Nat Comput (2010) 9:297–316
DOI 10.1007/s11047-009-9131-2

Watson–Crick palindromes in DNA computing

Lila Kari · Kalpana Mahalingam

Published online: 20 May 2009

Abstract This paper provides an overview of existing approaches to encoding information on DNA strands for biocomputing, with a focus on the notion of Watson–Crick (WK) palindromes. We obtain a closed form for, as well as several properties of WK palindromes: The set of WK-palindromes is dense, context-free, but not regular, and is in general not closed under catenation and insertion. We obtain some properties that link the WK palindromes to classical notions such as that of primitive words. For example we show that the set of WK-palindromic words that cannot be written as the product of two non-empty WK-palindromes equals the set of primitive WK-palindromes. We also investigate various simultaneous Watson–Crick conjugate equations of words and show that the equations have, in most cases, only Watson–Crick palindromic solutions. Our results hold for more general functions, such as arbitrary morphic and antimorphic involutions.

Keywords Theoretical DNA computing · DNA encodings · Combinatorics of words · Palindromes · Watson–Crick palindromes

1 Introduction

Theoretical DNA Computing is an area of biomolecular computing that loosely encompasses contributions to fundamental research in computer science originated in or motivated by research in DNA computing. Examples are numerous and they include theoretical aspects of self-assembly (Adleman 2000; Soloveichik and Winfree 2006), DNA sequence design (Garzon et al. 2006; Marathe et al. 1999), and mathematical properties of

L. Kari (✉) · K. Mahalingam
Department of Computer Science, University of Western Ontario, London, ON N6A 5B7, Canada
e-mail: lila@csd.uwo.ca

Present Address:
K. Mahalingam
Department of Mathematics, Indian Institute of Technology, Chennai 600035, India
e-mail: kmahalingam@iitm.ac.in

DNA-encoded information (Domaratzki 2006; Daley and McQuillan 2006). One of the most active areas of research in theoretical DNA computing is the search for ways to encode information on DNA for the purposes of biocomputation that ensure that no unwanted bindings occur. The main premise is that information-encoding strings that are used in DNA computing experiments have an important property that differentiates them from their electronic computing counterparts. This property is the Watson–Crick complementarity between DNA single-strands that allows information-encoding strands to potentially interact.

Recall that a *DNA single-strand* consists of four different types of units called *nucleotides* or *bases* strung together by an oriented *backbone* like beads on a wire. The bases are Adenine (A), Guanine (G), Cytosine (C) and Thymine (T), and A can chemically bind to an opposing T on another single strand, while C can similarly bind to G. Bases that can thus bind are called *Watson–Crick (WK) complementary*. A DNA single strand is assigned its direction based on what is found at the end of the strand: it can have the direction $5' \rightarrow 3'$ or $3' \rightarrow 5'$. Two DNA single strands with opposite orientation (one of them $5' \rightarrow 3'$ and the other $3' \rightarrow 5'$) and with WK complementary bases at each position can bind to each other to form a *DNA double strand* in a process called *base-pairing*, *annealing*, or *hybridization*. Note that in this paper we omit writing the orientation of a DNA strand by using the convention that any DNA sequence will represent a single strand in its $5' \rightarrow 3'$ orientation. It is now apparent that, when encoding information on single DNA strands, care must be taken that the strands do not interact in undesirable ways. One such situation can occur, for example, if a DNA strand has its first half WK complementary to its second half. In this case, the DNA strand will bind to itself forming a secondary structure called a hairpin (Fig. 1). This further implies that the information encoded on this hairpin will be de facto unavailable for future biocomputational steps. Such secondary structures have to be thus avoided by carefully designing the information-encoding DNA strands.

This paper aims to give an overview of the existing research into ways to optimally encode information on DNA single-strands for the purposes of DNA computing, followed by a focus on the specific concept of Watson–Crick palindromes and their theoretical properties. The paper is organized as follows.

