrix* C be an n-by-n matrix describing the observed cytoplasmic compatibility relationships among n strains, with males in rows and females in columns. For the *Culex pipiens* dataset, the content of the C matrix is directly given by Fig. 1. For each entry $C_{i,j}$ of this matrix, a value of 1 indicates that the cross between the i'th male and j'th female is incompatible, while a value of 0 indicates that it is compatible. No intermediate levels of incompatibility are observed in *Culex pipiens*, so that such a discrete code (0 or 1) is sufficient to describe the data. Let the *mod matrix* M be an n-by-k matrix, with n strains and k mod genes. For each $M_{i,j}$ entry, a 0 indicates that strain i does not carry gene j, and a 1 indicates that it does carry this gene. Similarly, the *rescue matrix* R is an n-by-k matrix, with n strains and k resc genes, where each $R_{i,j}$ entry indicates whether strain i carries gene j. A cross between male i and female j is compatible only if strain j carries at least all the rescue genes matching the mod genes present in strain i. Using this rule, one can assess whether an (M, R) pair is a solution to the C matrix, that is, to the observed data.

We can easily find non-parsimonious solutions to this problem, that is, large M and R matrices that are solutions to C, as will be proven in the next section. However, solutions may also exist with fewer mod and resc genes. The problem can be summarised as follows: let C (compatibility) be a boolean n-by-n matrix. A pair of n-by-k boolean matrices M (mod) and R (resc) is called a solution to C if, for any row j in R and row i in M, $C_{i,j} = 0$ if and only if $R_{j,\ell} \geqslant M_{i,\ell}$ holds for all ℓ, $1 \leqslant \ell \leqslant k$. This appropriately models the fact stated above that, for any cross to be compatible, the female must carry at least all the rescue genes matching the mod genes present in the male. For a given matrix C, we are interested in the minimum value of k for which solutions to C exist, as well as in the set of all solutions for this minimum k. We refer to this problem as the Mod/Resc Parsimony Inference problem (see also Section 2). Since in some cases, data (on females or males) may be missing, the compatibility matrix C has dimension n-by-m for n not necessarily equal to m. We will consider this more general situation in what follows.

In this paper, we present the Mod/Resc Parsimony Inference problem and prove that it is equivalent to a well-studied graph-theoretical problem known in the literature by the name of Biclique Edge Cover for Bipartite Graphs problem, henceforth called the Biclique Edge Cover problem for simplicity. In this problem, we are given a bipartite graph, and we want to cover its edges with a minimum number of complete bipartite subgraphs (bicliques). This problem is known to be NP-complete, and thus Mod/Resc Parsimony Inference turns out to be NP-complete as well. In Section 4, we investigate a previous fixed-parameter tractability approach [8] for solving the Biclique Edge Cover problem and improve its algorithm. In addition, we show a reduction between this problem and the Clique Edge Cover problem. Finally, in Section 5, we

present experimental results where we applied some of these techniques to the *Culex pipiens* data set presented in Fig. 1. This provided a finding that is surprising from a biological point of view.

2. Problem definition and notation

In this section, we briefly review some notation and terminology that will be used throughout the paper. We also give a precise mathematical definition of the Mod/Resc Parsimony Inference problem we study. For this, we first need to define a basic operation between two boolean vectors:

Definition 1. The $\otimes$ vectors multiplication is an operation between two boolean vectors $U, V \in \{0,1\}^k$ such that:

$$U \otimes V := \begin{cases} 1 & U[i] > V[i] \text{ for some } i \in \{1,\ldots,k\} \\ 0 & \text{otherwise} \end{cases}$$

In other words, the result of the $\otimes$ multiplication is 0 if, for all corresponding locations, the value in the second vector is not less than in the first.

The reader should note that this operation is not commutative. For example, if $U := (0,1,1,0)$ and $V := (1,1,1,0)$, then $U \otimes V = 0$, while $V \otimes U = 1$. We next generalise the $\otimes$ multiplication to boolean matrices. This follows easily from the observation that the boolean vectors $U, V \in \{0,1\}^k$ may be seen as matrices of dimension 1-by-k. We thus use the same symbol $\otimes$ to denote the operation applied to matrices.

Definition 2. The $\otimes$ row-by-row matrix multiplication is a function $\{0,1\}^{n\times k} \times \{0,1\}^{m\times k} \to \{0,1\}^{n\times m}$ such that $C = M \otimes R$ iff $C_{i,j} = M_i \otimes R_j$ for all $i \in \{1,\ldots,n\}$ and $j \in \{1,\ldots,m\}$. (Here M_i and R_j respectively denote the i'th and j'th rows of M and R.)

Definition 3. In the Mod/Resc Parsimony Inference problem, the input is a boolean matrix $C \in \{0,1\}^{n\times m}$, and the goal is to find two boolean matrices $M \in \{0,1\}^{n\times k}$ and $R \in \{0,1\}^{m\times k}$ such that $C = M \otimes R$ and with k minimum.

We first need to prove there is always a correct solution to the Mod/Resc Inference Problem. Here we show that there is always a solution for as many mod and resc genes as the minimum between the number of male and female strains in the dataset.

