Two Strikes Against the Phage Recombination Problem

Manuel Lafond[1], Anne Bergeron[2], and Krister M. Swenson[3]

[1] Université de Sherbrooke, Canada
[2] Université du Québec à Montréal, Canada
[3] CNRS, LIRMM, Université de Montpellier, France

Abstract. The recombination problem is inspired by genome rearrangement events that occur in bacteriophage populations. Its goal is to explain how to transform a bacteriophage population into another using the minimum number of recombinations. Here we show that the general combinatorial problem is NP-Complete, both when the target population contains only one genome of unbounded length, and when the size of the genomes is bounded by a constant. In the first case, the existence of a minimum solution is shown to be equivalent to a 3D-matching problem, and in the second case, to a satisfiability problem. These results imply that the comparison of bacteriophage populations using recombinations will have to rely on heuristics that exploit biological constraints.

1 Introduction

Genetic recombinations or, more generally, the exchange of DNA material between organisms, have been a source of computational problems since the 1865 report of Gregor Mendel on plant hybridization [1]. Recombinations occur in the reproduction of all living organisms, including asexual reproduction, and are fundamental producers of diversity. In this paper, we study the computational complexity of problems related to *modular recombination*, which is a form of exchange pervasive in viruses that infect bacteria, called *phages*.

The biological theory of modular recombination was proposed a few decades ago by Botstein [4], who envisioned " *... viruses as belonging to large interbreeding families, members of which share only a common genome organization consisting of interchangeable genetic elements each of which carries out an essential biological function.*" The common genome organization that Botstein refers to is the preservation of the order of biological functions, called *modules*, along the virus genome, although the actual sequences that carry the function may diverge substantially.

The computational models were slower to emerge, since genomic data about "large interbreeding families" were not commonplace until a few years ago. In 2010 a study of a few dozen sequenced strains of *Staphyloccoccus aureus* was conducted [9], and a scenario of interbreeding was inferred on the population [14]. The recent availability of other datasets monitoring phage populations evolving through time [11,12] or space [7,13] suggested the problem of computing the minimum number of recombination events that transforms one population of phages into another. In a previous paper [2] we developed a heuristic with approximation bounds based on certain properties of the input and found that, on phages infecting bacteria responsible for cheese fermentation, our heuristic performed well. The question remained, however, as to the computational complexity of the optimization problem.

We answer that question in this article, showing that two basic problems related to the comparison of phage populations are computationally difficult. The first one reduces the problem of finding a perfect 3D-matching to reconstructing a single phage, from a population of phages that represents the triples of the 3D matching instance, with a minimum number of recombinations. The second one reduces a variant of a classic satisfiability problem to the reconstruction of a population of phages, with only 4 modules that represent variables and clauses, with a minimum number of recombinations.

2 Basic definitions and properties

Phage genomes can adopt either a circular or linear or shape during their life cycle. Genomic data found in databases are linearized by choosing, as a starting point, one module shared by all members of a family, yielding the following representation of phages.

Given an alphabet $\mathcal{A}$, a phage p with n modules can be represented by $p = p[0..n-1]$ where $p[a] \in \mathcal{A}$. The *recombination* operation at positions a and b between two phages p and q :

$$\begin{aligned} p &= p[0..a-1]|p[a..b-1]|p[b..n-1] \\ q &= q[0..a-1]|q[a..b-1]|q[b..n-1] \end{aligned}$$

yields new phages c and d:

$$\begin{aligned} c &= p[0..a-1]|q[a..b-1]|p[b..n-1] \\ d &= q[0..a-1]|p[a..b-1]|q[b..n-1]. \end{aligned}$$

Positions a and b are called the *breakpoints* of the recombination. The recombining phages are called *parents*, and the newly constructed phages, their *children*. This relation allows us, when several recombinations are considered, to refer to *descendants* and *ancestors*, of both phages and positions; each recombination creates two descendants to the two parents, while the each character in each of the children has exactly one ancestral character from the parents. Note that, naturally, ancestor and descendant relationships are transitive through the generations.

A *recombination scenario* S from $\mathcal{P}$ to $\mathcal{Q}$ is a sequence of recombinations that constructs all phages of $\mathcal{Q}$ using phages of $\mathcal{P}$ and their descendants. Note that no phage is discarded in the process, in the sense that $\mathcal{P}$ *grows* until it is a superset of $\mathcal{Q}$.

The problem that we address in this article is the following:

Minimum Phage Population Reconstruction (MinPPR)

Input: Populations $\mathcal{P}$ and $\mathcal{Q}$ of equal-length phages, and an integer r.

Question: Does there exist a recombination scenario S from $\mathcal{P}$ to $\mathcal{Q}$ of length at most r?

A *break* in a phage q with respect to the set of phages $\mathcal{P}$ is a position b such that for all parents p in $\mathcal{P}$, $p[b-1..b] \neq q[b-1..b]$. A recombination *heals* a break b of a phage q if it creates a child c such that $c[b-1..b] = q[b-1..b]$. In order to be healed, a break b must be one of the breakpoints of the recombination.

Since a recombination can heal at most two breaks in a single phage, if a phage q has n breaks with respect to the set of phages $\mathcal{P}$, then the minimum number of recombinations to construct q is $\left\lfloor \frac{n+1}{2} \right\rfloor$.

A crucial remark is that, even if all the breaks are healed, the reconstruction of a phage q with $n = 2r$ breaks with respect to a set of parents might require more than r recombinations. This is the case, for example, if two parents $p_1 = 10111$ and $p_2 = 11101$ are used to reconstruct $q = 11111$: phage q has no break with respect to the set $\{p_1, p_2\}$, but one recombination is necessary to reconstruct q. This recombination must cut an already healed break in p_1 or p_2, and we say that the break is *reused*.

