The bacterial strains characterization problem

Fabien Chhel, Adrien Goëffon, Antoine Lafosse, Frédéric Lardeux, Frédéric Saubion, Gilles Hunault, Tristan Boureau

Experimental Approach for Bacterial Strains Characterization

Fabien Chhel[1], Adrien Goëffon[1], Frédéric Lardeux[1], Frédéric Saubion[1], Gilles Hunault[2], and Tristan Boureau[3]

1: LERIA, 2: HIFIH, 3: PAVE, University of Angers (France),
firstname.lastname@univ-angers.fr

Abstract. In plant biology, data acquisition is no longer necessarily a major problem but nevertheless the treatment and the use of these data are still difficult. In this work, we are particularly interested by the characterization of strains of phytopathogenic bacterias, which is an important issue in the study of plant diseases. We study and compare several methods computing the smallest possible characterizations. These experiments have allowed us to characterize specific strains and diagnosis tests have been produced and used.

1 Introduction

This paper proposes to formalize and study a problem in plant biology, and more specially in biological diagnosis and characterization. We focus on bacterial strains of *Xanthomonas*, which is a genus of bacterias, many of which cause plant diseases. The name pathovar is a subdivision of the phytopathogenic bacterial species that corresponds to the strains causing the same symptoms on plant species or varieties of plant species. In particular, *Xanthomonas* are used in many studies because they include hundred of different pathovars. The approach consists in identifying, among the directory of strains, the relevant genes (virulence genes) and in analyzing the correlation between the presence / absence of these genes and the host specificity of the pathovars (groups of bacterial strains) [**?**].

In this context, the characterization problem corresponds to the identification of a group of strains against other groups, based on the presence or absence of particular genes. A strain is therefore a vector of binary values that reflects the presence (value 1) or absence (value 0) of these genes. More practically, a problem instance with 5 strains, divided into 3 groups based on a set of 4 genes can be illustrated by Fig. **??**.

Strain	Group	Genes			
		x1	x2	x3	x4
e1	g1	1	1	1	0
e2	g1	1	1	1	1
e3	g2	0	0	1	0
e4	g2	0	1	1	1
e5	g3	1	1	0	0

Fig. 1. Example of instance

Solving this problem consists in characterizing each group. Therefore, for each group, we must find a combination of presence or absence of genes that is valid for all strains of group and not valid for all other strains of other groups. In the example in Fig. **??**, group 1 is characterized by the simultaneous presence of genes x_1, x_2 and x_3.

There exists a real need to develop new approaches to provide characterization tools that take into account simultaneously several genes. In addition, biologists are interested in two specific properties of the solutions:

- A solution that minimizes the number of used characters: this is especially important for building diagnostic tests based on DNA chips. [**?**]. The number of oberved genes must be minimized for cost reasons, for avoiding long experiments and for insuring reliability. Another point is that it is easier to detect the presence of a gene rather than its absence.
- The computation of all solutions: it should be useful, in terms of biological interpretation, to have a representation of all possible solutions as it could highlight a special relationship between genes and explain some functional characteristics of the bacteria (for phenotypic considerations).

In this work, we begin by modeling this problem as the search for sets of propositional logic formulas, which allow us to study its satisfiability. In the experimental part, we examine different algorithmic techniques that allow us to obtain experimental results which have already been used for the development of diagnosis tests. Our purpose is not to design here the most efficient method for this problem, but rather to compare different possible approaches.

2 Problem Study

In this section the characterization problem is defined using propositional Since this problem is more general than the plant biology application suggested in the introduction, we use, in the remaining of this paper, the terms *entities* for bacteria strains and *characters* for the genes. The entities are organized in *groups*. Therefore the purpose of the characterization problem is to exhibit for each group a formula built over the sets of characters that identify the entities of the group against the entities of the other groups.

The description of the characterization problem by means of presence or absence of several characters, led us to naturally use propositional logic formalism. We consider a Boolean matrix corresponding to n characters and m entities:

$$A \equiv \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \vdots & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix}$$

Each row of this matrix represents an entity, characterized by the presence or absence of a set of characters. We consider then the characters as Boolean variables and the entities as Boolean assignments.

For every column index $j \in \{1, \ldots, n\}$, we define a propositional variable x_i which corresponds to a character. X is the initial set of propositional variables. For each row $i \in \{1, \ldots, m\}$, we consider an entity e_i as the corresponding Boolean interpretation, i.e. a mapping from X to $\{0, 1\}$ (false, true), such that $\forall i, j, e_i(x_j) = a_{ij}$. We denote $\mathcal{E}$ the set of all entities. Given a propositional formula ϕ on X and $e \in \mathcal{E}$, we denote $e \models \phi$ the fact that the interpretation e satisfies the formula.

The definition of the groups corresponds to sets of rows of the matrix A. If there are two identical lines belonging to the same group, we may remove one of them. In this case, each group is then a subset of entities of $\mathcal{E}$. Note that at this time, two identical entities can belong to two different groups. We will study this aspect latter with regards to the satisfiability of the problem.