Lemma 1. *The* Mod/Resc Parsimony Inference *problem always has a solution.*

Proof. A satisfying output for the Mod/Resc Parsimony Inference problem always exists for any possible C of size n-by-m. For instance, let M be of size n-by-n and equal to the identity matrix, and let R be of size m-by-n and such that $R = \overline{C}^T$. This solution is correct since the only 1-value in an arbitrary row r_i of the matrix M is at location M_{ii}. Thus, the only situation where $C_{ij} = 1$ is when $R_{ji} = 0$, which is the case by construction. □

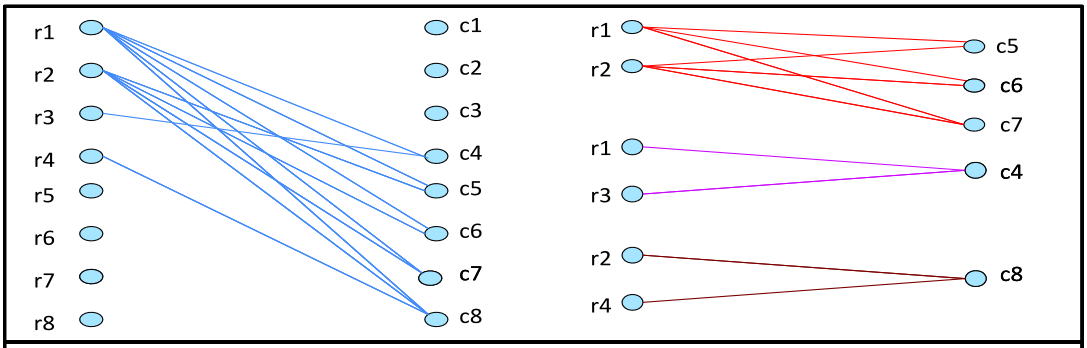
We now adopt some standard graph-theoretic terminology and notation. We thus use G, G', and so forth to denote graphs in general, where $V(G)$ denotes the vertex set of a graph G, and $E(G)$ its edge-set. By a *subgraph* of G, we mean a graph G' with $V(G') \subseteq V(G)$ and $E(G') \subseteq E(G)$. For a bipartite graph G, *i.e.* a graph whose vertex-set can be partitioned into two classes with no edges occurring between vertices of the same class, we use $V_1(G)$ and $V_2(G)$ to denote the two vertex classes of G. A *complete bipartite graph* (*biclique*) is a bipartite graph G with $E(G) := \{\{u,v\} \mid u \in V_1(G),\ v \in V_2(G)\}$. We sometimes use B, B_1, and so forth to denote bicliques.

3. Equivalence to biclique edge cover

In this section, we show that the Mod/Resc Parsimony Inference problem is equivalent to a classical and well-studied graph theoretical problem known in the literature as the Biclique Edge Cover problem. Using this equivalence, we first derive the complexity status of Mod/Resc Parsimony Inference problem, and later devise FPT algorithms for it. We begin with a formal definition of the Biclique Edge Cover problem.

Definition 4. In the Biclique Edge Cover problem, the input is a bipartite graph G, and the goal is to find the minimum number of bicliques $B_1,\ldots,B_k$ of G such that $E(G) := \bigcup_\ell E(B_\ell)$.

Given a bipartite graph G with $V_1(G) := \{u_1,\ldots,u_n\}$ and $V_2(G) := \{u_1,\ldots,u_m\}$, the *bi-adjacency* matrix of G is a boolean matrix $A(G) \in \{0,1\}^{n\times m}$ defined by $A(G)_{i,j} := 1 \iff \{u_i, v_j\} \in E(G)$. In this way, every boolean matrix C corresponds to a bipartite graph, and vice versa.



G	1	2	3	4	5	6	7	8
1	0	0	0	1	1	1	1	0
2	0	0	0	0	1	1	1	1
3	0	0	0	1	0	0	0	0
4	0	0	0	0	0	0	0	1
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0

M	1	2	3
1	1	1	0
2	1	0	1
3	0	1	0
4	0	0	1
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0

R	1	2	3
1	1	1	1
2	1	1	1
3	1	1	1
4	1	0	1
5	0	1	1
6	0	1	1
7	0	1	1
8	1	1	0

Fig. 2. Reduction illustrated.

Theorem 1. *Let C be a boolean matrix of size $n \times m$. Then there are two matrices $M \in \{0,1\}^{n\times k}$ and $R \in \{0,1\}^{m\times k}$ with $C = M \otimes R$ iff the bipartite graph G with $A(G) := C$ has a biclique edge cover with k bicliques.*

Proof. ($\Longleftarrow$) Let G be the bipartite graph with the bi-adjacency matrix C, and suppose G has biclique edge cover $B_1, B_2, \ldots, B_k$. We construct two boolean matrices M and R as follows. Let $V_1(G) := \{u_1, \ldots, u_n\}$ and $V_2(G) := \{v_1, \ldots, v_m\}$. We define:

1. $M_{i,\ell} = 1 \iff u_i \in V_1(B_\ell)$.
2. $R_{j,\ell} = 0 \iff v_j \in V_2(B_\ell)$.

An illustration of this construction is given in Fig. 2.