Definition 1. *In a recombination scenario, a break is said to be* reused *if it is a breakpoint of more than one recombination in the scenario.*

Finally, there is an easy upper bound for the number of recombination necessary to reconstruct a phage:

Proposition 1. *If there exists a scenario that reconstructs a phage q with n modules from a population $\mathcal{P}$, then there exists one of length at most $n-1$.*

Proof. A scenario exists if, for each position b, there exists a phage $p_b \in \mathcal{P}$ such that $p_b[b] = q[b]$, otherwise no recombination can produce the value $q[b]$ at position b. We first recombine p_0 and p_1 using breakpoints 1 and 2, to produce a child that equals q on its first 2 positions, and proceed in a similar way up to position $n-1$. □

Here we study the decision problem where one asks if $\mathcal{Q}$ can be generated from $\mathcal{P}$ using at most r recombinations, for some given r. Let us first argue that the problem is in NP. A given scenario of r recombinations can be verified in time proportional to $r, |\mathcal{P}|$, and $|\mathcal{Q}|$, but this is not polynomial if r is not polynomial in $|\mathcal{P}|$ and $|\mathcal{Q}|$ (e.g. if r is exponential). However, Proposition 1 gives an upper bound on the number of required recombinations based on the number of modules. Hence, we may assume that r is bounded by a polynomial in $|\mathcal{P}|$ and $|\mathcal{Q}|$ and a scenario can be verified in polynomial time, and thus the problem is in NP.

3 Reconstructing one target genome

We first consider the case in which the population $\mathcal{Q}$ consists of a single phage of unbounded length. We reduce the 3D-PERFECT-MATCHING problem to it, where we receive a set of triples $T = \{(i_1, j_1, k_1), \ldots, (i_n, j_n, k_n)\} \subseteq [1..m]^3$, where $m \geq 2$ is an integer [10]. The goal is to find a subset $T' \subseteq T$ of size m such that for any two distinct $(i, j, k), (i', j', k') \in T'$, we have $i \neq i', j \neq j'$, and $k \neq k'$. Such a set T' is called a perfect 3D-matching.

Since there is a single phage in $\mathcal{Q}$, the alphabet is the set $\{0, 1\}$, and $Q = 11111 \ldots 1111$ will be the only element of the target population. We consider the following phages, each of length $15m + 2$, that form the input population $\mathcal{P}$. See example in Figure 1.

1. For each element $(i, j, k) \in T$, we construct a phage P_{ijk} that has three 1's in positions $5i$, $5j + 5m$ and $5k + 10m$, and 0's elsewhere.
2. For each element $(i, j, k) \in T$, we associate three phages, $P_{ij\text{-}}$, $P_{\text{-}jk}$, and $P_{i\text{-}k}$ with two 1's respectively in positions $5i + 1$ and $5j - 1 + 5m$, $5j + 1 + 5m$ and $5k - 1 + 10m$, $5i - 1$ and $5k + 1 + 10m$, and 0's elsewhere.
3. $\hat{P}$ has 0's in every position in which one of the above phages has a 1.

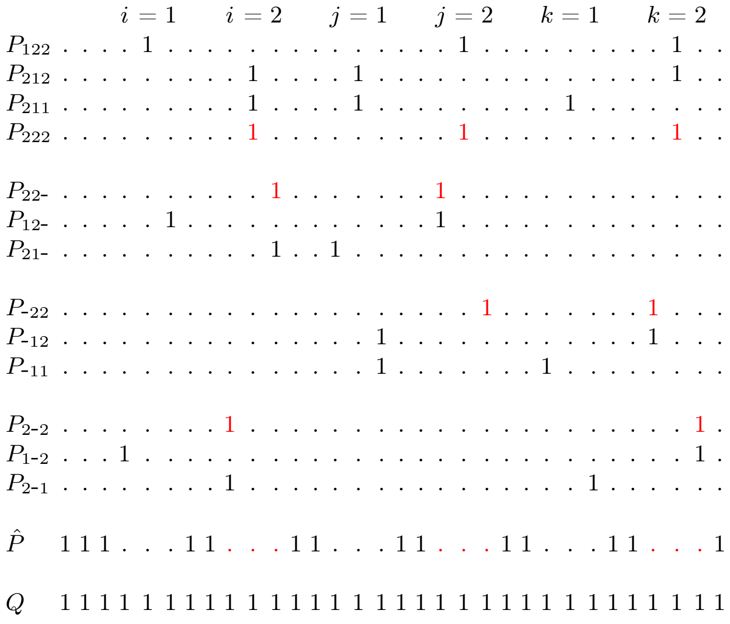

Fig. 1: Example input with $T = \{(1, 2, 2), (2, 1, 2), (2, 1, 1), (2, 2, 2)\}$ and $m = 2$. The 1's related to phage P_{222} are in red. Dots are used to represent the value 0, in order to better highlight the relative positions of the 1's.

We show that T has a 3D-matching if and only if $\mathcal{P}$ can generate Q with at most $6m$ recombinations, implying:

Theorem 1. *The* MINIMUM PHAGE POPULATION RECONSTRUCTION *problem is NP-complete, even when population $\mathcal{Q}$ has a single phage.*

We have already established that the problem is in NP. For NP-hardness, we show that there exists a 3D-matching $T' \subseteq T$ if and only if it is possible to reconstruct $\mathcal{Q}$ from $\mathcal{P}$ using at most $6m$ recombinations.

3.1 The ($\Rightarrow$) direction

Suppose that there is a 3D-matching $T' \subseteq T$. For each $(i, j, k) \in T'$, it is possible to apply three recombinations to $P_{ijk}, P_{ij\text{-}}, P_{\text{-}jk}$ and $P_{i\text{-}k}$ to obtain $00\ldots01110\ldots01110\ldots01110$, where the first 111 is centered at column i, the second 111 is centered at column j, the third 111 is centered at column k. The genome $\hat{P}$ has 000 in these three triples of positions, and so with three more recombinations we can bring in these 111 into $\hat{P}$. Use Figure 2 as an illustration. This costs 6 events. Since T' is a perfect 3D-matching, we can repeat this m times to fill in all the remaining 000's in $\hat{P}$, hence achieving cost $6m$.