Section 2 discusses existing approaches to the problem of finding good DNA encodings for biocomputations. The remainder of the paper investigates in depth a specific type of interaction that has to be avoided in DNA computing, namely that between Watson–Crick palindromes.

Section 3 describes basic properties of θ-palindromes, where θ is an antimorphic involution modelling the Watson–Crick complementarity relation. For an antimorphic involution, Lemma 4 gives a closed form for any θ-palindrome w, as being $w = p(qp)^i$, where p, q are both θ-palindromes.

In Sect. 4 we show, Proposition 6, that both the set of all palindromic words and the set of all non-palindromic words are dense for an antimorphic involution θ, providing thus a rich choice for biocomputational purposes. In fact, Lemma 11 gives the number of WK-palindromes of length $2k$, which is precisely 4^k. We also show that, for an antimorphic involution, the set of all θ-palindromes is not regular, Lemma 9, but context-free,

Fig. 1 Intramolecular hybridization: DNA secondary structure avoided in a hairpin-free language

Proposition 7. In the case of a morphic involution the situation is different. Indeed, Lemma 10 shows that if $\theta(a) \neq a$ for any $a \in \Sigma$, then the set of θ-palindromes contains only the empty word.

Section 5 solves several simultaneous WK-conjugate word equations. In most cases the solutions to these equations are θ-palindromes.

Section 6 discusses various closure and other properties of θ-palindromes, interesting for biocomputational purposes. For an antimorphic involution, in general, the set of θ-palindromes is not closed under concatenation, Lemma 13, or insertion Lemma 15. Lemma 19 provides a connection between θ-palindromes, θ-commutativity, and primitive words: For an antimorphic involution, u θ-commutes with v iff both v and primitive root of v can be written as product of two nonempty palindromes. Finally, Corollary 6 shows that for an antimorphic involution, the set of θ-palindromic words that cannot be written as the product of two nonempty θ-palindromes equals the set of primitive θ-palindromes.

Section 7 points to future work in this area.

2 DNA encodings for biocomputation

Most DNA-based computations consist of three basic stages. The first is *encoding* the input data using single- or double-stranded DNA molecules, the second is performing the *biocomputation* using bio-operations and the third is *decoding* the result. One of the main problems associated with such biocomputations is the design of the information-encoding oligonucleotides (short DNA strands, 6–20 bases each) such that undesirable pairing due to the Watson–Crick complementarity is minimized. Indeed, in laboratory biocomputing experiments, the complementarity of the bases may pose potential problems, for example if some DNA strands partially bind other DNA strands that are not their complete complements. Several approaches exist that address this sequence design problem. In this section we briefly discuss the software simulation approach, the algorithmic approach and the theoretical approach to the design of optimal data-encoding DNA strands.

The first approach, *software simulation tools*, verifies biocomputation protocol correctness before it is carried out in a laboratory experiment. Several software packages (Hartemink and Gifford 1999; Hartemik et al. 1999; Feldkamp et al. 2000, 2001) written for DNA computing purposes are available. For example the simulation software *Edna* simulates biochemical processes and reactions that can occur during a laboratory experiment. *Edna* (Garzon and Oehman 2001) is a simulation tool that uses a cluster of PCs and demonstrates the processes that could happen in test tubes. *Edna* can be used to determine if a particular choice of encoding strategy is appropriate, to test a proposed protocol and estimate its performance and reliability, and even to help assess the complexity of the protocols. Test tube operations are assigned a cost that takes into account many of the reaction conditions. The measure of complexity used by *Edna* is the sum of the costs added up over all operations in a protocol. Other features offered by the software allow the prediction of DNA melting temperature (the temperature at which a DNA double strand dissociates into single strands) taking into account various reaction conditions. All molecular interactions simulated by the software are local and reflect the randomness inherent in biomolecular processes.