We argue that $C = M \otimes R$. Consider an arbitrary location $C_{i,j} = 1$. By definition we have $\{u_i, v_j\} \in E(G)$. Since the bicliques $B_1, \ldots, B_k$ cover all edges of G, we know that there is some ℓ, $\ell \in \{1, \ldots, k\}$, with $u_i \in V_1(B_\ell)$ and $v_j \in V_2(B_\ell)$. By construction, we know that $M_{i,\ell} = 1$ and $R_{j,\ell} = 0$, and so $M_i \otimes R_j = 1$, which means that the entry at row i and column j in $M \otimes C$ is equal to 1. On the other hand, if $C_{ij} = 0$, then $\{u_i, v_j\} \notin E(G)$, and thus there is no biclique B_ℓ with $u_i \in V_1(B_\ell)$ and $v_j \in V_2(B_\ell)$. As a result, for all $\ell \in \{1, \ldots, k\}$, if $M_{i,\ell} = 1$ then $R_{i,\ell} = 1$ as well, which means that the result of the $\otimes$ multiplication between the i'th row in M and the j'th row in R will be equal to 0.

($\Longrightarrow$) Assume there are two matrices $M \in \{0,1\}^{n\times k}$ and $R \in \{0,1\}^{m\times k}$ with $C = M \otimes R$. Construct k subgraphs $B_1, \ldots, B_k$ of G, where the ℓ'th subgraph is defined as follows:

1. $u_i \in V_1(B_\ell) \iff M_{i,\ell} = 1$.
2. $v_j \in V_2(B_\ell) \iff R_{j,\ell} = 0$.
3. $\{u_i, v_j\} \in E(B_\ell) \iff \{u_i, v_j\} \in E(G)$.

We first argue that each of the subgraphs $B_1, \ldots, B_k$ is a biclique. Consider an arbitrary subgraph B_ℓ, and an arbitrary pair of vertices $u_i \in V_1(B_\ell)$ and $v_j \in V_2(B_\ell)$. By construction, it follows that $M_{i,\ell} = 1$ and $R_{i,\ell} = 0$. As a result, it must be that $C_{i,j} = 1$, which means that $\{u_i, v_j\} \in E(G)$. Next, we argue that $\bigcup_\ell E(B_\ell) = E(G)$. Consider an arbitrary edge $\{u_i, v_j\} \in E(G)$. Since $C = A(G)$, we have $C_{i,j} = 1$. Furthermore, since $M \otimes R = C$, there must be some $\ell \in \{1, \ldots, k\}$ with $M_{i,\ell} > R_{j,\ell}$. However, this is exactly the condition for having u_i and v_j in the biclique subgraph B_ℓ. It follows that indeed $\bigcup_\ell E(B_\ell) = E(G)$, and thus the theorem is proved. □

Due to the equivalence between Mod/Resc Parsimony Inference and Biclique Edge Cover, we can infer from known complexity results regarding Biclique Edge Cover the complexity of our problem. First, since Biclique Edge Cover is well known to be NP-complete [17], it follows that Mod/Resc Parsimony Inference is NP-complete as well. Furthermore, Gruber

and Holzer [12] recently showed that the Biclique Edge Cover problem cannot be approximated within a factor of $n^{1/3-\varepsilon}$ unless P = NP where n is the total number of vertices. Since the reduction given in Theorem 1 is linear, it is clearly an approximate preserving reduction. We can thus deduce the following:

Theorem 2. Mod/Resc Parsimony Inference *is* NP-complete*, and furthermore, for all* $\varepsilon > 0$*, the problem cannot be approximated within a factor of* $(n+m)^{1/3-\varepsilon}$ *unless* P = NP*.*

4. Fixed-parameter tractability

In this section, we explore a parameterised complexity approach [4,9,16] for the Mod/Resc Parsimony Inference problem. Due to the equivalence shown in the previous section, we focus for convenience reasons on Bipartite Graph Biclique Edge Cover. In parameterised complexity, problem instances are appended with an additional parameter, usually denoted by k, and the goal is to find an algorithm for the given problem which runs in time $f(k) \cdot n^{O(1)}$, where f is an arbitrary computable function. In our context, our goal is to determine whether a given input bipartite graph G with n vertices has a biclique edge cover of size k in time $f(k) \cdot n^{O(1)}$.

4.1. The kernelisation

Fleischner et al. [8] studied the Biclique Edge Cover problem in the context of parameterised complexity. The main result in their paper is to provide a kernel for the problem based on the techniques given by Gramm et al. [10] for the similar Clique Edge Cover problem. Kernelisation is a central technique in parameterised complexity which is best described as a polynomial-time transformation that converts instances of arbitrary size to instances of a size bounded by the problem parameter (usually of the same problem), while mapping "yes"-instances to "yes"-instances, and "no"-instances to "no"-instances. More precisely, a *kernelisation algorithm* $\mathcal{A}$ for a parameterised problem (language) Π is a polynomial-time algorithm such that there exists some computable function f that, given an instance (I,k) of Π, $\mathcal{A}$ produces an instance (I',k') of Π with:

- $|I'| + k' \leqslant f(k)$, and
- $(I,k) \in \Pi \iff (I',k') \in \Pi$.

We refer the reader to *e.g.* [13,16] for more information on kernelisation.