We will use the classic vocabulary related to propositional logic. A literal is a variable $x \in X$ or its negation, denoted $\neg x$. A clause is a disjunction of literals. A clause is unitary if it contains only one literal. We note L the set of literals built on X. A clause is called positive if it contains only positive literals.

A formula ϕ is said to be in conjunctive normal form (CNF) if it is a conjunction of clauses. A formula is said to be in disjunctive normal form (DNF) if it is a disjunction of conjunctions of literals.

We can now define an instance of a characterization problem:

Definition 1. Instance of Characterization Problem *An instance of a characterization problem is defined by a tuple $(X, \mathcal{E}, G)$ where X is a set of propositional variables, $\mathcal{E}$ is a set of entities defined over X and $G \subseteq 2^{\mathcal{E}}$.*

We now focus on defining the characterization of a group that should allow to recognize its own entities and to discriminate (i.e., to not accept) the entities of other groups (from a logical point of view it will thus correspond to the satisfaction or refutation of formulas).

Definition 2. Group Characterization *Given an instance $(X, \mathcal{E}, G)$, a formula ϕ_g is said to characterize group $g \in G$ iff:*

$\forall e \in g, e \models \phi_g$ (accepts of the group's entities)
and
$\forall g' \in G \setminus \{g\}, \forall e' \in g', e' \not\models \phi_g$ (discriminates other groups' entities).

By extension, we denote $g \models \phi_g$ the fact that ϕ_g characterizes g according to the previous definition. A solution is then a set of formulas that characterize each group.

Definition 3. Solution of a Characterization Problem *Given an instance $P \equiv (X, \mathcal{E}, G)$, an admissible solution of a characterization problem is a set of formulas $\Phi = \{\phi_1, \cdots, \phi_{|G|}\}$ such that $\forall g \in G$, $g \models \phi_g$. $Sol(P)$ is the set of all admissible solutions of P.*

Given a set of formulas $\Phi = \{\phi_1, \cdots, \phi_{|G|}\}$ and a set of groups G, we denote by extension $G \models \Phi$ the fact that $\forall g \in G, \forall \phi_g \in \Phi, g \models \phi_g$.

Definition 4. Satisfiability *An instance $P \equiv (X, \mathcal{E}, G)$ is satisfiable (resp. unsatisfiable) iff $Sol(P) \neq \emptyset$ (resp. $Sol(P) = \emptyset$).*

As usual, the size of a Boolean formula, denoted $|\phi|$, corresponds to the number of different literals that it contains. We define the size of a set of formulas Φ as $|\Phi| = max_{\phi \in \Phi}(|\phi|)$. For example, the formula $(x \wedge y \wedge \neg z) \vee \neg y \vee (\neg x \wedge z)$ is of size 3.

Definition 5. Optimal Solution *An optimal solution of an instance $P \equiv (X, \mathcal{E}, G)$ is a set of formulas $\Phi^* \in Sol(P)$ such that $\forall \Phi \in Sol(P), |\Phi^*| \leq |\Phi|$.*

CAR-OPT is the problem that consists in finding an optimal solution for a satisfiable instance.

3 Resolution Methods

The purpose of this section is to study possible resolution approaches that can be used to solve the characterization problem. This problem can be considered from different points of view with respect to different computer science areas. Note that, as claimed in the introduction, we do not want to provide the best possible results in terms of computation time but rather propose different, even complementary, approaches.

In the introduction, we have recalled that the problem of finding a Boolean function from examples was an old problem in machine learning community and was thus also related to basic classification techniques. Therefore, we have chosen to use a machine learning technique that has been developed to learn definitions from examples. Of course, many other techniques could have been tested here. Nevertheless, the number of characters to be taken into account seemed quite large for SVM approaches for instance. The system FOIL [**?**] has been designed to learn Horn clauses from examples and appears thus well-suited to our problem, even if the notion of minimality is not addressed by this system.

Due to the complexity of the problem, we naturally turned to combinatorial optimization techniques and especially metaheuristcs algorithms that have largely demonstrated their efficiency on the resolution of large constrained optimization problems. We have chosen to use an evolutionary algorithm, since it had already experimented such an approach on a problem with similar complexity (i.e., computing an extension in default logic [**?**]). Here, the size of the formula is taken into account but the drawback of this technique is that, as any incomplete algorithm, it does not insure to reach a global optimum (i.e., a minimal solution) but only provides a solution of good quality (i.e., a short formula).

In order to guarantee the computation of the minimal characterization for each group, we have also implemented a complete search algorithm that aim at exploring the whole search space in order to build the minimal solution. As usual this kind of tree-based exploration will be faced to computational space and time limits.

More precisely, we will test the following algorithms:

1. Exact-CAR is an exact method based on a Branch-and-Bound algorithm, which guarantee that each solution found is minimal. Because of computational limitations, this algorithm is able to find only short characterization formulas, and stops if no formula shorter than a given bound (typically 4 or 5) is found.
2. GA-CAR is a fast approximate method using a steady-state genetic algorithm, in order to find short solutions in the general case, without guarantee of optimality.