The second approach to finding optimal DNA encodings is the *algorithmic method*. In most DNA based computations there is an assumption that a strand will bind only to its perfect Watson–Crick complement. For example, the results of DNA computations are retrieved from test tubes by using strands that are complementary to the ones used in the biocomputation. However, in practice it is possible for a DNA molecule to bind to another molecule which differs from its complementary molecule by a few nucleotides, simply by virtue of the strength of the bond between the remaining "perfect-match" complementary bases. One way to avoid this is to ensure that every two molecules in the solution differ in more than d locations, where d is a number that is determined by experimental observations. This property can be formalized in terms of the Hamming distance between two DNA strands modelled as two strings w_1 and w_2 over the DNA alphabet $\{A, C, G, T\}$. The Hamming distance between two strings w_1 and w_2 of equal length is denoted by $H(w_1, w_2)$ and is defined as the number of locations in which two given words w_1 and w_2 are distinct. For a set of DNA words, the Hamming distance constraint requires that any two words w_1 and w_2 in the set have $H(w_1, w_2) \geq d$, where d is a given positive number. The second constraint that is usually imposed is that for any two words w_1, w_2 in the solution, we have $H(w_1, WK(w_2)) \geq d$, where for a word w, $WK(w)$ denotes its Watson–Crick complement. This constraint is necessary to ensure, for example, that retrieving the output of a biocomputation (usually done by hybridizing it with WK complement of parts of the expected output strand) proceeds error-free. Another consideration is that, when retrieving the results from the solution, hybridization should occur simultaneously for all molecules in the solution. This implies that respective melting temperatures should be comparable for all hybridization reactions that are taking place. This is the third main constraint that the set of words under consideration needs to adhere to.

To address the design of DNA code words according to these three constraints, an algorithm based on a stochastic local search method was proposed in Tulpan et al. (2003). The melting temperature constraint was simplified to the constraint requiring that the percentage of C and G nucleotides in each strand be 50%. The algorithm produces a set of DNA sequences that satisfies the Hamming distance and the temperature constraints:

Input: *Number k of words to be produced and the word length n.*
Step 1: *Produce a random set of k words of length n each.*
Step 2: *Modify the set so that the set satisfies the first constraint.*
Step 3: *Repeat Step 2 for all the given constraints.*
Output: *The set of words (if one can be found).*

More specifically, given the current word set, two words w_1 and w_2 are chosen from the set that violate at least one of the constraints. With a probability $1 - \gamma$, γ being the noise parameter, one of these words is altered by randomly substituting one base in a way that maximally decreases the number of conflict violations. The algorithm terminates either when there are no more conflicts in the set of words, or when the number of loop iterations has exceeded some maximum threshold. Empirical results prove this technique to be effective and the noise parameter γ is empirically determined to be optimal as 0.2, regardless of the problem instance.

The third approach to the problem of designing DNA code words is the *formal language theoretical approach* introduced by Kari et al. in Hussini et al. (2003). (For an introduction to formal language theory the reader is referred to Hopcroft et al. 2001, and for combinatorics of words to Lothaire 1997, Shyr 2001.) Every biomolecular protocol involving DNA generates molecules whose sequences of nucleotides form a language over the four letter alphabet $\Delta = \{A, G, C, T\}$. The Watson–Crick complementarity of the nucleotides can be formalized by an involution mapping θ, $A \mapsto T$ and $G \mapsto C$ which is an antimorphism on Δ^*. An involution θ is a mapping such that θ^2 is identity. An antimorphism θ is such that $\theta(uv) = \theta(v)\theta(u)$ for all words u, v from Δ^*. As Watson–Crick bonds are generally undesirable from a biocomputational perspective, they can be avoided for a given language, if the language satisfies certain properties, as described below.

There are two types of unwanted hybridizations: intramolecular and intermolecular. The *intramolecular hybridization* happens when two sequences, one being the reverse complement of the other appear within the same DNA strand (Fig. 1). In this case the DNA strand forms a hairpin. A language is called hairpin-free if its words cannot form such hairpin structures. Hairpin-free languages have been defined (Kari et al. 2005a) and studied, for example, in Kari et al. (2005a) and Domaratzki (2006).