A typical kernelisation algorithm works with reduction rules, which transform a given instance to a slightly smaller equivalent instance in polynomial time. The typical argument used when working with reduction rules is that once none of these can be applied, the resultant instance has size bounded by a function of the parameter. For the Biclique Edge Cover, two kernelisation rules have been applied by Fleischner et al. [8]:

RULE 1: If G has a vertex with no neighbours, remove this vertex without changing the parameter.
RULE 2: If G has two vertices with identical neighbours, remove one of these vertices without changing the parameter.

Lemma 2. *(See [8].) Applying Rules* 1 *and* 2 *above exhaustively gives a kernelisation algorithm for* Biclique Edge Cover *problem that runs in* $O(\max(n,m)^3)$ *time, and transforms an instance* (G,k) *to an equivalent instance* (G',k) *with* $|V(G')| \leqslant 2^k$ *and* $|E(G')| \leqslant 2^{2k}$*.*

We add two additional rules, which will be necessary for further interesting properties.

RULE 3: If there is a vertex v with exactly one neighbour u in G, then remove both v and u, and decrease the parameter by one.

Lemma 3. *Rule* 3 *is correct.*

Proof. Assume a biclique cover of size k of the graph, and assume that vertex v is a member of some of the bicliques in this cover. By definition, at least one of the bicliques covers the edge $\{u,v\}$. Since this is the only edge adjacent to v, the bicliques that cover $\{u,v\}$ include only vertex u among the vertices in its bipartite vertex class. Thus, a biclique that covers $\{u,v\}$ can be extended to cover all other edges of u while keeping the property of being a biclique. □

RULE 4: Assume Rule 3 does not apply. If there is a vertex v in G which is adjacent to all vertices in the opposite bipartition class of G, then remove v without decreasing the parameter.

Lemma 4. *Rule* 4 *is correct.*

Proof. After applying Rule 3 above, each remaining vertex in the graph has at least two neighbours. Assume a biclique cover of size k of all the edges except those adjacent to vertex v. Assume w.l.o.g. that $v \in V_1(G)$. Since each vertex $u \in V_2(G)$ has degree at least 2, it is adjacent to an edge which is covered by the biclique cover. It therefore belongs to some biclique in this cover. For each biclique in the cover, add now vertex v to its set of vertices. Since v is adjacent to all the vertices of $V_2(G)$, each changed component is a correct biclique and the new solution covers all the edges, including those of vertex v, and is of same size. □

Let us now consider the time complexity for checking the new rules introduced. Let us assume we have a counter for each vertex, which has the size of its set of neighbours. Once a vertex has been found to which the rule should be applied, applying each rule takes $O(\max(n, m))$ time, including updating the counters of the neighbours of the deleted vertex. Linearly running through the vertices and checking each rule condition also requires $O(\max(n, m))$ time using the counters. Since one can apply the reduction rules at most $O(\max(n, m))$ times, the total time required for the extended kernelisation remains $O(\max(n, m)^3)$. We observe that although the new rules do not change the kernelisation size, which remains 2^k vertices in a solution of size k, they can be useful in the following section.

4.2. Biclique Edge Cover *and* Clique Edge Cover

In this section, we show the connection between the Biclique Edge Cover and the Clique Edge Cover problems. We show that in the context of fixed-parameter tractability, we can easily translate our problem to the classical clique covering problem and then use it for a solution to our problem. For instance, it gives another way for the kernelisation of the problem and can provide interesting heuristics, mentioned in [10].

Given a kernelised bipartite graph G' as an instance to the Biclique Edge Cover problem, we transform G' into a (non-bipartite) graph G'' defined by $V(G'') := V(G') \cup \{v'\} \cup \{u'\}$ and $E(G'') := E(G') \cup \{\{u, v\}\colon\ u, v \in V_1(G') \cup \{v'\}$ and $u, v \in V_2(G') \cup \{u'\}\}$ where v' and u' are two new nodes not in $V(G')$.

Following this construction, for a clique $K = (V_1(K) \cup V_2(K), E(K))$ in G'', where $V_1(K) \subseteq V_1(G'), V_2(K) \subseteq V_2(G')$ and $E(K) = V_1(K) \otimes V_2(K) \cup \{\{u, v\}\colon\ u, v \in V_1(K)$ and $u, v \in V_2(K)\}$, we define its corresponding biclique $B = (V_1(K) \cup V_2(K), V_1(K) \otimes V_2(K)) \subseteq G'$.

Theorem 3. *The edges of G' can be covered with k bicliques iff the edges of G'' can be covered with $k + 2$ cliques.*

Proof. Suppose $B_1, \ldots, B_k$ is a biclique edge cover of G'. Then each $V(B_i)$, $i \in \{1, \ldots, k\}$, induces a clique in G''. Furthermore, the only remaining edges which are not covered in G'' are the ones between vertices in $V_1(G') \cup \{v'\}$ and vertices in $V_2(G') \cup \{u'\}$, which can be covered by the two cliques induced by these vertex sets in G''. Altogether this gives us $k + 2$ cliques that cover all edges in G''. Conversely, take a clique edge cover $K_1, \ldots, K_c$ of G''. By construction, v' cannot share the same clique with any node in $V_2(G') \cup \{u'\}$ and likewise u' cannot share the same clique with any node in $V_1(G') \cup \{v'\}$. It follows that there must be at least two cliques in $\{K_1, \ldots, K_c\}$, say K_1 and K_2, with $V(K_1) \subseteq V_1(G') \cup \{v'\}$ and $V(K_2) \subseteq V_2(G') \cup \{u'\}$. Thus, there is a subset of the cliques in $\{K_3, \ldots, K_c\}$ which have vertices in both partition classes of G', and which cover all the edges in G'. Taking the corresponding bicliques in G', and adding duplicated bicliques if necessary, gives us k bicliques that cover all edges in G'. □

