Natural Computing (2021) 20:647–658
https://doi.org/10.1007/s11047-021-09867-x

Conjugate word blending: formal model and experimental implementation by XPCR

Francesco Bellamoli[1] · Giuditta Franco[2] · Lila Kari[3] · Silvia Lampis[4] · Timothy Ng[5] · Zihao Wang[3]

Accepted: 21 June 2021 / Published online: 31 August 2021

Abstract
This paper introduces conjugate word blending as a formal model of molecular processes that occur during a DNA experimental protocol called cross-pairing Polymerase Chain Reaction (XPCR). We analyze this formal word and language operation from a computational viewpoint, by investigating closure properties of four Chomsky language families under it. We also report the molecular biology wet lab experiments based on XPCR amplification of gene sequences, which led to the notion of conjugate word blending.

Keywords Conjugate word blending · Cross-pairing Polymerase Chain Reaction (XPCR) · DNA computing · Gene assembly · Formal language operations · Molecular computing · Word blending · Word operations

Mathematics Subject Classification 68R01 · 92B99

1 Introduction

DNA computing (molecular computing, biomolecular computing) is fundamentally based on the idea that molecular biology processes can be used to perform arithmetic and logic operations on information encoded as DNA strands. Research in DNA computing includes DNA-based wet lab experiments that solve computational problems, as well as theoretical studies of models of DNA-based computations and their properties. Since DNA strings can be abstractly viewed as words over the four-letter alphabet of DNA bases, formal language theory, automata theory, and combinatorics on words (branches of theoretical computer science which study words and languages over arbitrary alphabets) emerged as some of the early formalisms for the modelling and study of DNA-based information and computation, (Head 1987). Examples of such models and studies abound, and include splicing systems, DNA words and encodings, sticker systems, Watson-Crick automata, hairpin finite automata, bond-free DNA languages, cellular computing, etc., see

This research was partially supported by NSERC (Natural Sciences and Engineering Research Council of Canada) Discovery Grant R2824A01 and Univ. of Waterloo Transition Grant to L.K., and by FUR (Single Fund for Research), Italian Ministry of Education, Universities, and Research (MIUR) to G.F.

✉ Lila Kari
lila.kari@uwaterloo.ca

✉ Silvia Lampis
silvia.lampis@univr.it

Francesco Bellamoli
francesco.bellamoli@gmail.com

Giuditta Franco
giuditta.franco@univr.it

Timothy Ng
timng@uchicago.edu

Zihao Wang
z465wang@uwaterloo.ca

[1] SOLE-LAB, Department of Biotechnology, University of Verona, Verona, Italy

[2] Department of Computer Science, University of Verona, Verona, Italy

[3] School of Computer Science, University of Waterloo, Waterloo, Canada

[4] Department of Biotechnology, University of Verona, Verona, Italy

[5] Department of Computer Science, University of Chicago, Chicago, USA

reviews (Păun et al. 1998; Amos 2005; Ignatova et al. 2008; Kari et al. 2012). This paper straddles both theoretical and experimental research by reporting the results of an experimental DNA wet lab protocol, called cross-pairing Polymerase Chain Reaction (XPCR), in a specific set-up, as well as modelling this molecular process as a formal language operation and investigating some of its properties.

XPCR is a wet lab procedure introduced in (Franco et al. 2005) for extracting all the strands containing a given pattern (a substring) from a heterogeneous pool of DNA strands. It was employed to implement several DNA recombination algorithms (Franco et al. 2006), for the creation of the solution space for a SAT problem (Franco 2005), and for mutagenesis (Franco and Manca 2011). The combinatorial power of this technique was explained by logical-symbolic schemes in (Manca and Franco 2008), while algorithms to create combinatorial libraries were experimented in (Franco and Manca 2011), and improved in (Franco et al. 2017), where all permutations of three genes were generated. Relevant to this paper, XPCR has been successfully used to concatenate two different genes, provided they are flanked by compatible primers (Bellamoli 2013). More specifically, if A and D are two strings (representing two genes) over the DNA alphabet $\{a, c, g, t\}$, and α, γ and β are strings over the same alphabet (representing the primers), then XPCR combines input strings $\alpha A\gamma$ and $\gamma D\beta$ to produce the output string $\alpha A\gamma D\beta$ (see Fig. 1).

However, it has been recently observed (Franco et al. 2017) that in the specific set-up where the goal is to assemble two copies of the same gene, that is, when $A = D$, the result of XPCR is not as expected. Namely, XPCR-based experiments with inputs $\alpha A\gamma$ and $\gamma A\beta$ repeatedly produced the result $\alpha A\beta$ (instead of the expected outcome $\alpha A\gamma A\beta$). In this paper, we report the results of these experiments, as well as define and investigate some computational properties of a formal language operation called *conjugate word blending*, that models the underlying DNA process.

Note that conjugate word blending is a special case of word blending, which was defined and studied in (Enaganti et al. 2020) as a model inspired by the XPCR. A related formal language operation, called "overlap assembly", introduced in (Csuhaj-Varjú et al. 2007) and investigated in (Brzozowski et al. 2018; Enaganti et al. 2017a, b), models the capability of XPCR to concatenate strings in the following way. An alphabet Σ is a finite non-empty set of symbols. We denote by Σ^* the set of all words (or strings) over Σ, including the empty word λ, and by Σ^+ the set of all non-empty words over Σ. The overlap assembly of two strings $x = uv$ and $y = vw$ that share a non-empty overlap v results in the string uvw, and it is defined by. A particular case of overlap assembly, called the "chop" operation,

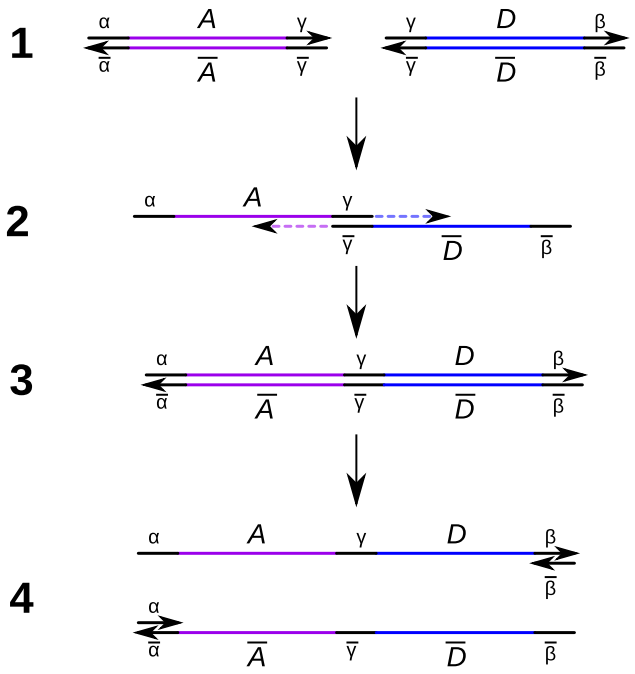

Fig. 1 XPCR technique for concatenation of two different genes, A and D, using the primer pair α and $\overline{\beta}$. The product $\alpha A\gamma D\beta$ is formed by overlap assembly of the two templates ($\alpha A\gamma$ and $\gamma D\beta$) and amplified by polymerase extension (primers α and $\overline{\beta}$). Adapted from Fig. 4 (Franco et al. 2017)

where the overlap is a singleton letter, was studied in (Holzer and Jakobi 2011, 2012), and generalized to an arbitrary-length overlap in (Holzer et al. 2017). Other similar operations have been studied in the literature of formal languages, such as the "short concatenation" (Carausu and Păun 1981), which uses only the maximum-length overlap x between operands (possibly empty), the "Latin product" of words (Golan 1992) where the overlap occurs at the extremities of words and consists of only one letter, the "crossover" operation (Ceterchi 2006) where an overlap consisting of only one letter may occur in the middle of the words, and the operation $\otimes$ which imposes the restriction that at least one of the non-overlapping parts u, w is not empty (Ito and Lischke 2007). Overlap assembly can also be considered a particular case of "semantic shuffle on trajectories", with trajectory $0^*\Sigma^+1^*$, or as a generalization of the operation $\odot_N$ from (Domaratzki 2009) which imposes the length of the overlap to be at least N. Lastly, biological phenomena based on overlap assembly can also be modelled by using splicing systems (Bonizzoni et al. 2005; Goode and Pixton 2007; Pixton 2000; Păun 1996).

This paper, which focuses on the specific set-up for XPCR described above in this introduction, is organized as

follows. Section 2 introduces the binary word/language operation called conjugate word blending, and investigates closure properties of the Chomsky families languages under it. Section 3 reports the molecular computing wet lab experiments, and the specific XPCR set-up that led to the notion of conjugate word blending. Section 4 contains the biotechnological details of the experiments. Section 5 concludes the paper.