Before introducing the formal definitions, we review some basic notations. An alphabet is a finite, non-empty set of symbols. Let Σ be such an alphabet. Then Σ^* denotes the set of all words over this alphabet, including the empty word λ. Σ^+ is the set of all non-empty words over Σ. The length of a word $u \in \Sigma^*$ is denoted by $|u|$, and Σ^i denotes the set of all words over Σ of length i. A language L over Σ is a subset of Σ^*. We denote by $\mathrm{Sub}_k(L)$, the set of all subwords of length k of words from a language L.

Suppose now that we want to avoid the type of hybridization shown in Fig. 1 between all the words of a given language L. We can achieve that by imposing the condition that L be a WK-k-m-subword code, where WK is the Watson–Crick complementarity function over the DNA alphabet Δ. A language L is called (Jonoska et al. 2005) a *θ-k-m-subword code* if for all words $u \in \Sigma^k$ we have $\Sigma^* u \Sigma^i \theta(u) \Sigma^* \cap L = \emptyset$, $1 \leq i \leq m$. This means that no word in a θ-k-m-subword code contains two complementary subwords of length k that are at most m bases apart. This further implies that, for example, in a DNA language with this property no unwanted secondary structures such as hairpins with stems that are k bases long and with loops that are up to m bases in length, can form.

DNA strand sets that avoid all types of unwanted *intermolecular bindings* (Fig. 2) were introduced in Jonoska et al. (2005) under the name of θ-k-codes, where θ denoted an arbitrary antimorphic involution. A language L is said to be θ-k-code if $\theta(x) \neq y$ for all $x, y \in \mathrm{Sub}_k(L)$. The relationship $\theta(x) = y$ indicates that the molecules corresponding to x and y can form complementarity bonds between them as shown in Fig. 2. For a suitable k, a θ-k-code avoids several types of unwanted intermolecular hybridizations.

Besides being theoretically interesting, properties such as the θ-k-code property are meant to ensure that DNA strands cannot form unwanted hybridizations during DNA computations, and has been successfully tested in practical laboratory experiments (Jonoska et al. 2005). In Kari et al. (2005b), the concept of θ-k-code has been extended to the *bond-free* property which requires that $H(\theta(x), y) > d$ for any subwords $x, y \in \mathrm{Sub}_k(L)$, where H is the Hamming distance function between two words.

Suppose we use codes that have one or more of the desirable language properties we have described. What may happen during the course of computation is that the properties initially present deteriorate over time. This leads to another issue, namely to investigate how bio-operations such as cutting, pasting, splicing, contextual insertion, and deletion

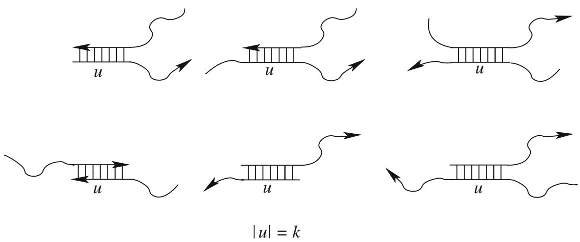

Fig. 2 Various intermolecular hybridizations of DNA single strands, one of which contains a subword of length k, while the other contains its WK complement. A θ-k-code avoids any DNA secondary structures like the ones above

affect the various bond-free properties of DNA languages. Invariance under these bio-operations has been studied in Jonoska et al. (2005, 2006), Kari et al. (2003). Bounds on the sizes of some other codes with desirable properties that can be constructed were explored by Marathe et al. (1999). More recently, the concepts of involution-bordered and unbordered words, (Kari and Mahalingam 2007a), as well as Watson–Crick conjugate and Watson–Crick commutative words, (Kari and Mahalingam 2007b), were introduced and studied from an algebraic point of view, as formal models of DNA strands that can form various types of bonds.