4.3. Algorithms

After the kernelisation algorithm is applied, the next step is usually to solve the problem using brute-force. This is what is done in [8]. However, the time complexity given there is inaccurate, and the parametric-dependent time bound of their algorithm is $O(k^{4^k} 2^{3k}) = O(2^{2^{2k} \lg k + 3k})$ instead of the $O(2^{2k^2+3k})$ bound stated in their paper. Furthermore, the algorithm they describe is initially given for the related Biclique Edge Partition problem (where each edge is allowed to appear exactly once in a biclique), and the adaptation of such algorithm to the Biclique Edge Cover problem is left vague and imprecise. Here, we suggest two possible brute-force procedures for the Biclique Edge Cover problem, each of which outperforms the algorithm of [8] in the worst-case. We assume throughout that we are working with a kernelised instance obtained by applying the algorithm described in Section 4.1, *i.e.* a pair (G', k) where G' is a bipartite graph with at most 2^k vertices (and consequently at most 4^k edges).

The first brute-force algorithm: For each $k' \leqslant k$, try all possible partitions of the edge-set $E(G')$ of G' into k' subsets. For each such partition $\Pi = \{E_1, \ldots, E_{k'}\}$, check whether each of the subgraphs $G'[E_1], \ldots, G'[E_{k'}]$ is a biclique, where $G'[E_i]$ is the subgraph of G induced by E_i. If yes, report $G'[E_1], \ldots, G'[E_{k'}]$ as a solution. If some $G'[E_i]$ is not a biclique, check whether edges in $E(G') \setminus E(G'_i)$ can be added to $E[G'_i]$ in order to make the graph a biclique. Continue with the next partition if some graph in $G'[E_1], \ldots, G'[E_{k'}]$ cannot be appended in this way in order to get a biclique, and otherwise report the solution found. Finally, if the above procedure fails for all partitions of $E(G')$ into $k' \leqslant k$ subsets, report that G' does not have a biclique edge cover of size k.

Lemma 5. *The above algorithm correctly determines whether G' has a biclique edge cover of size k in time $\frac{2^{2^{2k}\lg k+2k+\lg k}}{k!}$.*

Proof. Correctness of the above algorithm is immediate in case a solution is found. To see that the algorithm is also correct when it reports that no solution can be found, observe that for any biclique edge cover $B_1, \ldots, B_k$ of G, the set $\{E_1, \ldots, E_k\}$ with $E_i := E(G'_i) \setminus \bigcup_{j<i} E(G'_j)$ defines a partition of $E(G')$ (with some of the E_i's possibly empty), and given this partition, the algorithm above would find the biclique edge cover of G'. Correctness of the algorithm thus follows.

Regarding the time complexity, the time needed for appending edges to each subgraph is at most $O(|(V(G'))^2|) = O(2^{2k})$, and thus a total of $O(2^{2k}k) = O(2^{2k+\lg k})$ time is required for the entire partition. The number of possible partitions of $E(G')$ into k disjoint set is the *Stirling number of the second kind* $S(2^{2k}, k)$, which has been shown in [15] to be asymptotically equal to $O(\frac{k^{4^k}}{k!} = \frac{2^{2^{2k}\lg k}}{k!})$. Thus, the total complexity of the algorithm is $O(\frac{2^{2^{2k}\lg k+2k+\lg k}}{k!})$. □

The second brute-force algorithm: We generate the set $\mathcal{K}(G')$ of all possible inclusion-wise maximal bicliques [19] in G', and try all possible k-subsets of $\mathcal{K}(G')$ to see whether one covers all edges in G'. Correctness of the algorithm is immediate since one can always restrict oneself to using only inclusion-wise maximal bicliques in a biclique edge cover. To generate all maximal bicliques, we first transform G' into the graph G'' given in Theorem 3. Thus, every inclusion-wise maximal biclique in G' is an inclusion-wise maximal clique in G''. We then use the algorithm of [20] on the complement graph $\overline{G''}$ of G'', *i.e.* the graph defined by $V(\overline{G''}) := V(G'')$ and $E(\overline{G''}) := \{\{u, v\}\colon\ u, v \in V(\overline{G''}),\ u \neq v,\ \text{and } \{u, v\} \notin E(G'')\}$.