2 Conjugate word blending

A language over an alphabet Σ is a set of words $L \subseteq \Sigma^*$. We say that a language is *regular* if it can be recognized by a finite state automaton. A *nondeterministic finite automaton* (NFA) is a 5-tuple $A = (Q, \Sigma, \delta, s, F)$ where Q is a finite set of states, Σ is an alphabet, δ is a function $\delta : Q \times \Sigma \to 2^Q$, $s \in Q$ is the initial state, and $F \subseteq Q$ is a set of final states.

Given two words x, y over an alphabet Σ, the *word blending* of x with y was defined in (Enaganti et al. 2020) as

$$x \underline{\bowtie} y = \{\alpha w \beta \mid x = \alpha w \gamma_1, y = \gamma_2 w \beta, \alpha, \beta, \gamma_1, \gamma_2 \in \Sigma^*, w \in \Sigma^+\}.$$

Word blending allows γ_1 and γ_2 to be different strings. As a step towards a formal model that is closer to the XPCR process experimentally-observed in (Franco et al. 2017), we now require that $\gamma_1 = \gamma_2 \neq \lambda$ and define the *conjugate word blending* of two words as follows.

Definition 1 Given two words x and y over an alphabet Σ, the *conjugate word blending* of x with y is defined as

$$x \bowtie y = \{\alpha w \beta \mid x = \alpha w \gamma, y = \gamma w \beta, \alpha, \beta \in \Sigma^*, \gamma, w \in \Sigma^+\}.$$

The term "conjugate word blending" alludes to the fact that the common segments of the operands, $w\gamma$ and γw, are conjugate words (a word u is a conjugate of a word v if u can be obtained from v by cyclically shifting its letters (Rozenberg and Salomaa 1997)). We can extend this word operation to languages in the natural way:

$$L_1 \bowtie L_2 = \bigcup_{x \in L_1, y \in L_2} (x \bowtie y).$$

As an example, if $u = 1010101$ and $v = 10101$, then $u \bowtie v = \{10101\underline{0}101,\ 101\underline{0}101,\ 101\underline{0}1\}$ (the underlined subwords are the corresponding overlaps w). If Σ is an alphabet, and $a \in \Sigma$, then $\Sigma^* \bowtie \Sigma^* = \Sigma^+$, $\Sigma^* \bowtie \{aa\} = \Sigma^* a$, and $\{aa\} \bowtie \Sigma^* = a\Sigma^*$. As this example shows, the conjugate word blending operation is not commutative.

As with the original version of word blending (Enaganti et al. 2020), we can express the conjugate word blending operation as a splicing scheme. Recall that splicing schemes (Păun 1996) are defined via sets of quadruples (called splicing rules) $r = u_1 \# u_2 \$ u_3 \# u_4$ with $\#, \$ \notin \Sigma$, and $u_1, u_2, u_3, u_4 \in \Sigma^*$. For two strings $x = x_1 u_1 u_2 x_2$ and $y = y_1 u_3 u_4 y_2$, applying the rule $r = u_1 \# u_2 \$ u_3 \# u_4$ produces a string $z = x_1 u_1 u_4 y_2$, and this is denoted by $(x, y) \vdash^r z$. A splicing scheme is a pair $\sigma = (\Sigma, \mathcal{R})$ where Σ is an alphabet and $\mathcal{R}$ is a set of splicing rules. A splicing system is a splicing scheme together with an initial language, and the language generated by a splicing system is the set of all words obtained starting from the initial language by iteratively applying splicing rules. The connection between splicing and word blending was shown in (Enaganti et al. 2020), where it was proved that word blending $\underline{\bowtie}$ can be expressed as one step of the splicing scheme consisting of the rules

$$R_{\underline{\bowtie}} = \{a \# \lambda \$ a \# \lambda \mid a \in \Sigma\}.$$

In a splicing rule $u_1 \# u_2 \$ u_3 \# u_4$, the words u_1 and u_4 are called visible sites, while u_2 and u_3 are called invisible sites. A splicing scheme $\sigma = (\Sigma, \mathcal{R})$ is said to be regular if $\mathcal{R}$ is a regular language contained in $\Sigma^* \# \Sigma^* \$ \Sigma^* \# \Sigma^*$. In (Pixton 2000), it is shown that the families of regular, context-free, and recursively enumerable languages are closed under regular splicing systems with finitely many visible sites.

It is not difficult to see that the conjugate word blending operation cannot be expressed in the same way. Conjugate word blending can be thought of as a single step of a splicing scheme with the following set of rules

$$R_{\bowtie} = \{w \# \gamma \$ \gamma w \# \lambda \mid w, \gamma \in \Sigma^+\}.$$

However, observe that this splicing scheme is not regular. In fact, it is context-sensitive. If we apply a morphism φ that erases $\#$ and \$, we get

$$\varphi(R_{\bowtie}) = \{w\gamma\gamma w \mid w, \gamma \in \Sigma^+\},$$

which is not context-free. This suggests that the closure properties of the conjugate word blending operation may be different from the original version of word blending operation.

We will now investigate closure properties of the main Chomsky families of languages under conjugate word blending. We first show that we can construct a nondeterministic finite automaton that recognizes the conjugate word blending of two regular languages.

Proposition 1 *Given two NFAs A and B, we can effectively construct an NFA C such that* $L(C) = L(A) \bowtie L(B)$.

Proof Consider states $p \in Q_A$ and $q \in Q_B$ and NFAs $A = (Q_A, \Sigma, \delta_A, s_A, F_A)$ and $B = (Q_B, \Sigma, \delta_B, s_B, F_B)$. We define the following two languages:

$$L(A_p) = \{w \in \Sigma^+ \mid \delta_A(p, w) \cap F_A \neq \emptyset\},$$
$$L({}_qB) = \{w \in \Sigma^+ \mid q \in \delta_B(s_B, w)\}.$$

We can now construct the NFA $C = (Q', \Sigma, \delta', s', F')$ that accepts exactly the language $L(A) \bowtie L(B)$, as follows. The sets of states of the NFA C is $Q' = Q_A \cup (Q_A \times Q_B \times Q_B) \cup Q_B$, the initial state is $s' = s_A$, and the set of final states is $F' = F_B$. The transition function $\delta' : Q' \times \Sigma \cup \{\lambda\} \to 2^{Q'}$ is constructed as follows:

1. $\delta'(p, a) = \delta_A(p, a) \cup \{\langle p', q', r'\rangle \mid p' \in \delta_A(p, a),\ q' \in \delta_B(r', a),\ r' \in Q_B\}$ for all $p \in Q_A$, and $a \in \Sigma$;
2. $\delta'(\langle p, q, r\rangle, a) = \{\langle p', q', r\rangle \mid p' \in \delta_A(p, a),\ q' \in \delta_B(q, a)\}$ for all $p \in Q_A$, $q, r \in Q_B$ and $a \in \Sigma$;
3. $\delta'(\langle p, q, r\rangle, \lambda) = \{q\}$ for all $p \in Q_A$ and $q, r \in Q_B$ for which $L(A_p) \cap L({}_rB) \cap \Sigma^+ \neq \emptyset$;
4. $\delta'(q, a) = \delta_B(q, a)$ for all $q \in Q_B$ and $a \in \Sigma$.

The idea behind this construction is that for a word $\alpha w \beta \in \alpha w \gamma \bowtie \gamma w \beta$, the states in Q_A are used for the derivation of α, the states in Q_B are used for the derivation of β, and the states in $Q_A \times Q_B \times Q_B$ are used for the derivation of w, as follows. If the NFA C is in a state from Q_A and reads a letter, it nondeterministically decides the letter is in α or the letter is the first letter of w by transitions of type 1. If the state $\langle p, q, r\rangle$ is reached after a transition of type 1, we assume that the state of B after reading γ is r, the state of A (respectively B) after reading the first letter of w is p (respectively q). Transitions of type 2 simulate the simultaneous processing, by both A and B, of letters from w. By transitions of type 3, the NFA C checks if the non-empty subword γ exists and, if it does, it continues with the derivation of β according to transitions of type 4.