In addition to being of interest in DNA computing experiments, the newly defined notions such as bond-free languages, hairpin-free languages, involution-bordered words, Watson–Crick commutative and Watson–Crick conjugate words are of theoretical interest since they turned out to be proper generalizations of classical notions in the theory of codes and combinatorics of words such as prefix codes, suffix codes, infix codes, comma-free codes, bordered words, commutative and conjugate words. In the remainder of the paper we will investigate one such concept, the Watson–Crick palindrome, which is a generalization of the classical notion of palindrome, and which arose from studying information encoding in the DNA computing context.

3 Watson–Crick palindromes

The notion of θ-palindrome was defined in Kari and Mahalingam (2007b) and obtained independently in de Luca (2006). Note that if θ is the Watson–Crick involution, then the notion of Watson–Crick palindromes (Fig. 3) coincides with the term "palindrome" as used in molecular biology, especially in the study of enzymes.

A restriction enzyme (or restriction endonuclease) is an enzyme that "recognizes" a specific double-stranded DNA subsequence and cuts the double-stranded DNA according to a pattern that is specific for each enzyme. The result is either two "blunt-cut" DNA double-strands, or two DNA strands that are partially double-stranded and partially single stranded, with the single-stranded parts usually called "sticky ends". While recognition sequences vary widely, many of them are palindromic: The sequence on the "top strand" read in the $5' \rightarrow 3'$ direction is the same as the sequence on the "bottom strand" read in the $5' \rightarrow 3'$ direction. The meaning of "palindromic" in this context is different from what one might expect from its linguistic usage: 5′-GTAATG-3′ is not a palindromic DNA sequence, but 5′-GTATAC-3′ is (5′-GTATAC-3′ is WK complementary to 3′-CATATG-5′, which is the same as 5′-GTATAC-3′). It is exactly this biological meaning of the word "palindrome" that we attempt to model here, by the notion of Watson–Crick palindrome. Using our formalization and convention on strand directionality, if WK denotes the Watson–Crick antimorphic involution, WK(GTATAC) = GTATAC.

Thus, the study of θ-palindromes for antimorphic involutions is interesting from two points of view: firstly, it may be desirable for certain DNA computing experiments to use DNA strands that contain θ-palindromic enzyme restriction sites as subwords, and secondly, in general, a set of DNA codewords should be free of θ-palindromic words, due to the intermolecular hybridizations that these would entail.

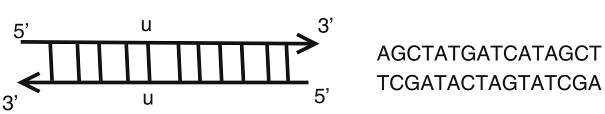

Fig. 3 An example of a Watson–Crick palindrome

The notion of θ-palindrome was introduced and studied in Kari and Mahalingam (2007b), whereby a relation on words was defined using the θ-commutativity and it was showed that, for an antimorphic involution θ, the set of all θ-palindromes can be characterized using this relation. In this paper we study several closure and algebraic properties of θ-palindromes where θ is an arbitrary involution function. In particular we concentrate on θ-palindromes where θ is the Watson–Crick involution.

This section recalls some definitions, introduces the notion of θ-palindrome and proves some basic properties of θ-palindromes. For example, Lemma 4 provides a closed form for θ-palindromes when θ is an antimorphic involution.

We begin by reviewing some basic notions in combinatorics of words. A bordered word is a nonempty word that has a non-empty prefix equal to one of its suffixes. A word which is not bordered is called unbordered. Bordered words have been also called overlapping or unipolar words and unbordered words have also been called non-overlapping, dipolar or d-primitive words. For properties of bordered and unbordered words we refer the reader to Yu (1998, 2005). In Kari and Mahalingam (2007a), we extended the concept of bordered words to involution-bordered words and studied some of its algebraic properties. We now recall some definitions introduced and used in Kari and Mahalingam (2007a, b).