3.2 The ($\Leftarrow$) direction

We next show that if Q can be reconstructed from $\mathcal{P}$ using at most $6m$ recombinations, then T admits a 3D-matching. We first establish several properties that hold in general for scenarios that transform $\mathcal{P}$ into Q, before proving the main result of the section.

Given a scenario S that reconstructs phage Q, we identify the following subsets of parents:

1. $S_{\mathbf{ijk}}$ contains phages of the form P_{ijk} that belong to the scenario.
2. $S_{\mathbf{xy}}$ contains phages of the form $P_{ij\text{-}}, P_{\text{-}jk}$ or $P_{i\text{-}k}$ that belong to the scenario.

Let $\mathcal{P} = S_{\mathbf{ijk}} \cup S_{\mathbf{xy}} \cup \{\hat{P}\}$ be the set of parents that initially belong to scenario S. We prove that scenario S reconstructs phage Q in $6m$ recombinations only if the set $S_{\mathbf{ijk}}$ corresponds to a perfect matching.

By construction, phage Q has $12m$ breaks with respect to the set of phages $\mathcal{P}$. Since a recombination can heal at most two breaks of a single phage, we need at least $6m$ recombinations to reconstruct Q. In a scenario of length $6m$, no break can be reused.

We distinguish two types of breaks: *red* breaks connect a phage in $S_{\mathbf{xy}}$ to a phage in $S_{\mathbf{ijk}}$, and *green* breaks connect a phage in $S_{\mathbf{xy}}$ to phage $\hat{P}$, (see Figure 2). We say that a recombination is *red* when its two breaks are red, and *green* if they are green. There is an equal number of red and green breaks in Q, thus, in a scenario of length $6m$, the number of red recombinations is equal to the number of green recombinations, and is at most $3m$, allowing for eventual red-green recombinations.

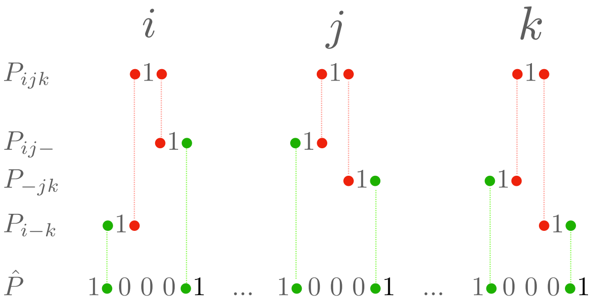

Fig. 2: Red and green breaks. *Red* breaks connect a phage in $S_{\mathbf{xy}}$ to a phage in $S_{\mathbf{ijk}}$, and *green* breaks connect a phage in $S_{\mathbf{xy}}$ to phage $\hat{P}$. Phage Q has $6m$ red breaks and $6m$ green breaks.

The following easy result links properties of a recombination scenario to the existence of a perfect matching:

Lemma 1. *In any scenario that reconstructs Q, $|S_{\mathbf{ijk}}| \geq m$. If $|S_{\mathbf{ijk}}| = m$, then the set $\{(i,j,k)|P_{ijk} \in S_{\mathbf{ijk}}\}$ is a perfect matching.*

Proof. In order to reconstruct Q, all values of i, j and $k \in [1..m]$ must appear at least once in the indices of elements $P_{ijk} \in S_{\mathbf{ijk}}$, thus $|S_{\mathbf{ijk}}| \geq m$. If $|S_{\mathbf{ijk}}| = m$, then all values of i, j and $k \in [1..m]$ appear exactly once implying that the set $\{(i,j,k)|P_{ijk} \in S_{\mathbf{ijk}}\}$ corresponds to a perfect matching. See Figure 3 for an example. □

In order to show that $|S_{\mathbf{ijk}}| = m$, we first introduce three lemmas that constrain the order of recombinations contained in a scenario of length $6m$. The first one concerns the *red interval* of a phage in $S_{\mathbf{xy}}$, which contains the 0's adjacent to its red breaks, along with the (circularly) intervening columns that are all 0's. See Figure 4 for an example of a red interval.

Lemma 2 (red interval). *All descendants of a phage $p \in S_{\mathbf{xy}}$ must heal red breaks shared with p before acquiring a 1 in p's red interval.*

Proof. Consider a descendant of p where one of its red breaks b is not yet healed, along with the first recombination producing a child c that contains b and a 1 in p's red interval. If this recombination does not heal b, then it must first heal a break between b and the 1 in p's red interval, thereby creating a phage c containing both the 1, and the break b. This implies that the second break β of this recombination must be in p's red interval, which is a contradiction since the second break cannot be healed in an interval with all 0's. See Figure 4 for an illustration. □