Let us now prove that $L(A) \bowtie L(B) \subseteq L(C)$. Consider a word $z \in L(A) \bowtie L(B)$ with $z = \alpha w \beta$ where $x = \alpha w \gamma \in L(A)$, $y = \gamma w \beta \in L(B)$, and $\gamma \in \Sigma^+$. Now, write $w = aw'$ for $a \in \Sigma$ and $w' \in \Sigma^*$. Since $x \in L(A)$, there must be a path in A

$$s_A \xrightarrow{\alpha} p_1 \xrightarrow{a} p_2 \xrightarrow{w'} p_3 \xrightarrow{\gamma} p_4, \text{where } p_4 \in F_A.$$

Similarly, since $y \in L(B)$, there must be a path in B

$$s_B \xrightarrow{\gamma} q_1 \xrightarrow{a} q_2 \xrightarrow{w'} q_3 \xrightarrow{\beta} q_4, \text{where } q_4 \in F_B.$$

From this, we will show that there exists an accepting computation path for z in C,

$$s' = s_A \xrightarrow{\alpha} p_1 \xrightarrow{a} \langle p_2, q_2, q_1\rangle \xrightarrow{w'} \langle p_3, q_3, q_1\rangle \xrightarrow{\lambda} q_3 \xrightarrow{\beta} q_4,$$

where $q_4 \in F_B = F'$.

More precisely, we observe that at the beginning of the blending on a, we have $\langle p_2, q_2, q_1\rangle \in \delta'(p_1, a)$ since $q_2 \in \delta_B(q_1, a)$. Since $p_4 \in \delta_A(p_3, \gamma)$, $p_4 \in F_A$, $q_1 \in \delta_B(s_B, \gamma)$ and $\gamma \in \Sigma^+$, we have $\gamma \in L(A_{p_3})$ and $\gamma \in L({}_{q_1}B)$, so we have $\gamma \in L(A_{p_3}) \cap L({}_{q_1}B) \cap \Sigma^+$. Therefore, $q_3 \in \delta'(\langle p_3, q_3, q_1\rangle, \lambda)$. Thus, we have shown that $z \in L(C)$, and consequently $L(A) \bowtie L(B) \subseteq L(C)$.

Now we will show that $L(C) \subseteq L(A) \bowtie L(B)$. Let $z \in L(C)$. Then there exists a path on z in C from s_A to a state in F_B. Recall that there are three types of states in C: states of A, Q_A; triples of states $\langle p, q, r\rangle$, $p \in Q_A$, $q, r \in Q_B$; states of B, Q_B. The definition of C implies that any accepting computation of a word w in C must contain all three types of states, in this order. Then we can consider an accepting path for $z = \alpha a w' \beta$, where $\alpha, \beta, w' \in \Sigma^*$ and $a \in \Sigma$, by

$$s' = s_A \xrightarrow{\alpha} p_1 \xrightarrow{a} \langle p_2, q_2, q_1\rangle \xrightarrow{w'} \langle p_3, q_3, q_1\rangle \xrightarrow{\lambda} q_3 \xrightarrow{\beta} q_4 \in F_B = F'.$$

In this path, p_1 is the last state of A that occurs, $\langle p_2, q_2, q_1\rangle$ is the first triple that occurs, $\langle p_3, q_3, q_1\rangle$ is the final triple that occurs, q_3 is the first state of B that occurs, and q_4 is an accepting state of C, which by definition is an accepting state of B.

From the definition of the transition function, it is clear that the words α, αa, and $\alpha a w'$ are all prefixes of a word in $L(A)$ and the words β and $w'\beta$ are all suffixes of a word in $L(B)$.

The final observation is that we need to consider transitions from $\langle p_3, q_3, q_1\rangle$ to q_3 on the empty word λ. Such a transition can only occur if there exists a non-empty word $\gamma \in L(A_{p_3}) \cap L({}_{q_1}B) \cap \Sigma^+$. From this, we have $\alpha a w' \gamma \in L(A)$ since $p_3 \in \delta_A(s_A, \alpha a w')$ and $\delta_A(p_3, \gamma) \cap F_A \neq \emptyset$ by definition of $L(A_{p_3})$. We also have $\gamma a w' \beta \in L(B)$ since by definition of $L({}_{q_1}B)$, we have $q_1 \in \delta_B(s_B, \gamma)$. By the definition of δ', we have $\langle p_2, q_2, q_1\rangle \in \delta'(p_1, a)$ if $q_2 \in \delta_B(q_1, a)$. Therefore, there is a path in B

$$s_B \xrightarrow{\gamma} q_1 \xrightarrow{a} q_2 \xrightarrow{w'} q_3 \xrightarrow{\beta} q_4 \in F_B.$$

Taking $w = aw'$, we can now write $z = \alpha w \beta \in L(C)$, with $x = \alpha w \gamma \in L(A)$, and $y = \gamma w \beta \in L(B)$. By the definition of conjugate word blending, this implies $z \in x \bowtie y$ and thus $z \in L(A) \bowtie L(B)$.

Therefore $L(C) \subseteq L(A) \bowtie L(B)$, and we can conclude that $L(C) = L(A) \bowtie L(B)$. □

Corollary 1 *The class of regular languages is closed under conjugate word blending.*

Next, we will show that unlike word blending, the family of context-free languages is not closed under conjugate word blending.

Proposition 2 *The class of context-free languages is not closed under conjugate word blending.*

Proof This can be proved by a counterexample. Consider two context-free languages $L_1 = \{a^n\$b^n\$\# \mid n \in \mathbb{N}\}$ and $L_2 = \{\#\$b^m\$a^m \mid m \in \mathbb{N}\}$, we have that $(L_1 \bowtie L_2) \cap a^*\$b^*\$a^* = \{a^n\$b^n\$a^n \mid n \in \mathbb{N}\}$, which is not context-free. Thus, since the class of context-free languages is closed under intersection with regular languages, it is not closed under conjugate word blending. □

Proposition 3 *The class of context-sensitive languages is not closed under conjugate word blending.*

Proof Assume that the class of context-sensitive languages is closed under conjugate word blending. Let L_0 be a recursively enumerable but not context-sensitive language over an alphabet Σ, and let $a, b \notin \Sigma$. Then, there is a context-sensitive language L such that L consists of words of the form wba^i where $i \geq 0$ and $w \in L_0$, and for every word w in L_0 there is an $i \geq 0$ such that $wba^i \in L$. We have that $(La \bowtie a^+b) \cap \Sigma^*b = L_0b$. If the class of context-sensitive languages were closed under conjugate word blending, then L_0b would be context-sensitive—a contradiction. □

We will now show that the class of recursively enumerable languages is closed under conjugate word blending. Recall that *sequential deletion* was defined in (Kari 1991) as the binary language operation $L_1 \rightarrow L_2 = \bigcup_{u \in L_1, v \in L_2}(u \rightarrow v)$, where $u \rightarrow v = \{w \in \Sigma^* \mid u = w_1vw_2, w = w_1w_2, w_1, w_2 \in \Sigma\}$. We begin with the following lemma.

Lemma 1 *Consider two languages L_1, L_2 over an alphabet Σ, two symbols $\#, \$ \notin \Sigma$, and a homomorphism $h(a) = a$, for $a \in \Sigma$, and $h(\#) = h(\$) = \lambda$. Conjugate word blending can be expressed as*

$$L_1 \bowtie L_2 = (L \cap L') \rightarrow (\#\Sigma^+\$\Sigma^+\$),$$

where

$$L = (h^{-1}(L_1) \cap (\Sigma^*\Sigma^+\#\Sigma^+\$)) \overline{\odot} \left(h^{-1}(L_2) \cap (\#\Sigma^+\$\Sigma^+\$\Sigma^*)\right),$$

$\rightarrow$ is the sequential deletion operation, $\overline{\odot}$ is the overlap assembly operation, and $L' = \bigcup_{w \in \Sigma^*} \Sigma^*w\#\Sigma^+\$w\$\Sigma^*$.

Proof Consider a word $z \in L_1 \bowtie L_2$, and there exists a decomposition $z = \alpha w\beta$ where $x = \alpha w\gamma \in L_1$, $y = \gamma w\beta \in L_2$, and $\gamma \in \Sigma^+$. We have that

$$\begin{aligned} z &= \alpha w\beta \\ &\in \alpha w\#\gamma\$w\$\beta \rightarrow (\#\Sigma^+\$\Sigma^+\$) \\ &= (\alpha w\#\gamma\$w\$\beta \cap L') \rightarrow (\#\Sigma^+\$\Sigma^+\$) \\ &\subseteq (L \cap L') \rightarrow (\#\Sigma^+\$\Sigma^+\$). \end{aligned}$$

Next, consider a word $z \in (L \cap L') \rightarrow (\#\Sigma^+\$\Sigma^+\$)$. There exist words $\alpha, \beta \in \Sigma^*$ and $w, \gamma \in \Sigma^+$ such that $z = \alpha w\beta$ and $z' = \alpha w\#\gamma\$w\$\beta \in (L \cap L') \subseteq L$. Thus, there exist words $x' = \alpha w\#\gamma\$ \in (h^{-1}(L_1) \cap (\Sigma^*\Sigma^+\#\Sigma^+\$))$ and $y' = \#\gamma\$w\$\beta \in (h^{-1}(L_2) \cap (\#\Sigma^+\$\ \ \Sigma^+\$\Sigma^*))$ such that , $x = \alpha w\gamma \in L_1$ and $y = \gamma w\beta \in L_2$. Thus, we have that $z \in L_1 \bowtie L_2$. □

Proposition 4 *The class of recursively enumerable languages is closed under conjugate word blending.*

Proof This follows from Lemma 1, since the class of recursively enumerable languages is closed under overlap assembly (Enaganti et al. 2017b), inverse homomorphism and intersection (Salomaa 1973), as well as sequential deletion (Kari 1991). □

In summary, the results of this section show that the classes of regular and recursively enumerable languages are closed under conjugate word blending, while the classes of context-free and context-sensitive languages are not. As conjectured earlier, these closure properties are different from those of word blending, the difference being that the class of context-free languages is closed under word blending, but not under conjugate word blending.