Definition 1 Let θ be either a morphic or an antimorphic involution on Σ^*.

1. A word $u \in \Sigma^+$ is said to be θ-bordered if there exists $v \in \Sigma^+$ such that $u = vx = y\theta(v)$ for some $x, y \in \Sigma^+$.
2. A non-empty word which is not θ-bordered is called θ-unbordered.
3. A word u is a θ-conjugate of another word w if $uv = \theta(v)w$ for some $v \in \Sigma^*$.
4. A word u is said to θ-commute with v if $uv = \theta(v)u$.
5. A word $x \in \Sigma^*$ is called a θ-palindrome if $x = \theta(x)$.

We also recall some of the basic observations based on the above definition (Kari and Mahalingam 2007b). For a given alphabet Σ, and a morphic or an antimorphic involution θ, let B_θ denote the set of all θ-bordered words over Σ^* and P_θ denote the set of all θ-palindromes. We denote by $\bar{P_\theta}$ the set of all non θ-palindromes. Note that if θ is the morphic involution, then $P_\theta = \Gamma^*$ where $\Gamma \subseteq \Sigma$ and $\theta(a) = a$ for all $a \in \Gamma$ and $\theta(a) \neq a$ for all $a \in \Sigma \setminus \Gamma$. Throughout the paper we assume that the alphabet Σ is such that $|\Sigma| \geq 2$ and the involution θ is not the identity function.

Lemma 1 *Let θ be either a morphic or an antimorphic involution and let Σ be such that for all $a \in \Sigma$, $a \neq \theta(a)$.*

1. *A θ-palindrome $x \in \Sigma^+$ has length greater than or equal to 2.*
2. *For all $a \in \Sigma$, $a \in \bar{P_\theta}$.*
3. *For all $a \in \Sigma$, $a^n \in \bar{P_\theta}$ for all $n \geq 1$.*

A word u is called primitive if it is not a power of another word, i.e., there exits no word z such that $w = z^k$ for some $k > 1$. If u is not primitive such that $u = z^k$ then the primitive root of u is z and is denoted by $\sqrt{u}$. We have the following observation.

Observation 1 *Let θ be either a morphic or an antimorphic involution and let $u \in \Sigma^*$. Then*

1. *$u \in P_\theta$ iff $\sqrt{u} \in P_\theta$*
2. *$u \in P_\theta$ iff $u^n \in P_\theta$ for all $n \geq 1$.*

Lemma 2 *Let θ be an antimorphic involution and for all $a \in \Sigma$ let $a \neq \theta(a)$. Then $x \in \Sigma^+$ is a θ-palindrome iff $x = ay\theta(a)$ for some $a \in \Sigma$ and $y \in P_\theta$.*

Proof If x is a θ-palindrome then $x = \theta(x)$. Let $x = aq$ for some $a \in \Sigma$ and $q \in \Sigma^*$. Then $\theta(x) = \theta(q)\theta(a)$ and since $x = \theta(x)$, we have $aq = \theta(q)\theta(a)$. If $q = \lambda$ then $a = \theta(a)$ a contradiction to our assumption. Thus $q \in \Sigma^+$ and there exists $y \in \Sigma^*$ and $b \in \Sigma$ such that $q = yb$ and $x = aq = ayb = \theta(b)\theta(y)\theta(a)$. Thus $b = \theta(a)$ and $y = \theta(y)$ and $x = ay\theta(a)$ with $y \in P_\theta$. The converse is obvious. □

We recall the following propositions from Kari and Mahalingam (2007b) and Lyndon and Schutzenberger (1962) regarding conjugacy, commutativity, θ-conjugacy and θ-commutativity of words, which we will use in this paper.