The second lemma establishes the property that all four breaks spanning two consecutive groups of 1's in $\hat{P}$ will be healed in the same ancestral lineage of Q.

```
             i=1       i=2       j=1       j=2       k=1       k=2
P122  . . . . 1 . . . . . . . . . . . . . . 1 . . . . . . . . . 1 . .
P211  . . . . . . . . . 1 . . . . 1 . . . . . . . . . 1 . . . . . . .

P12-  . . . . . 1 . . . . . . . . . . . . 1 . . . . . . . . . . . . .
P21-  . . . . . . . . . . 1 . . 1 . . . . . . . . . . . . . . . . . .

P-11  . . . . . . . . . . . . . . . 1 . . . . . . . 1 . . . . . . . .
P-22  . . . . . . . . . . . . . . . . . . . . 1 . . . . . . . 1 . . .

P1-2  . . . 1 . . . . . . . . . . . . . . . . . . . . . . . . . . 1 .
P2-1  . . . . . . . . 1 . . . . . . . . . . . . . . . . 1 . . . . . .

P̂     1 1 1 . . . 1 1 . . . 1 1 . . . 1 1 . . . 1 1 . . . 1 1 . . . 1

Q     1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Fig. 3: A possible output for the example of Figure 1, with the sets $S_{\mathbf{ijk}}$ and $S_{\mathbf{xy}}$ used by a recombination scenario of length $6m = 12$. Here $|S_{\mathbf{ijk}}| = m$, and $\{(1, 2, 2), (2, 1, 1)\}$ is a perfect matching.

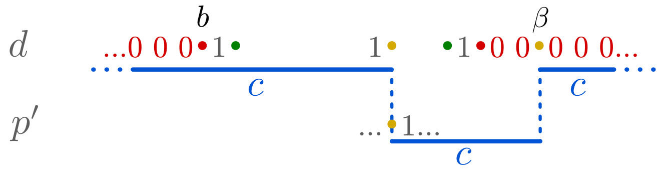

Fig. 4: A recombination between a descendant d of $p \in S_{\mathbf{xy}}$ and a phage p'. The recombination creates phage c, in blue, which contains both break b and a 1 from a phage p', where the 1 is located in a column from p's red interval. This recombination must have one breakpoint between b and the other red break of p, and one breakpoint β after the location of the 1 in phage p'. Breakpoint β cannot heal a break because it is between two 0's.

Lemma 3 (sticky breaks). *Consider a break that is healed when phage p is produced by a recombination in a scenario of length $6m$. Any adjacent break must be healed in a descendant of p.*

Proof. Consider the 1 in column x that is adjacent to a single healed break in phage p, so that $p[x-1..x+1] = 110$, or $p[x-1..x+1] = 011$. Consider the 1 that is put in column $x+1$, or $x-1$, while healing the remaining break with column x, producing some child c. Since any break is healed exactly once, the 1's to either side of a healed break must be ancestors of 1's in Q. Therefore, the 1 in column x of both phages p and c must be the ancestor of a 1 in Q, which is only possible if c is a descendant of p. □

Lemma 4 (independent $S_{\mathbf{xy}}$). *Consider phages p_1 and p_2 in $S_{\mathbf{xy}}$. There cannot exist a descendant of both phages with an unhealed red break from both p_1 and p_2.*

Proof. Define the green interval of a phage $p \in S_{\mathbf{xy}}$ to be the interval containing the 0's next to the green breaks in p, along with the (circularly) intervening columns that are all 0's. If the green intervals of p_1 and p_2 do not intersect, the red interval lemma gives the result, since there can be no 1 in either red interval before both of the red breaks in a phage are healed.

Suppose their green intervals intersect and, without loss of generality, that the green interval for p_1 starts to the left of the green interval of p_2. Say that there is a descendant containing the left 1 of p_1 and the left 1 of p_2. The red interval lemma implies that a recombination happened directly to the left of the 1 in p_2, which healed that red break. Say that there is a descendant with the left 1 of p_1 and the right 1 of p_2. Due to the red interval lemma, this implies a recombination happened directly to the right of the 1 in p_2, which healed that red break. By symmetry, the other cases are covered by those already listed. □

The next lemma states a desirable property of recombination scenarios of length $6m$, saying that if a phage is in $S_{\mathbf{xy}}$, then its two – unique – siblings are also in $S_{\mathbf{xy}}$. We say that phages p and p' *eventually recombine* in a scenario S if there exists a recombination in S between p, or one of its descendants, and p', or one of its descendants.

Lemma 5. *In a scenario S of length $6m$, if a phage $P_{ijk} \in S_{\mathbf{ijk}}$ eventually recombines using the red breaks of a phage in $S_{\mathbf{xy}}$, then all three phages $P_{ij\text{-}}$, $P_{\text{-}jk}$ and $P_{i\text{-}k}$ eventually recombine with P_{ijk} using both of their red breaks.*

Proof. Consider the first time that a phage P_{ijk} appears in scenario S, and suppose that this recombination involves a phage in $S_{\mathbf{xy}}$, or its descendant. Without loss of generality we may assume this phage is $P_{i\text{-}k}$, due to the circularity of the genomes and symmetry of our construction. We will show that both $P_{ij\text{-}}$ and $P_{\text{-}jk}$ must eventually recombine with P_{ijk} in the scenario, using both of their red breaks.

By the lemma statement, both red breaks of $P_{i\text{-}k}$ were healed in this recombination producing some child c, having only 0's between columns i and k, with the exception of column j. See Figure 5 for an illustration. Now, the recombination healing the remaining red break adjacent to column i must be in a descendant of c, due to the sticky breaks lemma. Aside from the descendant of c, the other parent p in this recombination must be a descendant of some phage $P_{ij'\text{-}} \in S_{\mathbf{xy}}$, for some j'.