3 DNA implementation of conjugate word blending

In this section we describe the wet lab experiments that motivate and implement the conjugate word blending operation. Section 3.1 introduces some basic notions of molecular biology. Section 3.2 outlines the initial experimental evidence that led to the definition of conjugate word blending operation. Section 3.3 reports the experiments which confirmed and validated the XPCR-based implementation of the conjugate word blending. Note that some preliminary work for these experiments was developed in (Bellamoli 2013; Franco et al. 2017) with the aim of generating a DNA library of operons (i.e., permutations of genes) able to optimize the PAH degradation work of *Burkholderia fungorum DBT1*. The wet lab experiments reported in this paper involve three different genes: *dbtAa*

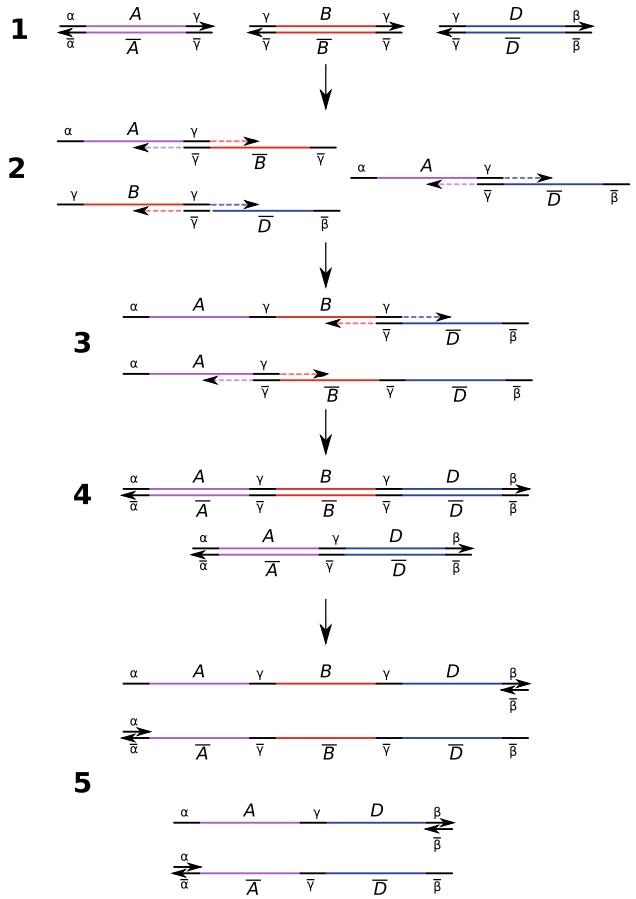

Fig. 2 XPCR-based two-gene concatenation (genes *A* and *D*), from input template $\alpha A\gamma + \gamma B\gamma + \gamma D\beta$ and primers α and $\overline{\beta}$. In step 3, only sequences which are exponentially amplified are illustrated. They were formed by the (iteration of) binding of W/C complementary sequences and polymerase extension. The amplification of the longer formation $\alpha A\gamma B\gamma D\beta$ was produced in an insignificant quantity, as illustrated in Fig. 3. Adapted from Fig. 8 (Franco et al. 2017)

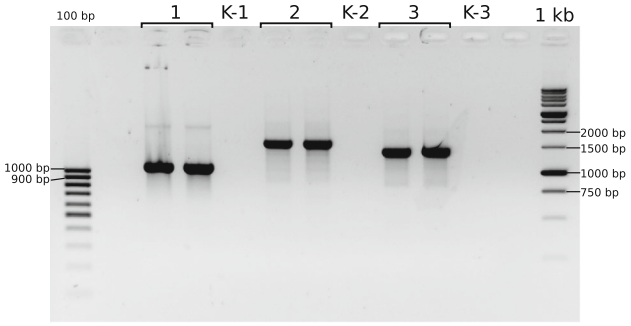

Fig. 3 XPCR with templates containing the same gene, respectively containing different genes. Amplifications with primers (α, $\overline{\beta}$) and *Taq* polymerase. Lane 1: templates $\alpha A\gamma + \gamma A\gamma + \gamma A\beta$ (for details, see 5.1, Table 1) exhibit a main product of about 1000 bp ($\alpha A\beta$, dark band) and a secondary product of about 2000 bp ($\alpha A\gamma A\beta$, faint grey band, in the same lane). Lane 2: Input templates $\alpha A\gamma + \gamma B\gamma + \gamma D\beta$ (see 1.1, Table 1) exhibit an amplification product of about 1600bp which corresponds to the output $\alpha A\gamma D\beta$. Lane 3: Input templates $\alpha B\gamma + \gamma D\gamma + \gamma A\beta$ (see 1.2, Table 1), with output amplification product 1400 bp long, corresponding to $\alpha B\gamma A\beta$. Lanes K-1, K-2, K-3: respective negative controls, without templates. Picture from Fig 4.14 of (Bellamoli 2013)

(Ferredoxin Reductase, 1,019 bp)[1], *dbtAb* (Ferredoxin, 311 bp), and *dbtAd* (β Dioxygenase subunit, 518 bp), extracted from the widely studied bacterial strain *Burkholderia fungorum* DBT1 (Di Gregorio et al. 2004). In the following, we will denote these three genes by the capital letters *A*, *B*, and *D* respectively, and the primers (21 bp long DNA sequences used in XPCR, as detailed in the next section) by α, β, and γ.

3.1 Molecular biology preliminaries

Recall that a DNA single strand consists of four different types of units called *bases*, strung together by an oriented backbone. The distinct ends of a DNA single strand are called the 5′-end and the 3′-end respectively, with the 3′-end pictorially denoted by an arrow tip, as in Figs. 1, 2 and 4. The bases are Adenine (*a*), Guanine (*g*), Cytosine (*c*) and Thymine (*t*), and *a* can chemically bind to an opposing *t* being on another single strand, while *c* can similarly bind to *g*. Bases that can thus bind are called Watson-Crick (W/C) complementary, and two DNA single strands with opposite orientation (that is, opposite 3′ ends) and with W/C complementary bases at each position can bind to each other to form a DNA double strand in a process called base-pairing or *annealing* (achieved by decreasing the temperature)[2]. By convention, a string *w* over the alphabet $\Delta = \{a, c, g, t\}$ represents a DNA single strand in the 5′ to 3′ direction. In the remainder of this paper we will denote the W/C complement of a string *w* by $\overline{w}$, and the union of the sets *U* and *V* by $U + V$. Often we will use as synonyms the terms *strand, word, string, fragment, molecule*. The term *amplicon* is used to denote a fragment of DNA which is the product of molecular amplification (i.e., replication).

Amplification experiments involve the *DNA Polymerase* enzyme, the activity of which presupposes the existence of a DNA single strand called template, and of a second short DNA strand called primer, that is W/C complementary to a portion of the template and binds to it. Given a supply of individual bases, the DNA polymerase enzyme extends the 3′-end of the bound primer by adding individual bases

[1] The length of a DNA double strand is measured in basepairs (bp), whereby 1 bp is a unit consisting of one base on a DNA strand together with its corresponding complementary base on the opposite strand.

[2] The opposite process, that of a DNA double strand breaking apart into its constituent single strands, is called *melting* or *denaturation* (achieved by increasing the temperature).

complementary to the template bases, one by one, until the end of the template is reached. The newly formed DNA strand is a strand that starts with the primer and is W/C complementary to the rest of the template. An iterated version of this process is used to obtain an exponential amplification of DNA strands, in a standard protocol called Polymerase Chain Reaction (PCR). Namely, if a couple of primers α and $\bar{\beta}$ is used in conjunction with a double-stranded template $x\alpha y\beta z$, then the result of PCR is the formation (and exponential amplification) of the new DNA molecule $\alpha y\beta$—the substring of the template flanked by the primers. When used in non-standard ways, PCR can produce a combinatorial richness of molecules, but it may also behave in ways which are complex and difficult to control (Hommelsheim et al. 2015; Kalle et al. 2014; Manca and Franco 2008).