Theorem 4. *The* Biclique Edge Cover *problem can be solved in $O(f(k) + \max(n, m)^3)$ time, where $f(k) := 2^{k2^{k-1}+3k}$.*

Proof. Given a bipartite graph G as an instance to Biclique Edge Cover, we first apply the kernelisation algorithm to obtain an equivalent graph G' with 2^k vertices, and then apply the brute-force algorithm described above to determine whether G' has a biclique edge cover of size k. Correctness of this algorithm follows directly from Section 4.1 and the correctness of the brute-force procedure. To analyse the time complexity of this algorithm, we first note that Prisner showed that any bipartite graph on n vertices has at most $2^{n/2}$ inclusion-wise maximal bicliques [20]. This implies that $|\mathcal{K}(G')| \leqslant 2^{2^{k-1}}$. The algorithm of [19] runs in $O(|V(G')||E(G')||\mathcal{K}(G')|)$ time, which is $O(2^k 2^{2k} 2^{2^{k-1}}) = O(2^{2^{k-1}+3k})$. Finally, the total number of k-subsets of $\mathcal{K}(G')$ is $O(2^{k2^{k-1}})$, and checking whether each of these subsets covers the edges of G' requires $O(|V(G')||E(G')|) = O(2^{3k})$ time. Thus, the total time complexity of the entire algorithm is $O(2^{2^{k-1}+3k} + 2^{k2^{k-1}+3k} + \max(n, m)^3) = O(2^{k2^{k-1}+3k} + \max(n, m)^3)$. □

It is worthwhile mentioning that some particular bipartite graphs have a number of inclusion-wise maximal bicliques which is polynomial in the number of their vertices. For these types of bipartite graphs, we could improve on the worst-case analysis given in the theorem above. For instance, a bipartite chordal graph G has at most $|E(G)|$ inclusion-wise maximal bicliques [20]. A bipartite graph with $n + m$ vertices and no induced cocktail-party graph of order ℓ has at most $\max(n, m)^{2(\ell-1)}$ inclusion-wise maximal bicliques [19]. The cocktail party graph of order ℓ is the graph with nodes consisting of two rows of paired nodes in which all nodes but the paired ones are connected with a graph edge (for a full definition, see [19]). Observing that the algorithm in Section 4.1 preserves chordality and does not introduce any new cocktail-party induced subgraphs, we obtain the following corollary:

Corollary 1. *The* Biclique Edge Cover *problem can be solved in $O(2^{2k^2+3k} + \max(n, m)^3)$ time when restricted to chordal bipartite graphs, and in $O(2^{2k^2(\ell-1)+3k} + \max(n, m)^3)$ time when restricted to bipartite graphs with no induced cocktail-party graphs of order ℓ.*

5. Experimental results

We saw in the previous section two FPT algorithms that, *given that the size of the solution is known to be at most k*, solve the Biclique Edge Cover problem, and thus the Mod/Resc Parsimony Inference problem (find one solution) taking as input the matrix obtained by applying the kernelisation rules described in Section 4.1 until no further rules can be applied. Of course, the order in which the rules are applied matter, and different orders may lead to different biclique coverings.

This leaves open solving the problem when the size of the solution is not known *a priori*. We give some indications below on how in practice we can address this issue. Should we become able to know a lower bound on the size of an optimal solution, the heuristic would enable to spell out such a solution as soon as one is found that has this size (and indicate that the lower bound is reached by the given input). Such a lower bound is known, as we discuss below. However, it remains an open question what is the complexity of computing it.

In addition to such theoretical difficulties, we also present some practical ones that may be met on the way to producing a realistic solution, such as missing data which may be common in this kind of application. We indicate a few approaches to solve this problem, that may lead to different solutions depending on the choice made.

It may be that in some cases and for a given order of application of the rules, the first step of kernelisation leads to a matrix that is empty (all rows and all columns have been eliminated by application of one of the rules). We know in this

case that there is a solution that has for size the number of times Rule 3 was applied. This solution is necessarily optimal, as proven by the correctness of the rules. This is what happens with the *Culex pipiens* dataset as we indicate below. However, for a different interpretation of the missing data for *Culex pipiens*, the kernelisation rules are not sufficient to find a solution and we need to perform the heuristic algorithm in order to find one.

Finally, we discuss another parameter that is biologically important and was not considered so far. This additional parameter which is related to the number of 1's in the solution (that is, in the matrices M and R), would require finding not just one optimal solution, but enumerating all optimal solutions. We do not discuss here the complexity of such an enumeration process, but describe an approach that will be reasonable only for relatively small and simple datasets. It remains also an open question devising good heuristics, a problem that we are currently addressing.

5.1. Heuristic algorithm

As we saw, a first difficulty in practice is to find the minimum size k of an optimal solution without running either of the algorithms described in Section 4.3 for all possible values of k, especially large ones.

Different approaches can be used. One possible would proceed by first checking if there is no solution of small sizes since this is easy to check using the *FPT* approach, and then increasing the size until reaching a smallest size k for which one solution exists. Another approach would proceed by using a fast and efficient heuristic to discover a solution of a given size k' that in general will be greater than the optimal size k sought. By then applying dichotomy (the optimal solution is between 1 and $k'-1$), the minimal size can be found using the *FPT* approach. This will be fine if k' is not too big, and thus the middle value between 1 and k' still small enough that one of the algorithms in Section 4.3 may run in reasonable time. Obviously, we should then further hope that the optimal size for a solution lies between 1 and this middle value rather than in the other half of the interval. Often, neither of the two approaches will be reasonable in practice and heuristics have to be used throughout.