In the following, we show that the only possible companion breakpoint for this recombination occurs when $P_{ij'\text{-}} = P_{ij\text{-}}$. Since every recombination must heal two breaks, there is a companion breakpoint between columns i and j, between columns j and k, or (circularly) between columns k and i.

If the companion breakpoint lies (circularly) between k and i, this implies that p has a 1 in the red interval of $P_{ij'\text{-}}$, which is impossible by Lemma 2.

Say the companion breakpoint lies between columns i and j. In order to heal two breakpoints, there must be a break b between columns i and j in a descendant p' of c. If b is adjacent to j, then it can only be healed using a 1 descending from a phage in $S_{\mathbf{xy}}$. If this phage is not $P_{ij\text{-}}$, then the independent $S_{\mathbf{xy}}$ lemma prohibits a common descendant between this phage and $P_{ij'\text{-}}$, a contradiction. Say b is not adjacent to j, but rather in the zone of all 0's in c. Then the existence of b implies that there has been a recombination at the breakpoint adjacent to column j in an ancestor of p'. This leads to the same contradiction as in the previous case.

The same argument applies to a breakpoint occurring between j and k.

Now consider the symmetric case, where a recombination heals the remaining break adjacent to column k in phage c. The same reasoning shows that both red breaks of $P_{\text{-}jk}$ are used to recombine with a descendant of P_{ijk}. □

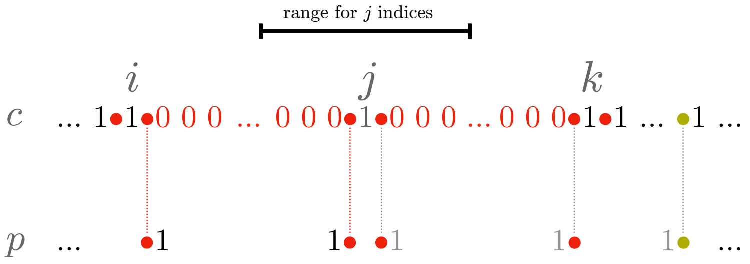

Fig. 5: At the creation of child c, it has only 0's between columns i and k, with the exception of column j. The gray 1's in p are possible breaks that can be healed.

The previous lemma depends on the assumption that the first recombination with a phage in $S_{\mathbf{ijk}}$ heals the two red breaks of a phage in $S_{\mathbf{xy}}$. The following lemma show that this must be the case.

Lemma 6. *In a scenario S of length $6m$, the first recombination using $P_{ijk} \in S_{\mathbf{ijk}}$ must heal both red breaks of a phage in $S_{\mathbf{xy}}$, be it $P_{ij\text{-}}$, $P_{\text{-}jk}$, or $P_{i\text{-}k}$.*

Proof. Since a phage $P_{ijk} \in S_{\mathbf{ijk}}$ has only red breaks, it must eventually recombine using exactly two red breaks, b_1 and b_2. Suppose that P_{ijk} does not eventually recombine with a single phage of $S_{\mathbf{xy}}$, then b_1 and b_2 are breaks on different phages p_1 and p_2 of $S_{\mathbf{xy}}$. This implies that p_1 and p_2 eventually recombine to produce a child containing both b_1 and b_2. Such a recombination is impossible, due to Lemma 2. □

Thus we have the result:

Proposition 2. *A scenario S reconstructs Q in $6m$ recombinations only if the set $S_{\mathbf{ijk}}$ is a perfect matching.*

Proof. Lemma 5 shows that for each P_{ijk} in $S_{\mathbf{ijk}}$, the three corresponding $P_{ij\text{-}}$, $P_{\text{-}jk}$ and $P_{i\text{-}k}$ must belong to the scenario. Since there are $3m$ pairs of red breaks, the maximum number of elements of $S_{\mathbf{ijk}}$ is m. Lemma 1 gives the result. □

4 NP-hardness for genomes of length 4

In the preceding section, we showed that the Minimum Phage Population Reconstruction was hard when the length of the genomes was unbounded. Is it still the case for genome of bounded length? The answer is yes, and we dedicate the remainder of the section to the proof of the following statement:

Theorem 2. *The* Minimum Phage Population Reconstruction *problem is NP-complete, even when the genomes of P and Q have length 4.*

We reduce from the Balanced-4Occ-SAT problem, where we are given a boolean formula ϕ in conjunctive normal form, such that each variable has exactly two positive occurrences in the clauses of ϕ, and exactly two negative occurrences [3].

Consider an instance ϕ of Balanced-4Occ-SAT with variables $x_1, \ldots, x_n$ and clauses $C_1, \ldots, C_m$. We construct a corresponding instance $(\mathcal{P}, \mathcal{Q}, r)$ of the phage problem. See Figure 6 for an example with 3 variables and 4 clauses, and Figure 7 for a more abstract view.

The alphabet for the phages of $\mathcal{P}$ and $\mathcal{Q}$ has, for each variable x_i, a corresponding symbol $i \in [1..n]$, and for each clause C_j, a corresponding symbol c_j. We also add two unique symbols '$-$' and '$\circ$' to the alphabet.

Consider a variable x_i, where $i \in [1..n]$. Let C_g, C_h be the clauses in which x_i occurs positively, and C_r, C_s those in which x_i occurs negatively. Add the following phages to $\mathcal{P}$:

$$\begin{aligned} X_i &= i \;\circ\; i \;\; i \\ X_i^+ &= c_g \; i \; c_h \; - \\ X_i^- &= c_r \; i \; c_s \; - \end{aligned}$$

and add the following to $\mathcal{Q}$:

$$X_i^* = i\ i\ i\ i$$

Now for each clause C_j, $j \in [1..m]$, add the following to $\mathcal{P}$:

$$\begin{aligned} D_{j,1} &= -\ -\ c_j\, c_j \\ D_{j,2} &= c_j\ -\ -\, c_j \end{aligned}$$

and add the following to $\mathcal{Q}$:

$$D_j^* = c_j \circ c_j c_j$$

We show that ϕ is satisfiable if and only if $\mathcal{Q}$ can be reconstructed from $\mathcal{P}$ with at most $r = n + m$ recombinations.

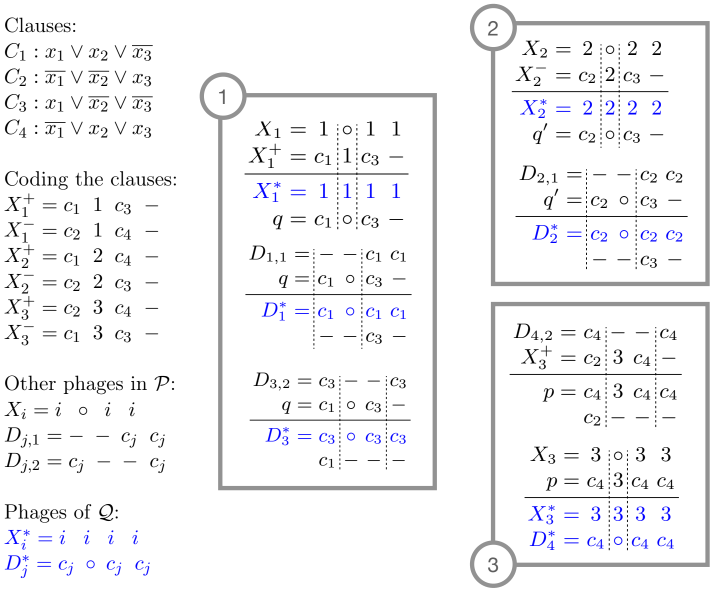

Fig. 6: In this example there are $n = 3$ variables and $m = 4$ clauses, thus $n+m = 7$ phages in $\mathcal{Q}$. One possible recombination scenario of length 7 is depicted. The three recombinations in Group 1 first construct X_1^* using X_1^+ that asserts that clauses 1 and 3 are satisfied when variable x_1 is *true*. The resulting phage q is then used to generate both D_1^* and D_3^*. In Group 2, X_2^* and D_2^* are constructed using X_2^- that asserts that clause 2 is satisfied when when variable x_2 is *false*. Group 3 shows an alternative strategy that first constructs phage $p = c_4\ 3\ c_4\ c_4$, and uses it to simultaneously construct X_3^* and D_4^*.

4.1 The (⇒) direction

Suppose that ϕ is satisfied by an assignment $\alpha : \{x_1, \ldots, x_n\} \rightarrow \{true, false\}$ of the variables. Let us produce $\mathcal{Q}$ from $\mathcal{P}$. For each $i \in [1..n]$, if $\alpha(x_i) = true$, then recombine X_i with X_i^+ by exchanging 2nd positions:

$$\begin{array}{rc:c:cc}
X_i = & i & \circ & i & i \\
X_i^+ = & c_g & i & c_h & - \\
\hline
X_i^* = & i & i & i & i \\
p = & c_g & \circ & c_h & -
\end{array}$$

This produces children X_i^* and $p = c_g \circ c_h -$, where C_g and C_h are the clauses that are satisfied by setting x_i to *true*. At this point, if D_g^* is not already in $\mathcal{Q}$, then recombine $D_{g,1} = - - c_g c_g$ with $p = c_g \circ c_h -$ by exchanging positions 1 and 2, thereby producing D_g^*:

$$\begin{array}{r:cc:cc}
D_{g,1} = & - & - & c_g & c_g \\
p = & c_g & \circ & c_h & - \\
\hline
D_g^* = & c_g & \circ & c_g & c_g \\
 & - & - & c_h & -
\end{array}$$

For an illustration of the previous two recombinations, see the black edges in Figure 7. In the same way, if D_h^* is not already in $\mathcal{Q}$, recombine $D_{h,2} = c_h - - c_h$ with $c_j \circ c_h -$ by exchanging positions 2 and 3, which produces D_h^*.

If instead $\alpha(x_i) = false$, recombine X_i with X_i^- by exchanging 2nd positions. This creates X_i^* and $c_r \circ c_s -$, where C_r and C_s are satisfied by setting x_i to *false*. Produce D_r^* and D_s^*, if not already there, as in the previous case.

Since every clause C_j is satisfied by α, for each D_j^*, there will be some X_i in the above procedure that produces D_j^*. Moreover, there are exactly $n + m$ recombinations: one to produce each X_i^*, and one to produce each D_j^*.

4.2 The (⇐) direction

Suppose that there exists a sequence $S = (R_1, \ldots, R_r)$ of at most $r \leq n + m$ recombinations that reconstructs $\mathcal{Q}$. Let $\mathcal{X} = \{X_1, \ldots, X_n\}$ and $\mathcal{D} = \{D_1, \ldots, D_m\}$ where D_j is either $D_{j,1}$ or $D_{j,2}$, whichever contains the character c_j that is the ancestor of the c_j in the 4th position of D_j^*.

We define a function $f : \mathcal{X} \cup \mathcal{D} \rightarrow S$ in the following way: set $f(X_i)$ to the first recombination that creates X_i^*, or an ancestor of X_i^*, with $[1..n]$ in the 2nd position and i in the 4th. Note that the parent used by $f(X_i)$, having i in the 4th position, must also have $\{-, \circ\}$ in the 2nd position, as otherwise there would be a previous recombination with the required properties for being $f(X_i)$.

Set $f(D_j)$ to the first recombination that creates D_j^*, or an ancestor of D_j^*, with $[1..n] \cup \{\circ\}$ in the 2nd position and c_j in the 4th. As previously, note that one of the parents used by $f(D_j)$ contains '$-$' in the 2nd position and c_j in the 4th position.

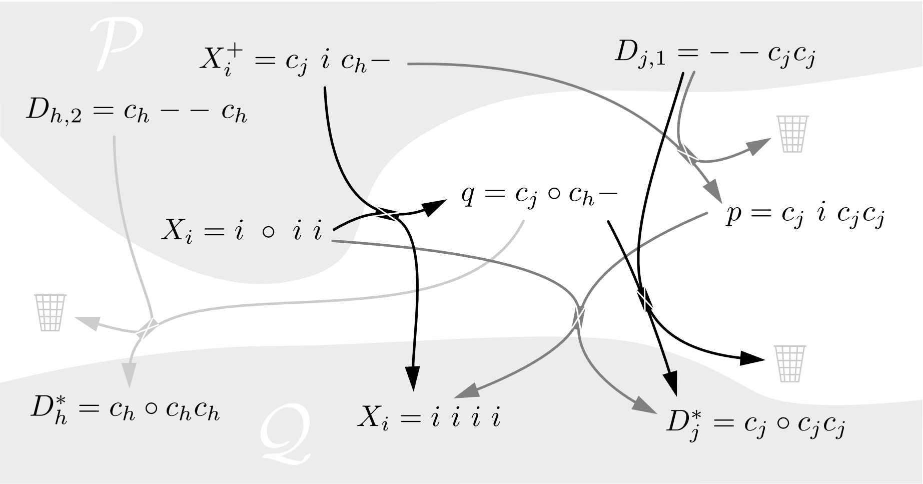

Fig. 7: An illustration of how phages X_i, X_i^+, $D_{j,1}$, and $D_{h,2}$ can recombine to produce X_i^*, D_j^*, and D_h^*. Recombinations are represented by pairs of arrows that meet in the middle at a box with a ⧖ symbol. In black, X_i and X_i^+ first recombine to produce $q = c_j \circ c_h -$, which then recombines with $D_{j,1}$ to produce D_j^* (and some other unused phage). Notice that q can also be used to produce D_h^* in a similar manner. In dark gray, an alternate way to produce X_i^* and D_j^* is depicted. Here, X_i^+ is first recombined with $D_{j,1}$ to produce $p = c_j i c_j c_j$. Phage p is not in $\mathcal{Q}$, but can be recombined with X_i to produce both X_i^* and D_j^* in one operation.

Proposition 3. *The function f is a bijection.*

Proof. We prove that f is injective, and since there can be at most $n + m$ recombinations in S, we conclude that f is a bijection.

If $i \neq k$, then $f(X_i) \neq f(X_k)$ since equality implies a recombination where at least one parent has an element of $\{-, \circ\}$ in 2nd position, and both children have an element of $[1..n]$.