3. FOIL is a machine learning algorithm. Contrary to GA-CAR, FOIL finds systematically one (or more) characterization formula. Consequently, the size of the shorter formulas given by FOIL constitute upper bounds of optimal solutions.

4 Experimental Results

The biologists requirements can be summarized as follows: a solution that minimizes the number of used characters and the set of all solutions. For small instances, it is possible to answer to these two objectives with an exact algorithm. Nevertheless, for large instances, the computation time is too long, especially concerning the second point. In order to find solution (not necessarily minimal), we also test a genetic algorithm and a learning approach (FOIL). Therefore, in this section we only address the first objective. Note that all the four instances provided by the biologists are satisfiable:

- **A** : 21 groups, 132 entities (from 2 to 10 by group), 38 characters.
- **B** : 8 groups, 108 entities (from 2 to 54 by group), 155 characters.
- **C** : 4 groups, 112 entities (from 5 to 69 by group), 155 characters.
- **D** : 7 groups, 112 entities (from 2 to 40 by group), 155 characters.

A is an instance about bacterial strains of *Xanthomonas* and B, C and D are instances coming from the BioMérieux API for *Ralstonia* species.

Table **??** presents the experimental results that we have obtained with our three algorithms. The four first columns provide the instance characteristics. The three last columns give the size of the formula obtained by each method for the groups of all the problems. Each method runs once for each group, except GA-CAR which is executed 20 times, due to its intrinsic nondeterministic nature. The results for GA-CAR correspond to the average of all runs. A sign "-" indicates that no result is found for a group (after 2 hours for Exact-CAR or after 100000 iterations for GA-CAR). Note that FOIL always return an answer but which may be close to the description of the instance itself.

For instance A, the table **??** shows that the three methods provide similar results. This instance is defined for only 38 characters and seems easy to solve. For instance B, it is interesting to see that results obtained by GA-CAR are very close to those of Exact-CAR, whereas those of FOIL are worst for several groups. Instances C and D seem to be the hardest ones. Exact-CAR finds a solution in less than 2 hours of running time for only 4 of the 11 groups. On the contrary, FOIL provides a formula for all the groups but their sizes are often very large. Finally, GA-CAR seems to be more reliable than the two other methods on hard problems because it obtains a solution for almost all the groups of the instances C and D. Moreover, all solutions have limited sizes (less than 10). The results obtained for groups 3 and 6 on the instance D highlight the power of the GA-CAR approach.

5 Conclusion

This article presents an approach based on propositional logic formalism for the characterization of bacterial strains. We first formalized this problem as the problem of finding

Instance	ent.	groups	ent.	Exact-CAR	GA-CAR	FOIL
A	130	1	5	3	3	3
		2	10	2	2	2
		3	5	4	4	7
		4	2	2	2	2
		5	5	4	4	5
		6	8	2	2	2
		7	6	4	4	6
		8	5	2	2	2
		9	5	3	3	3
		10	5	3	3	5
		11	6	3	3	4
		12	10	3	3	3
		13	6	2	2	2
		14	14	3	3	4
		15	8	3	3	3
		16	4	3	3	3
		17	4	3	3	3
		18	5	2	2	2
		19	7	2	2	2
		20	7	1	1	1
		21	2	1	1	2

Instance	ent.	groups	ent.	Exact-CAR	GA-CAR	FOIL
B	109	1	21	2	2	3
		2	5	2	2	6
		3	3	2	2	2
		4	54	5	5	6
		5	9	4	4	5
		6	8	3	4	117
		7	7	3	3	4
		8	2	3	3.4	3
C	113	1	31	3	3	4
		2	69	-	6	9
		3	8	-	5	6
		4	5	2	2	2
D	112	1	38	-	-	132
		2	15	-	-	120
		3	5	-	6	133
		4	6	-	5	6
		5	2	2	2	3
		6	40	-	6.4	99
		7	6	4	4	4

Fig. 2. Charaterizations obtained by the three methods: Exact-CAR, GA-CAR and FOIL.

a set of minimal propositional formulas and we exhibit some conditions for its satisfiability. In the second part of this paper, we have proposed three different resolution approaches for this problem in order to provide practical results and to highlight the respective benefits and drawbacks of these techniques. We have to mention that these results have already been used by biologists to define diagnosis tests and that a patent is currently under consideration.

References

1. A. Hajri, C. Brin, G. Hunault, F. Lardeux, C. Lemaire, C. Manceau, T. Boureau, and S. Poussier. A "repertoire for repertoire" hypothesis: Repertoires of type three effectors are candidate determinants of host specificity in xanthomonas. *PLoS ONE*, 4(8):e6632, 08 2009.
2. P. Nicolas, F. Saubion, and I. Stéphan. Gadel: a genetic algorithm to compute default logic extensions. In *Proc. ECAI'00,*, pages 484–490. IOS Press, 2000.
3. J. Ross Quinlan. Learning logical definitions from relations. *Machine Learning*, 5:239–266, 1990.
4. M. Schena, D. Shalon, R.W. Davis, and P.O. Brown. Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, 270(5235):467–470., 2005.