One heuristic to obtain one upper bound k', the one we used, could be the following. For simplification, we assume C is a matrix on which no kernelisation rule can be applied anymore (or one obtained after all such possible rules have been applied). Since the order in which the rules are used matter, we may also consider as input to the heuristic below all matrices C that may be obtained by application of the kernelisation rules in any order (respecting the fact that Rule 4 may be used only if Rule 3 is not possible).

The heuristic is then as follows. We start by selecting the row in matrix C that contains the smallest number of 1's; this row is referred to as the *base row*. This step is referred to as Step 1. We then find the second row in C that has the largest intersection of 1's with the base row, that is for which there is the greatest number of positions where this and the base row both have a 1. Extending this approach to all rows in the matrix provides a way to determine the largest biclique using the base row as a reference. This step is referred to as Step 2. The 1's in C corresponding to the edges in this biclique are then labelled as *covered* and will represent one column in the M and R matrices. We then recurse by selecting the next row containing the smallest number of uncovered 1's. The process ends when all 1's are included in at least one biclique. This provides a solution of size k' to the problem that is an upper bound to the Biclique Edge Cover problem, and hence to the Mod/Resc Parsimony Inference problem.

It may be that at some point at either Step 1 or 2, we had more than one choice. We may then elect to try all possible options as each may lead to a different solution. We would then select for k' the best (smallest) value obtained.

5.2. Lower bound

Given the difficulty of the problem, it is useful to establish also a lower bound for the possible size of an optimal solution. In [14], various such bounds were discussed. For lower bound, we use the one discussed in [11] and related to the set of so-called *isolated ones*. This corresponds to a maximum set of locations (i, j) in the input matrix C such that $C(i, j) = 1$ and for each pair (i_1, j_1), (i_2, j_2) in the set either $C(i_1, j_2)$ or $C(i_2, j_2)$ contain a 0, or both. Fig. 3 illustrates the input matrix with a set of 8 isolated ones, marked in grey.

This bound has an important influence on the running time of the algorithm, as it indicates the step when the algorithm can halt. For instance, in the *Culex pipiens* case, showing that the isolated-ones set is of size 8 proves that the solution that has been found is indeed the most parsimonious.

5.3. Missing data

An important complication that is met in practice is missing data, *i.e.* locations in the input matrix C whose values are not known. This is the case in particular of the *Culex pipiens* dataset, but not exclusively. Other datasets present in general the same problem. In the *Culex pipiens* dataset, two kinds of missing data may be distinguished. One corresponds to missing observations (male and female strains that could not be crossed), the other to observations that could not be interpreted in an unambiguous way (the corresponding matrix cell contains thus both a 0 and a 1).

To deal with the problem of missing data, different approaches may be considered:

1. Decide that all the missing or ambiguous entries correspond in fact to compatible male/female strains. In other words, all missing or ambiguous entries are set to 0.

C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	0	0	1	0	0	0	1	1	0	0	0	0	0	1	1	0	1	1	1
2	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
3	0	1	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0
4	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
7	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
8	1	1	1	1	1	0	1	0	0	1	0	1	1	1	1	1	1	1	0
9	0	0	1	0	0	0	1	1	0	0	0	0	0	1	1	1	1	1	0
10	1	0	1	0	0	0	1	0	1	0	1	0	0	1	0	0	1	1	0
11	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
12	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
16	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 3. Isolated ones concept illustrated.

2. Set a value for missing data that corresponds to the majority of the values observed in either the same row (corresponding to a male strain) or column (corresponding to a female strain). This approach is probably the most realistic one, however it requires additional analysis and an expert evaluation in the case there remains ambiguity.
3. Check all combinations of values possible for the missing data. The next step, then, is to analyse the results for all such combinations to reach a decision. The additional difficulty here is related to the fact that the number of combinations can be enormous: this indeed leads to 2^{28} different matrices in the *Culex pipiens* case.

For the results obtained on the *Culex pipiens* case shown below, we adopted the first strategy. Others will be explored in a forthcoming biological paper and the corresponding results no longer discussed here.

5.4. Another important parameter

So far, the only parameter we considered in the optimisation procedure is k, the number of columns of the matrices M and R. However, there is another parameter that is also biologically relevant. This is the overall number of 1's in the solution, that is the number of 1's in M and R. These correspond indeed to the presence of a given gene in either the male or female strain. It may therefore be important to not only find the optimal value for k, but also to enumerate all optimal solutions for further analysis.

This problem is obviously hard since finding one solution already is. For small enough datasets and assuming we have already identified the size k of an optimal solution (see the heuristic above), one procedure could be to apply for instance the first brute-force algorithm described in Section 4.3 to the initial input matrix C (without application of any kernelisation rules), with two modifications: 1. we need to consider only all possible k-partitions of the edge set (*i.e.* those that have size k); 2. we must not stop once a first solution has been found but must consider all possible k-partitions as well as all possible ways of adding edges to each partition.

This approach will in general not produce all solutions in a reasonable time. It did in the case of the *Culex pipiens* dataset that is in a way "simple enough". Other methods will need to be explored for more general situations but are beyond the scope of this paper. Randomisation techniques may be useful here to obtain all solutions with a reasonable margin of error that some may be missed.