If $g \neq h$, then $f(D_g) \neq f(D_h)$ since equality implies a recombination where at least one parent has a '$-$' in 2nd position, and both children do not.

Finally, $f(X_i) \neq f(D_j)$ since equality implies a recombination where one parent has a '$-$' in 2nd position, and both children do not. □

The crucial consequences of Proposition 3 are the three following results:

Proposition 4. *There is exactly one element of $\mathcal{X} \cup \mathcal{D}$ in each recombination of S.*

Proof. Since all phages of $\mathcal{X} \cup \mathcal{D}$ are necessary to produce $\mathcal{Q}$, then each one must be used by at least one recombination. We show that the image of f contains no recombination between two of these phages.

Suppose X_i and X_k recombine. Then this recombination is not $f(X_i)$ or $f(X_k)$ since both parents have a '$\circ$' in 2nd position, implying that none of the children have an element of $[1..n]$.

Suppose D_g and D_h recombine. Then this recombination is not $f(D_g)$ or $f(D_h)$ because both parents have a '$-$' in 2nd position, implying that both children have a '$-$' in 2nd position.

Suppose X_i and D_j are used in a recombination R. Then R is not $f(X_i)$ because one parent has a '$\circ$' in 2nd position, and the other has a '$-$', implying that none of the children have an element of $[1..n]$. So R must be $f(D_j)$, implying that one of the children p is an ancestor of D_j^*, having $\circ$ in 2nd position and c_j in 4th position. In the remaining paragraph we show that there is no recombination from S that makes the c_j in 4th position of p the ancestor of the c_j in 4th position of D_j^*, contradicting the existence of R.

Consider any subsequent recombination R' that uses p as a parent. Each R' cannot be a $f(X_i)$ since p cannot contribute an element of $[1..n]$ in 2nd position, so it must be $f(D_h)$, for $D_h \in \mathcal{D}$ and $h \neq j$. Note that the other parent q, used in R', must have '$-$' in 2nd position and c_h in 4th position, so the children are then q' with $\circ$ and c_h in 2nd and 4th positions, and p' with '$-$' and c_j in 2nd and 4th positions. But, while p' is the child that contains the c_j that could be ancestral to the 4th position of D_j^*, it cannot be used as a parent in a recombination of $f(\mathcal{X} \cup \mathcal{D})$, since it has '$-$' in 2nd position, and $f(D_j)$ has already been applied. Therefore, p' is not an ancestor of D_j^*. This is true for any such child p' produced by a subsequent recombination, which contradicts that p is an ancestor of D_j^*. This contradict our initial supposition that R is $f(D_j)$. □

Corollary 1. *The recombination that uses X_i produces X_i^*.*

Proof. Note that X_i is the only phage in $\mathcal{X} \cup \mathcal{D}$ that shares any character with X_i^*. Since both parents of any recombination creating X_i^* must share at least one character with X_i^*, the recombination in S that uses X_i is the only one that can create X_i^*. □

Corollary 2. *The recombination that uses D_j produces either D_j^* or $c_j i c_j c_j$.*

Proof. By definition D_j is an ancestor of D_j^*, and by Corollary 4 we know that any descendant of D_j must recombine with an element of $\mathcal{X} \cup \mathcal{D}$. Since no other member of $\mathcal{X} \cup \mathcal{D}$ has c_j in positions 1, 3 or 4, a child of D_j must be an ancestor of D_j^* and have that character c_j in those positions. This child must be either D_j^*, $p = c_j i c_j c_j$, or $q = c_j - c_j c_j$.

A subsequent recombination using child $q = c_j - c_j c_j$ does not exist since it would either recombine with an X_k, but not produce X_k^* in contradiction with Corollary 1, or with a D_h, whose 2nd position is also '$-$'. Therefore, q has no descendant, contradicting that it is an ancestor of D_j^*. □

We now establish that a recombination scenario S of length $n + m$ implies a valid, and satisfiable truth assignment for ϕ.

For an $X_i \in \mathcal{X}$, we say that X_i *chose* X_i^+ if the only recombination that uses X_i is with an X_i^+ or its descendant, and we say that it chose X_i^- if X_i

recombined with X_i^- or its descendant. If X_i chose X_i^+, we set $x_i = true$, and if X_i chose X_i^- we set $x_i = false$. Let us call the resulting assignment α, which we claim is satisfying. We first argue that each x_i is assigned only one value, and then show that each clause is satisfied.

Proposition 5. *A recombination scenario S of length $n + m$ implies a valid, and satisfiable truth assignment for ϕ.*

Proof. We first show that the assignment α is well-defined, *i.e.* each variable x_i is assigned either $true$ or $false$, but not both. Since each X_i chooses at least one of X_i^+ or X_i^-, we know that x_i is assigned $true$ or $false$. Assume now that x_i is assigned both. Then X_i chose both X_i^+ and X_i^-, meaning that X_i recombines with a phage that descends from both X_i^+ and X_i^-. But the existence of this phage requires a recombination between descendants of both X_i^+ and X_i^-, which contradicts Corollary 4.