XPCR is a PCR-based protocol which realizes what in the context of splicing systems is called a *null-context splicing rule* (Head 1987), which is a particular splicing rule $u_1\#u_2\$u_3\#u_4$ having $u_2u_4 = \lambda$ and $u_1 = u_3$. In its general form, XPCR takes as input sequences $\alpha X_1\gamma Y_2\beta + \alpha Y_1\gamma X_2\beta$, where X_1, Y_1, Y_2, X_2 are genes, and α, β and γ are primer sequences, and produces as an output the chimeric sequences[3] $\alpha X_1\gamma X_2\beta$ and $\alpha Y_1\gamma Y_2\beta$ - this corresponds to the application of a null-context splicing rule with $u_1 = u_3 = \gamma$. The essential feature of this process (e.g., the recombination between $\alpha X_1\gamma$ and $\gamma X_2\beta$ that produces $\alpha X_1\gamma X_2\beta$) can also be formalized as the overlap assembly operation between two strings xy and yz, resulting in the string xyz. Figure 1 illustrates the overlap assembly between xy and yz where $x = \alpha A$, $y = \gamma$, and $z = D\beta$. If A and D are the genes introduced in this section, then the expected length of the chimeric amplicon $\alpha A\gamma D\beta$ is 1600 bp, due to the primer and gene length.

All experiments of DNA strand amplification were performed in double sampling (that is, on two test tubes in parallel), with negative controls (test tubes with the same contents, except without any DNA templates), under different experimental conditions (including temperature, concentration, gene and length variations), and repeated with two different polymerase enzymes, *Taq* polymerase and *Pfu* polymerase. To ensure higher duplication fidelity, *Pfu* DNA polymerase was chosen over the routinely used *Taq* DNA polymerase for initial reactions (gene extraction from the original genome), due to its proofreading capabilities and thermal resistance. More technical biotechnological details are reported in Sect. 4.

[3] A chimeric sequence is a sequence formed from the prefix of one sequence and the suffix of another sequence joined together.

3.2 The initial experimental evidence

Concatenation of two (different) genes by XPCR was successfully implemented, even starting from three different genes as templates, as illustrated in Fig. 2, where a third input template $\gamma B\gamma$ was added (to favour the formation of additional longer molecules $\alpha A\gamma B\gamma D\beta$), apt to perturb the expected two-genes amplification. This was a way to prove the stability and robustness of XPCR, that is, its reliability under perturbation. In fact, also in some experiments reported in next subsection, an *interference molecule* $\gamma X\gamma$ was added (with $X = A, B, D$), at higher concentrations than the other input molecules, to see whether it would interfere with the amplification of molecules containing γ as prefix or suffix by forming longer concatenation (of three genes, as in Fig. 2).

XPCR did not behave as expected when attempts were made to concatenate copies of the same gene using the method illustrated in Fig. 2. In (Bellamoli 2013) several experiments were carried out with the aim of concatenating two (or more) copies of the same gene, using primer pairs (α, $\bar{\beta}$), and templates $\alpha X\gamma + \gamma X\beta$ (or templates $\alpha X\gamma + \gamma X\gamma + \gamma X\beta$). The output of these experiments was, unexpectedly, $\alpha X\beta$, rather than $\alpha X\gamma X\beta$ (respectively $\alpha X\gamma X\gamma X\beta$). These results were observed in presence of the interference molecules $\gamma X\gamma$ at different concentration ratios.

As exemplification of these phenomena, in Fig. 3 we report experimental results exhibiting as outputs both the two-gene concatenation described in Fig. 2 (in the presence of a long interference molecule containing a different gene), and the unexpected amplicon $\alpha A\beta$, when copies of the gene A were present in the templates. More precisely, amplification of an input composed of three different templates, $\alpha A\gamma + \gamma B\gamma + \gamma D\beta$ (respectively $\alpha B\gamma + \gamma D\gamma + \gamma A\beta$) produced as an output $\alpha A\gamma D\beta$ (respectively $\alpha B\gamma A\beta$), as seen in Lane 2 (respectively Lane 3) of Fig. 3. On the other hand, amplification of an input composed of three different templates all containing the gene A, that is, $\alpha A\gamma + \gamma A\gamma + \gamma A\beta$, produced as an output only the sequence $\alpha A\beta$, as seen in Lane 1 of Fig. 3. In all cases we amplified three different templates, present in *equal* concentrations, by PCR reactions under identical experimental conditions, with basic *Taq* polymerase.

These results provided experimental evidence of a limitation of the XPCR protocol, which indeed cannot be used to concatenate multiple occurrences of the same gene in a significant quantity (Franco et al. 2017).

In case of multiple occurrences of the same gene, the unexpected outcome of XPCR might be due to phenomena similar to those observed in (Hommelsheim et al. 2015), that alter the normal amplification of DNA strands sharing long fragments. In particular, when we perform PCR with

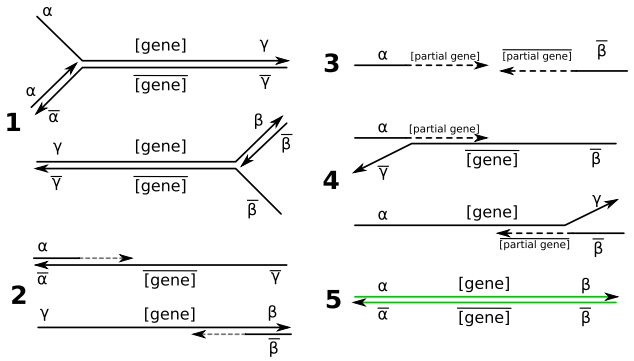

Fig. 4 A possible explanation for the formation of the conjugate blending operation output. (The implicit assumption is that there always exists one template—out of millions—for which the described premature detachment occurs, and that this is enough to generate an exponential amplification of $\alpha X\beta$, with gene X and primers α and $\overline{\beta}$, in next PCR cycles.) **(1)** Both primers anneal. **(2)** Primer polymerase extension occurs along single templates. Over long segments X this process takes long time, and it may be interrupted by high denaturation temperature expected in next step of PCR. This causes a premature detachment of the polymerase enzyme, and then the generation of incomplete template copies, visible in **(3)**. In the next PCR cycle, the resulting incomplete strands may anneal to each other and also to the other template **(4)**, then generating (by polymerase extension) single strands $\alpha X\beta$ and $\overline{\alpha X\beta}$ which will work as templates in step **(5)** where they will be exponentially amplified due to primer annealing. The single strands containing γ or $\overline{\gamma}$ do not anneal with any of the primers and are not amplified

primers $(\alpha, \overline{\beta})$ on sequences with a common substring X, such as $\alpha X\gamma$ and $\gamma X\beta$, it results in the biased production of the shortest amplicon $\alpha X\beta$, as depicted in Fig. 4, up to the point where longer fragments are not detectable. On the electrophoresis gel, this leads to faint or indistinguishable bands for the longer products, and a strong signal for the short product. In other words, once the shortest sequence $\alpha X\beta$ has been formed, it is amplified faster than the longer strand $\alpha X\gamma X\beta$, probably due to the higher annealing efficiency of primers on shorter sequences.

3.3 Conjugate word blending: experimental results

In this section, we report the details of additional wet lab DNA experiments that motivated and validated the notion of conjugate word blending explored in this paper. Below is a summary of all PCR experiments that demonstrate the conjugate word blending operation in action. The primers used are $(\alpha, \overline{\beta})$, and, as detailed below, these experiments confirm the amplified production of $\alpha X\beta$ sequences. Note that, based on the length of primers α, γ, β (21bp), and genes A (1,019bp), B (311 bp), and D (518 bp), the expected length of the amplicons $\alpha X\beta$ is 1061 bp (for $X = A$), 353 bp (for $X = B$), and 560 bp (for $X = D$).