Proposition 1 (Lyndon and Schutzenberger 1962) *Let $u, v, w \in \Sigma^+$ such that $uv = vw$. Then there exist $p, q \in \Sigma^+$ such that $u = pq$, $w = qp$* and *$v = p(qp)^i$.*

Proposition 2 (Lyndon and Schutzenberger 1962) *Let $u, v \in \Sigma^+$ such that $uv = vu$. Then both u and v are powers of a common word.*

Proposition 3 (Kari and Mahalingam 2007b) *Let $u, v, w \in \Sigma^+$ such that $uv = \theta(v)w$.*

1. *If θ is a morphic involution, then there exist $x, y \in \Sigma^*$ such that $u = xy$ and one of the following hold*:
   (a) *$w = y\theta(x)$ and $v = (\theta(xy)xy)^i\theta(x)$ for some $i \geq 0$.*
   (b) *$w = \theta(y)x$ and $v = (\theta(xy)xy)^i\theta(xy)x$ for some $i \geq 0$.*
2. *If θ is an antimorphic involution, then either $u = xy$ and $w = y\theta(x)$ for some $x, y \in \Sigma^*$* or $u = \theta(w)$.

We recall the following result from Kari–Mahalingam–Seki.

Proposition 4 (Kari–Mahalingam–Seki) *Let θ be an antimorphic involution and let $u \in \Sigma^+$ such that $u = \alpha\beta$ for some non-empty $\alpha, \beta \in P_\theta$. Then there uniquely exist two distinct θ-palindromes $x, y \in P_\theta$* and *$n \geq 1$, such that $u = (xy)^n$ and every factorization $u = pq$, $p, q \in \Sigma^+ \cap P_\theta$, has the property that $p = x(yx)^i$, $q = y(xy)^j$ such that $i + j = n - 1$.*

In (Kari–Mahalingam–Seki) the words x and y have been called the *antimorphic twin-roots of u* relative to θ, or simply antimorphic twin-roots of u, if θ is obvious from the context. It was also shown in Kari–Mahalingam–Seki that if a word u can be decomposed as a product of two non-empty θ-palindromes then the primitive root of u is the catenation of its antimorphic twin-roots.

Proposition 5 (Kari and Mahalingam 2007b) *Let $u, v \in \Sigma^+$ such that u θ-commutes with v, i.e., $uv = \theta(v)u$.*

1. *If θ is a morphic involution, then one of the following hold*:
   (a) *$u = \alpha^n$, $v = \alpha^m$ for $\alpha \in P_\theta$, $m, n \geq 1$.*
   (b) *$u = \theta(\alpha)[\alpha\theta(\alpha)]^n$, $v = [\alpha\theta(\alpha)]^m$ for some $m \geq 1$ and $k \geq 0$.*
2. *If θ is an antimorphic involution, then $u = \alpha(\beta\alpha)^n$, $v = (\beta\alpha)^m$ for some $\alpha, \beta \in P_\theta$, $m \geq 1$ and $n \geq 0$.*

Note that for an antimorphic involution θ if $uv = \theta(v)u$ then v can be written as a product of two palindromes and, from Proposition 4, we deduce the existence of unique distinct θ-palindromes x, y such that $v = (xy)^n$ and such that every factorization of v into two non-empty θ-palindromes $v = pq$ has the property that p and q can be written in terms of x and y. We have thus the following result.

Lemma 3 *Let θ be an antimorphic involution and let $u, v \in \Sigma^+$ such that u θ-commutes with v. Then $u = x(yx)^j$, $v = (yx)^i$ for some $i \geq 1$ and $j \geq 0$ where x and y are the antimorphic twin-roots of v.*

It was shown in Kari and Mahalingam (2007b) that for an antimorphic involution θ, $w \in P_\theta$ iff there exists $v \in \Sigma^*$ such that $v \neq w$ and $w = vx = \theta(x)v$ for some $x \in \Sigma^+$. We also show a similar kind of relation (Lemma 5) between the words that θ-commute and the set of all Watson–Crick palindromes. Using this result and Proposition 5 we can deduce the following.