We now show that each clause is satisfied by α. By Corollary 2, we know that $D_{j,1}$ and $D_{j,2}$ do not recombine, and only one of the two, that we called D_j, appears in S. Since D_j contributes at most two c_j characters to D_j^*, the third c_j can only be a descendant of an X_i^+ or X_i^-, since they are the only other phages in $\mathcal{P}$ that may contain c_j. By construction, this means that the clause C_j is satisfied by the variable x_i being set to the truth assignment implied by the corresponding X_i^+ or X_i^-. □

5 Conclusion

The notion of recombination used in this article is the same as the *two-point crossover* function [8] used for characterizing fitness landscapes for the exploration of genotypes. This two-point crossover has since been studied in a general form as a *k-point crossover* [5]. While, to the best of our knowledge, there is no work directly linking the area of fitness landscape exploration to the minimization problem discussed in this article, we hope that these related areas can be fused in the future.

In this paper, we showed that the Minimum Phage Population Reconstruction is NP-Complete in two extreme cases: bounded length, and a single target phage. Although negative, such results may come as a relief, since we can turn our focus to algorithms that, by accounting for biological constraints, could provide drastically reduced search spaces for parsimonious solutions. For example, the use of other measures of evolution, or information about community structure [6] might play a significant role in reducing the complexity of the problem: after all, phages recombine all the time, and thrive doing so.

Acknowledgments

ML was partially supported by Canada NSERC Grant number 2019-05817. AB was partially supported by Canada NSERC Grant number 05729-2014. KMS was partially supported by the grant ANR-20-CE48-0001 from the French National Research Agency (ANR).

References

1. Abbott, S., Fairbanks, D.J.: Experiments on Plant Hybrids by Gregor Mendel. Genetics **204**(2), 407–422 (10 2016). `https://doi.org/10.1534/genetics.116.195198`, `https://doi.org/10.1534/genetics.116.195198`
2. Bergeron, A., Meurs, M.J., Valiquette-Labonté, R., Swenson, K.M.: On the comparison of bacteriophage populations. In: Jin, L., Durand, D. (eds.) Comparative Genomics. pp. 3–20. Springer International Publishing, Cham (2022)
3. Berman, P., Karpinski, M., Scott, A.: Approximation hardness of short symmetric instances of max-3sat. In: Electronic Colloquium on Computational Complexity (ECCC). vol. TR03-049 (2004)
4. Botstein, D.: A theory of modular evolution for bacteriophages. Annals of the New York Academy of Sciences **354**(1), 484–491 (1980). `https://doi.org/10.1111/j.1749-6632.1980.tb27987.x`
5. Changat, M., Narasimha-Shenoi, P.G., Nezhad, F.H., Kovše, M., Mohandas, S., Ramachandran, A., Stadler, P.F.: Transit sets of-point crossover operators. AKCE International Journal of Graphs and Combinatorics **17**(1), 519–533 (2020)
6. Chevallereau, A., Pons, B.J., van Houte, S., Westra, E.R.: Interactions between bacterial and phage communities in natural environments. Nature Reviews Microbiology **20**(1), 49–62 (2022)
7. Chmielewska-Jeznach, M., Bardowski, J.K., Szczepankowska, A.K.: Molecular, physiological and phylogenetic traits of lactococcus 936-type phages from distinct dairy environments. Scientific Reports **8**(1), 12540 (2018). `https://doi.org/10.1038/s41598-018-30371-3`
8. Gitchoff, P., Wagner, G.P.: Recombination induced hypergraphs: a new approach to mutation-recombination isomorphism. Complexity **2**(1), 37–43 (1996)
9. Kahánková, J., Pantuček, R., Goerke, C., Ružičková, V., Holochová, P., Doškar, J.: Multilocus pcr typing strategy for differentiation of *Staphylococcus aureus* siphoviruses reflecting their modular genome structure. Environmental microbiology **12**(9), 2527–2538 (2010)
10. Karp, R.: Reducibility among combinatorial problems. Complexity of Computer Computations pp. 85–103 (1972)
11. Kupczok, A., Neve, H., Huang, K.D., Hoeppner, Marc P Heller, K.J., Franz, C.M.A.P., Dagan, T.: Rates of Mutation and Recombination in Siphoviridae Phage Genome Evolution over Three Decades. Molecular Biology and Evolution **35**(5), 1147–1159 (02 2018). `https://doi.org/10.1093/molbev/msy027`
12. Lavelle, K., Murphy, J., Fitzgerald, B., Lugli, G.A., Zomer, A., Neve, H., Ventura, M., Franz, C.M., Cambillau, C., van Sinderen, D., Mahony, J.: A decade of streptococcus thermophilus phage evolution in an irish dairy plant. Applied and Environmental Microbiology **84**(10) (2018). `https://doi.org/10.1128/AEM.02855-17`
13. Murphy, J., Bottacini, F., Mahony, J., Kelleher, P., Neve, H., Zomer, A., Nauta, A., van Sinderen, D.: Comparative genomics and functional analysis of the 936 group of lactococcal siphoviridae phages. Scientific Reports **6**, 21345 EP – (02 2016)
14. Swenson, K.M., Guertin, P., Deschênes, H., Bergeron, A.: Reconstructing the modular recombination history of staphylococcus aureus phages. In: BMC bioinformatics. vol. 14, p. S17. Springer (2013)