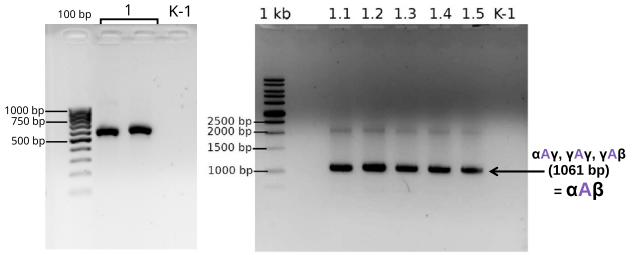

Fig. 5 XPCR implementation of conjugate word blending over gene **D** with no interference (left panel), and over gene **A**, with high interference (right panel). Reactions with primers $(\alpha, \overline{\beta})$ and *Pfu* polymerase. Left panel: Lane 1 - templates $\alpha D\gamma + \gamma D\beta$, exhibited product of about 500 bp ($\alpha D\beta$) (see 9, Table 1 for details). Lane K-1: negative control without templates for all reactions. Right Panel: Lanes 1.1, 1.2, 1.3, 1,4, 1.5 - templates $\alpha A\gamma + \gamma A\gamma + \gamma A\beta$, with concentration ratio 1:10:1, and different annealing temperatures (see 6, Tables 1 and 3). All lanes exhibit a main product of about 1000 bp ($\alpha A\beta$) and a faint band of about 2000 bp ($\alpha A\gamma A\beta$). Picture from Fig. 6 of (Franco et al. 2017)

1. Input composed of two different templates containing gene D, namely $\alpha D\gamma + \gamma D\beta$. The output is $\alpha D\beta$. The reaction, performed with *Pfu* polymerase, has the output amplicon reported in Lane 1 on the left panel of Fig. 5.
2. Input composed of three different templates containing gene D, namely, $\alpha D\gamma + \gamma D\gamma + \gamma D\beta$. The string $\gamma D\gamma$ is the interference molecule. The template concentration ratio was 1:10:1 (concentration of $\alpha D\gamma$, vs. that of $\gamma D\gamma$, vs. that of $\gamma D\beta$) to favour the amplification of longer amplicons. The output was $\alpha D\beta$. Reactions were carried out with *Pfu* polymerase, and five different annealing temperatures, corresponding to the amplicons visible in the lanes 1.1, 1.2, 1.3, 1.4, and 1.5 of right panel of Fig. 6.
3. Input composed of three different templates containing gene B, namely $\alpha B\gamma + \gamma B\gamma + \gamma B\beta$ with concentration ratio 1:2:1. The output was $\alpha B\beta$. Reactions were carried out with *Taq* polymerase, and corresponding products are visible in left gel lane of Fig. 6.
4. Input composed of three different templates containing gene A, namely $\alpha A\gamma + \gamma A\gamma + \gamma A\beta$. The output was $\alpha A\beta$. Reactions were carried out with *Pfu* polymerase, and with different concentrations for the interference molecule $\gamma A\gamma$ with respect to the other two templates $\alpha A\gamma$ and $\gamma A\beta$. Template concentration ratios $\alpha A\gamma + \gamma A\gamma + \gamma A\beta$ of 1:2:1 and 1:5:1 (respectively 1:10:1) produced amplicons visible in Fig. 7 (respectively in the right panel of Fig. 5). Each of these three experiments, corresponding to the different concentrations, was performed at five different annealing temperatures (reported in Table 1), to test the

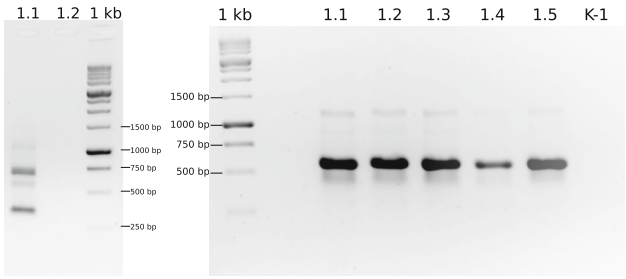

Fig. 6 XPCR implementation of conjugate word blending over gene **B** and over gene **D**, respectively, at different ratios of molecular interference. Left panel: Lane 1.1: reaction with *Taq* polymerase and templates $\alpha B\gamma + \gamma B\gamma + \gamma B\beta$ with concentration ratio 1:2:1 (see 5.2, Table 1, for details). It exhibits a main product of about 300 bp ($\alpha B\beta$) and a secondary product of about 650 bp ($\alpha B\gamma B\beta$). Lane 1.2: negative control without template for reaction in lane 1.1. Right panel: Lanes 1.1, 1.2, 1.3, 1,4, 1.5: reactions with *Pfu* polymerase and templates $\alpha D\gamma + \gamma D\gamma + \gamma D\beta$, with concentration ratio 1:10:1; a different annealing temperature was used for every lane (see 7, Tables 1 and 3). All aforementioned lanes exhibit a main product of about 500 bp ($\alpha D\beta$) and a very faint band of about 1100 bp ($\alpha D\gamma D\beta$). Lanes K-1: negative control without templates for reactions in lanes 1.1, 1.2, 1.3, 1.4, 1.5. Picture from Fig 4.12, Fig 4.13 of (Bellamoli 2013)

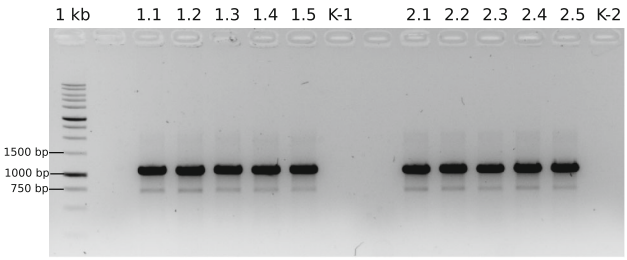

Fig. 7 XPCR implementation of conjugate word blending over gene **A**, at two different ratios of molecular interference. Amplifications with primers (α, $\overline{\beta}$) and *Pfu* polymerase. Lanes 1.1, 1.2, 1.3, 1,4, 1.5: templates $\alpha A\gamma + \gamma A\gamma + \gamma A\beta$, with concentration ratios 1:2:1; a different annealing temperature was used for every lane (see 6.1, Tables 1 and 3 for details). Lane K-1: negative control without templates for reactions in lanes 1.1, 1.2, 1.3, 1.4, 1.5. Lanes 2.1, 2.2, 2.3, 2.4, 2.5: templates $\alpha A\gamma + \gamma A\gamma + \gamma A\beta$, with concentration ratios 1:5:1 and different annealing temperatures for every lane (see 6.2, Tables 1 and 3). Lane K-2: negative control without templates for reactions in lanes 2.1, 2.2, 2.3, 2.4, 2.5. All aforementioned lanes exhibit a main product of about 1061 bp (corresponding to $\alpha A\beta$) and other very faint bands of biased products. Picture from Fig 4.9 of (Bellamoli 2013)

implementation reliability of the conjugate word blending over gene *A*.

We now describe in more details a typical reaction, such as the one shown in Fig. 5, where the conjugate word blending operation is implemented starting from input sequences $\alpha D\gamma$ and $\gamma D\beta$ (respectively from $\alpha A\gamma$, $\gamma A\gamma$, and $\gamma A\beta$, with concentration ratios 1:10:1). The left panel gel of Fig. 5 exhibits the outcome of XPCR (with *Pfu* polymerase) over templates $\alpha D\gamma$ and $\gamma D\beta$ in the form of a band confirming the presence of a product of about 500 bp. Sequencing showed that this product was indeed the amplicon $\alpha D\beta$. On the right panel of Fig. 5, in all five reactions with different temperatures (1.1 through 1.5, while K-1 reports the negative control), the main products of about 1000 bp ($\alpha A\beta$) are evident as the result of an XPCR over templates $\alpha A\gamma + \gamma A\gamma + \gamma A\beta$. A faint band of about 2000 bp is visible as well, possibly containing expected concatenations $\alpha A\gamma A\beta$ (2101 bp), while concatenations $\alpha A\gamma A\gamma A\beta$ (3141 bp) were not formed in observable quantities.

As a conclusion of this section, we observe that the experiments illustrated in Fig. 5 and Fig. 7 demonstrate that XPCR-based implementation of conjugate word blending (Definition 1) with gene $w = A$ is robust, as the same outcome was obtained under different primer annealing temperatures, and different concentrations of the interference molecule. Similar experiments were repeated with genes $w = B$ and $w = D$, with different DNA polymerases, different annealing temperatures, and different interference molecule concentration ratios, as illustrated in Fig. 5 and Fig. 6. This suggests that conjugate word blending can be efficiently implemented in the lab, with no restrictions on the length of w.

4 Material and methods

Reagents PCR buffer, $MgCl_2$, dNTP, GoTaq® DNA Polymerase and Pfu DNA Polymerase were furnished by Promega (Milan, Italy). All the synthetic DNA oligonucleotides and all the primers were from Sigma–Aldrich (Milan, Italy).