5.5. Results obtained on the Culex pipiens dataset

As mentioned, application of the kernelisation rules in a given order to the *Culex pipiens* dataset (setting all missing and ambiguous values to 0) led to eliminate matrix C. Since the given order contained 8 applications of Rule 3, and we could exhaustively compute the lower bound for this matrix, which is of 8, we know that 8 corresponds to the size of an optimal solution in this case.

Therefore, 8 pairs of mod/resc genes are required to explain the dataset. This appears to be in sharp contrast to simpler patterns seen in other host species [2,3,1] that had led to the general belief that cytoplasmic incompatibility can be

explained with a single pair of mod/resc genes. In biological terms, this result means that contrary to earlier beliefs, the number of genetic determinants of cytoplasmic incompatibility present in a single *Wolbachia* strain can be large, consistent with the view that it might involve repeated genetic elements such as transposable elements or phages.

We then applied the enumeration procedure described in the previous section. This led to 3976 different solutions, that is, 3976 pairs of mod and resc matrices with a minimum number of columns (genes). This number accounts for the fact that two solutions are identical if one solution can be obtained from another by a permutation of the columns (genes). Further discussion of these results is beyond the scope of this paper, and is left for a forthcoming biological paper.

The source code of the algorithm and the results on the *Culex pipiens* dataset can be viewed on the webpage pbil.univ-lyon1.fr/members/sagot/htdocs/code/Culex_FPT.cpp.

Acknowledgments

This work was funded by the French project ANR MIRI BLAN08-1335497 and the ERC Advanced Grant SISYPHE.

References

[1] S.R. Bordenstein, J.H. Werren, Bidirectional incompatibility among divergent Wolbachia and incompatibility level differences among closely related Wolbachia in Nasonia, Heredity 99 (3) (2007) 278–287.
[2] H. Merçot, S. Charlat, Wolbachia infections in Drosophila melanogaster and D. simulans: Polymorphism and levels of cytoplasmic incompatibility, Genetica 120 (1–3) (2004) 51–59.
[3] S.L. Dobson, E.J. Marsland, W. Rattanadechakul, Wolbachia-induced cytoplasmic incompatibility in single- and superinfected Aedes albopictus (Diptera: Culicidae), J. Med. Entomol. 38 (3) (2001) 382–387.
[4] R.G. Downey, M.R. Fellows, Parameterized Complexity, Springer-Verlag, 1999.
[5] O. Duron, C. Bernard, S. Unal, A. Berthomieu, C. Berticat, M. Weill, Tracking factors modulating cytoplasmic incompatibilities in the mosquito Culex pipiens, Mol. Ecol. 15 (10) (2006) 3061–3071.
[6] J. Engelstadter, G.D.D. Hurst, The ecology and evolution of microbes that manipulate host reproduction, Annu. Rev. Ecol. Evol. Syst. 40 (2009) 127–149.
[7] J. Engelstadter, A. Telschow, Cytoplasmic incompatibility and host population structure, Heredity 103 (2009) 196–207.
[8] H. Fleischner, E. Mujuni, D. Paulusma, S. Szeider, Covering graphs with few complete bipartite subgraphs, Theoret. Comput. Sci. 410 (21–23) (2009) 2045–2053.
[9] J. Flum, M. Grohe, Parameterized Complexity Theory, Springer, 2006.
[10] J. Gramm, J. Guo, F. Huffner, R. Niedermeier, Data reduction, exact, and heuristic algorithms for clique cover, in: Proceedings of the 8th ACM/SIAM Workshop on Algorithm Engineering and Experiments (ALENEX), 2006, pp. 86–94.
[11] D.A. Gregory, N.J. Pullman, Semiring rank: Boolean rank and nonnegative rank factorizations, J. Comb. Inf. Syst. Sci. 8 (1983) 223–233.
[12] H. Gruber, M. Holzer, Inapproximability of nondeterministic state and transition complexity assuming $P \neq NP$, in: Proceedings of the 11th International Conference on Developments in Language Theory (DLT), 2007, pp. 205–216.
[13] J. Guo, R. Niedermeier, Invitation to data reduction and problem kernelization, SIGACT News 38 (1) (2007) 31–45.
[14] S. Jukna, A.S. Kulikov, On covering graphs with complete bipartite subgraphs, Discrete Math. 309 (2009) 3399–3403.
[15] A.D. Korshunov, Asymptotic behaviour of Stirling numbers of the second kind, Diskret. Anal. 39 (1) (1983) 24–41.
[16] R. Niedermeier, Invitation to Fixed-Parameter Algorithms, Oxford University Press, 2006.
[17] J. Orlin, Contentment in graph theory: Covering graphs with cliques, Indag. Math. 80 (5) (1977) 406–424.
[18] D. Poinsot, S. Charlat, H. Merçot, On the mechanism of Wolbachia-induced cytoplasmic incompatibility: Confronting the models with the facts, Bioessays 25 (1) (2003) 259–265.
[19] E. Prisner, Bicliques in graphs I: Bounds on their number, Combinatorica 20 (1) (2000) 109–117.
[20] S. Tsukiyama, M. Ide, H. Ariyoshi, I. Shirakawa, A new algorithm for generating all the maximal independent sets, SIAM J. Comput. 6 (3) (1977) 505–517.