Bacterial genes. Gene fragments were extracted from *Burkholderia fungorum* DBT1 (Di Gregorio et al. 2004), an environmental bacterial isolate with remarkable PAH (polycyclic aromatic hydrocarbon) degrading capabilities. In particular three catabolic genes coding for three subunits of the initial dioxygenase were used. They were *dbtAa*(1019 bp) encoding for ferredoxyn reductase, *dbtAb* (311 bp) encoding for ferredoxyn, and *dbtAd* (518 bp) encoding for dioxygenase sub-unit β. Gene sequences are available in GenBank with accession number AF380367 for gene *dbtAd* and AF404408 for genes *dbtAa* and *dbtAb*.

Primers design and PCR conditions All the primers[4] and oligos used in this study were designed and checked with the aid of MATLAB® (The MathWorks, Inc.) and its Bioinformatics toolbox™.

[4] α = 5′-TTCTACAAGGAGGATATTACC-3′, $\overline{\beta}$ = 5′-TATGGAGATGTACCTGATATC-3′, γ = 5′-ATATTGGAGGAGGTATACAAC-3′, $\overline{\gamma}$ = 5′-GTTGTATACCTCCTCCAATAT-3′.

Table 1 PCR conditions (experiment labels in boldface)

1.1, 1.2, 5.1		5.2		Cycles
Temp.	Time	Temp.	Time	
94°C	2′	94°C	2′	
94°C	45″	94°C	45″	x30
51°C	30″	51°C	30″	
72°C	4′	72°C	1′30″	
72°C	5′	72°C	5′	
6, 6.1, 6.2, 7		9		Cycles
Temp.	Time	Temp.	Time	
94°C	5′	94°C	5′	
94°C	1′	94°C	1′	x30
47.9°C	1′	49°C	1′	
48.7°C				
50.7°C				
51.8°C				
53.8°C				
72°C	4′	72°C	2′	
72°C	5′	72°C	5′	

The top table reports amplification experiments executed by *Taq* polymerase, and the bottom table reports experiments executed by *Pfu*. The correspondence between experiment labels and experiments illustrated in previous figures is: Experiment label **1.1** (experiment in Fig. 3, Lane 2), **1.2** (Fig. 3, Lane 3), **5.1** (Fig. 3, Lane 1), **5.2** (Fig. 6, left panel), **7** (Fig. 6, right panel), **9** (Fig. 5, left panel), **6** (Fig. 5, right panel), **6.1** (Fig. 7, lanes 1.1 through 1.5), and **6.2** (Fig. 7, lanes 2.1 through 2.5). See Table 3 for PCR components

All PCR reactions, except otherwise stated, were carried out in 25 µL of total volume containing 0.8 μM of each primer (α and β), 0.4 mM of dNTPs, 2.5 U of *Pfu* DNA polymerase (Promega, Milan, Italy) and 2.5 µL of 10x PCR buffer. Concentration of template was 20 ng per reaction. The negative control for every reaction was obtained by substituting the template with an equivalent volume of sterile mQ water.

PCR thermocycler conditions were: 94°C for 2 min, then 30 cycles of 94°C for 1 min, 1 min of incubation at different annealing temperatures and 72°C for 4 min, with a final extension step at 72°C for 5 min. For the concatenation of three copies of the same gene, amplification was performed over $\alpha A\gamma/\gamma A\gamma/\gamma A\beta$ genes with the following template concentrations: 10 ng/reaction for $\alpha A\gamma$ and $\gamma A\beta$, and 100 ng/reaction for $\gamma A\gamma$. Experiment was carried out at different annealing temperature (47.9, 48.7, 50.7, 51.8, 53.8°C, as in Table 1).

To entirely amplify any fragment of the type primer-gene-primer, a duration of 2 minutes per thermal cycle would have been more than enough, according to standard polymerase protocols, in all cases (of the three genes, and of the two enzymes). The decision to assign at least double that time (4 minutes per thermal cycle, see Table 1) was to ensure that the obtained output of the conjugate blending operation did not change if the duration of each thermal cycle was extended.

Table 2 Electrophoresis buffers. TAE 50X Stock solution, and TBE 10X Stock solution

Reagent	TAE	TBE
Tris base (Sigma Aldrich)	242 g	108 g
Glacial acetic acid (Sigma Aldrich)	57.1 mL	–
Boric acid (Sigma Aldrich)	–	55 g
$Na_2EDTA{\cdot}2H_2O$ (Sigma Aldrich)	37.2 g	–
0.5 mol L^{-1} EDTA (Sigma Aldrich), pH 8.0	–	40 mL
H_2O	to 1 L	to 1 L

Gel Electrophoresis All agarose gels were cast using TAE or TBE (Table 2) buffers with the addition of 0.8-2% Agarose LE Analytical Grade (Promega) and ethidium bromide to a final concentration of 0.5 µg mL^{-1}. Sharpmass™ 1 (0.25 to 1 kb) and Sharpmass™ 100 (100 to 1000bp) were used as DNA mass ladders, both purchased from EuroClone S.p.A (IT); often we refer to them as ladder of 1 kb and 100 bp, respectively. Unless otherwise specified one tenth of the volume of PCR reactions was loaded on every agarose gel with the appropriate amount of 6X loading dye. Agarose gels were prepared at 1 or 2 % (w/vol) on the basis of expected dimensions of PCR products. Electrophoretic runs were carried out in TAE (1X) at 10 volt/cm^2. The presence of bands was detected by a digital gel scanner. The DNA bands (final PCR products) of interest were excised from the gel and further purified through QIAEX® II Gel Extraction Kit (QIAGEN, Milan, Italy) following the manufacturers instructions. Finally, eluted DNA fragments were sequenced on both strands.

Sub-cloning of DNA fragments DNA fragments that needed sequencing were ligated using pGEM®-T Easy Vector Systems (Promega, Milan, Italy) kit with T4 DNA Ligase (Promega, Milan, Italy) following the manufacturer's instructions. An additional step was necessary for PCR products amplified with *Pfu* DNA polymerase, since it produces blunt ended fragments: deoxyadenosine overhangs were added according to the instructions.

5 Conclusion and future work

This paper introduces and studies conjugate word blending, a binary string operation that models the unexpected outcome of the wet lab XPCR procedure under a specific set up, namely when used to attempt concatenating two copies of the same gene. We investigate computational properties

Table 3 PCR components (experiment labels in boldface)

Component	Quantity (in μL)								
	1.1	1.2	5.1	5.2	6	6.1	6.2	7	9
5X Colorless GoTaq® Reaction Buffer	5	5	5	5	–	–	–	–	–
Pfu DNA Polymerase 10X Buffer with $MgSO_4$	–	–	–	–	2.5	2.5	2.5	2.5	2.5
dNTP mix, 10mM each	1	1	1	1	1	1	1	1	1
α primer	1	1	1	1	1	1	1	1	1
$\overline{\beta}$ primer	1	1	1	1	1	1	1	1	1
GoTaq® DNA Polymerase	0.5	0.5	0.5	0.5	–	–	–	–	–
Pfu DNA Polymerase	–	–	–	–	0.5	0.5	0.5	0.5	0.5
Template $\alpha A\gamma$	1	–	1	–	1	1	1	–	–
Template $\alpha B\gamma$	–	1	–	1	–	–	–	–	–
Template $\alpha D\gamma$	–	–	–	–	–	–	–	1	2
Template $\gamma A\gamma$	–	–	1	–	10	2	5	–	–
Template $\gamma B\gamma$	1	–	–	2	–	–	–	–	–
Template $\gamma D\gamma$	–	1	–	–	–	–	–	10	–
Template $\gamma A\beta$	–	1	1	–	1	1	1	–	–
Template $\gamma B\beta$	–	–	–	1	–	–	–	–	–
Template $\gamma D\beta$	1	–	–	–	–	–	–	1	2
Sterile mQ H_2O	13.5	13.5	13.5	12.5	7	15	12	7	15
	Total volume: 25								

Experiment labels **1.1** and **1.2** refer to the overlap concatenation of two different genes, obtained in Lane 2 and Lane 3 of Fig. 3, respectively. Label **5.1** reports details of the experiment producing the conjugate word blending operation, on inputs involving only gene *A*, in Lane 1 of Fig. 3. Labels **9** and **6** refer to the left and right panel of Fig. 5, respectively, where the conjugate word blending operation was realized both without (left panel), and with (right panel) the presence of an interference molecule. Labels **5.2** and **7** refer to the left and right panels of Fig. 6, respectively, where the conjugate word blending output, involving gene *B* (left gel) and gene *D* (right gel), was produced. Labels **6.1** and **6.2** refer to the results reported in Fig. 7, where the conjugate word blending output involving the gene *A* was obtained at different concentration of the interference molecule. See PCR cycles in Table 1

of this operation, and prove that the classes of regular and recursively enumerable languages are closed under conjugate word blending, while the classes of context-free and context-sensitive languages are not. We also report the wet lab experiments that conjugate word blending is modelled upon, with three bacterial genes of different lengths, and verify its outcome under several experimental conditions, such as using different DNA polymerase enzymes (*Taq* and *Pfu*), different primer annealing temperatures, and in the presence of a so-called interference molecule, at various concentration ratios to the template molecules.

While in (Franco et al. 2017) it was hypothesized that the unexpected behaviour of XPCR under these specific conditions is caused by strand displacement (template switching and/or hydrolysis of competing strands (Kanagawa 2003)), in this paper we propose another explanation for this phenomenon, illustrated in Fig. 4. Further experimental work is needed to validate this explanation, and the exact mechanisms of the observed molecular biology phenomenon on which the conjugate word blending is based.

As a future work we will consider to carry out a more general experimental validation of the word blending operation as originally defined (Enaganti et al. 2020), including the particular case when (only) one of the two flanking strings γ_1 or γ_2 may be the empty word, and to develop theoretical investigations of its variants which more closely model the experimental reality of DNA string computation implemented by the XPCR wet lab procedure.

References

Amos M (2005) Theoretical and experimental DNA computation. Springer, Berlin. https://doi.org/10.1007/3-540-28131-2

Bellamoli F (2013) Production of Gene Libraries by Multiple XPCR. Master's thesis, University of Verona, Department of Biotechnology, Italy, https://doi.org/10.13140/RG.2.2.34146.96968

Bonizzoni P, De Felice C, Zizza R (2005) The structure of reflexive regular splicing languages via Schützenberger constants. Theoret Comput Sci 334(1–3):71–98. https://doi.org/10.1016/j.tcs.2004.12.033

Brzozowski JA, Kari L, Li B, Szykuła M (2018) State complexity of overlap assembly. In: Câmpeanu C (ed) Implementation and

Application of Automata - 23rd International Conference, CIAA 2018, Springer, Lecture Notes in Computer Science, vol 10977, pp 109–120, https://doi.org/10.1007/978-3-319-94812-6_10
Carausu A, Păun G (1981) String intersection and short concatenation. Revue Roumaine de Mathématiques Pures et Appliquées 26(5):713–726
Ceterchi R (2006) An algebraic characterization of semi-simple splicing. Fundam Inform 73(1–2):19–25
Csuhaj-Varjú E, Petre I, Vaszil G (2007) Self-assembly of strings and languages. Theoret Comput Sci 374(1–3):74–81. https://doi.org/10.1016/j.tcs.2006.12.004
Di Gregorio S, Zocca C, Sidler S, Toffanin A, Lizzari D, Vallini G (2004) Identification of two new sets of genes for dibenzothiophene transformation in Burkholderia sp. DBT1. Biodegradation 15:111–123. https://doi.org/10.1023/B:BIOD.0000015624.52954.b6
Domaratzki M (2009) Minimality in template-guided recombination. Inf Comput 207(11):1209–1220. https://doi.org/10.1016/j.ic.2009.02.009
Enaganti SK, Ibarra OH, Kari L, Kopecki S (2017a) Further remarks on DNA overlap assembly. Inf Comput 253:143–154. https://doi.org/10.1016/j.ic.2017.01.009
Enaganti SK, Ibarra OH, Kari L, Kopecki S (2017b) On the overlap assembly of strings and languages. Nat Comput 16:175–185. https://doi.org/10.1007/s11047-015-9538-x
Enaganti SK, Kari L, Ng T, Wang Z (2020) Word blending in formal languages. Fundam Inform 171(1–4):151–173. https://doi.org/10.3233/FI-2020-1877
Franco G (2005) A polymerase based algorithm for SAT. In: Coppo M, Lodi E, Pinna GM (eds) Theoretical Computer Science, 9th Italian Conference, ICTCS 2005, Springer, Lecture Notes in Computer Science, vol 3701, pp 237–250, https://doi.org/10.1007/11560586_20
Franco G, Manca V (2011) Algorithmic applications of XPCR. Nat Comput 10(2):805–819. https://doi.org/10.1007/s11047-010-9199-8
Franco G, Giagulli C, Laudanna C, Manca V (2005) DNA extraction by XPCR. In: Ferretti C, Mauri G, Zandron C (eds) DNA Computing, 10th International Workshop on DNA Computing, DNA 10, Springer, Lecture Notes in Computer Science, vol 3384, pp 104–112, https://doi.org/10.1007/11493785_9
Franco G, Manca V, Giagulli C, Laudanna C (2006) DNA recombination by XPCR. In: Carbone A, Pierce NA (eds) DNA Computing, 11th International Workshop on DNA Computing, DNA 11, Springer, Lecture Notes in Computer Science, vol 3892, pp 55–66, https://doi.org/10.1007/11753681_5
Franco G, Bellamoli F, Lampis S (2017) Experimental analysis of XPCR-based protocols. arXiv preprint arXiv:171205182
Golan JS (1992) The Theory of Semirings with Applications in Mathematics and Theoretical Computer Science, Pitman Monographs and Surveys in Pure and Applied Mathematics, vol 54. Longman Scientific & Technical
Goode E, Pixton D (2007) Recognizing splicing languages: syntactic monoids and simultaneous pumping. Discret Appl Math 155(8):989–1006. https://doi.org/10.1016/j.dam.2006.10.006
Head T (1987) Formal language theory and DNA: an analysis of the generative capacity of specific recombinant behaviors. Bull Math Biol 49:737–759. https://doi.org/10.1007/BF02481771
Holzer M, Jakobi S (2011) Chop operations and expressions: Descriptional complexity considerations. In: Mauri G, Leporati A (eds) Developments in Language Theory - 15th International Conference, DLT 2011, Springer, Lecture Notes in Computer Science, vol 6795, pp 264–275, https://doi.org/10.1007/978-3-642-22321-1_23
Holzer M, Jakobi S (2012) State complexity of chop operations on unary and finite languages. In: Kutrib M, Moreira N, Reis R (eds) Descriptional Complexity of Formal Systems - 14th International Workshop, DCFS 2012, Springer, Lecture Notes in Computer Science, vol 7386, pp 169–182, https://doi.org/10.1007/978-3-642-31623-4_13
Holzer M, Jakobi S, Kutrib M (2017) The chop of languages. Theoret Comput Sci 682:122–137. https://doi.org/10.1016/j.tcs.2017.02.002
Hommelsheim CM, Frantzeskakis L, Huang M, Ülker B (2015) PCR amplification of repetitive DNA: a limitation to genome editing technologies and many other applications. Sci Rep 4(5052):1–13. https://doi.org/10.1038/srep05052
Ignatova Z, Martínez-Pérez I, Zimmermann KH (2008) DNA computing models. Springer, Berlin. https://doi.org/10.1007/978-0-387-73637-2
Ito M, Lischke G (2007) Generalized periodicity and primitivity for words. Math Logic Q 53(1):91–106. https://doi.org/10.1002/malq.200610030
Kalle E, Kubista M, Rensing C (2014) Multi-template polymerase chain reaction. Biomol Detect Quantif 2:11–29. https://doi.org/10.1016/j.bdq.2014.11.002
Kanagawa T (2003) Bias and artifacts in multitemplate polymerase chain reactions (PCR). J Biosci Bioeng 96(4):317–323. https://doi.org/10.1016/S1389-1723(03)90130-7
Kari L (1991) On insertion and deletion in formal languages. PhD thesis, University of Turku
Kari L, Seki S, Sosík P (2012) DNA computing–foundations and implications. In: Rozenberg G, Bäck T, Kok JN (eds) Handbook of Natural Computing, Springer, pp 1073–1127, https://doi.org/10.1007/978-3-540-92910-9_33
Manca V, Franco G (2008) Computing by polymerase chain reaction. Math Biosci 211(2):282–298. https://doi.org/10.1016/j.mbs.2007.08.010
Păun G, Rozenberg G, Salomaa A (1998) DNA computing: new computing paradigms. Springer. https://doi.org/10.1007/978-3-662-03563-4
Pixton D (2000) Splicing in abstract families of languages. Theoret Comput Sci 234(1–2):135–166. https://doi.org/10.1016/S0304-3975(98)00046-2
Păun G (1996) On the splicing operation. Discret Appl Math 70(1):57–79. https://doi.org/10.1016/0166-218X(96)00101-1
Rozenberg G, Salomaa A (eds) (1997) Handbook of formal languages, Vol. 1: Word, Language, Grammar. Springer, https://doi.org/10.1007/978-3-642-59136-5
Salomaa A (1973) Formal languages. Academic Press Inc, Cambridge