Lemma 4 *Let θ be an antimorphic involution. Then $w \in P_\theta$ iff $w = \alpha(\beta\alpha)^i$ for some $\alpha, \beta \in P_\theta$ and $i \geq 0$.*

Lemma 5 *Let θ be an antimorphic involution and let $u, v \in \Sigma^+$ such that $uv \in P_\theta$. Then,*

1. *u θ-commutes with v iff $u \in P_\theta$.*
2. *v θ-commutes with $\theta(u)$ iff $v \in P_\theta$.*

Proof

1. Let u θ-commute with v. Then $uv = \theta(v)u$ and by Proposition 5 we have $u = \alpha(\beta\alpha)^i$ for some $\alpha, \beta \in P_\theta$ which implies that $u \in P_\theta$. Conversely let $u \in P_\theta$. Given that $uv \in P_\theta$, we have $uv = \theta(uv) = \theta(v)\theta(u) = \theta(v)u$ which implies that u θ-commutes with v.
2. Similar. □

Lemma 6 *Let θ be an antimorphic involution. Then $u \in P_\theta$ iff there exists a $v \in \Sigma^+$ such that u θ-commutes with v.*

Proof Let $u \in P_\theta$. Then for $v = u$ we have $uv = \theta(v)u$ i.e., u θ-commutes with itself. Conversely let u θ-commute with v for some $v \in \Sigma^+$. Then from Proposition 5 there exist $\alpha, \beta \in P_\theta$ such that $u = \alpha(\beta\alpha)^i$ which is clearly a θ-palindrome. □

4 Classification of the set of Watson–Crick palindromes

In this section we discuss the properties satisfied by the set of all θ-palindromes over a given alphabet. We show that for an antimorphic involution the set of all θ-palindromes is context-free (Proposition 7) but not regular (Lemma 9). We also prove several other properties of θ-palindromes. If θ is an antimorphic involution then both the set of all θ-palindromes and its complement are dense (Proposition 6). In fact, Lemma 11 gives the precise number of such θ-palindromes of length $2k$ for an antimorphic involution: m^k where m is the cardinality of the alphabet. This implies that, in the case of the DNA alphabet and WK complementarity, there is a rich set of both WK-palindromic and WK-non-palindromic sequences to choose from. The situation is quite different in the case of a morphic involution, where the set of θ-palindromes is much smaller. Indeed, for a morphic involution θ over Σ, the set of all θ-palindromes equals Σ'^*, where $\Sigma' \subseteq \Sigma$ and $\theta(a) = a$ for all $a \in \Sigma'$ while $\theta(b) \neq b$ for all $b \in \Sigma \setminus \Sigma'$ (Corollary 1). In particular, if $\Sigma' = \emptyset$, the only θ-palindrome is the empty word (Lemma 10).

We recall the following definitions.

Definition 2 A language L is said to be:

1. θ-stable if $\theta(L) \subseteq L$.
2. Transitive if for all $x, y \in L$ there exists $z \in \Sigma^*$ such that $xzy \in L$.
3. Prolongable if for all $x \in L$ there exist $p, q \in \Sigma^+$ such that $pxq \in L$.
4. Dense if for all $u \in \Sigma^*$, $L \cap \Sigma^* u \Sigma^* \neq \emptyset$.

Given a finite alphabet set Σ and let θ be either a morphic or an antimorphic involution on Σ^*. In the next propositions we show that the set of all θ palindromes is θ-stable for both morphic and antimorphic involutions θ. We denote by P_θ the set of all θ-palindromes and by $\bar{P}_\theta$ the set of all non θ-palindromes.

Lemma 7 *Let θ be a morphic or an antimorphic involution. Then both P_θ and $\bar{P}_\theta$ are θ-stable.*

Proof Let P_θ be the set of all θ-palindromes and then for all $w \in P_\theta$, $\theta(w) = w \in P_\theta$. Thus P_θ is θ-stable and also $w \in \bar{P}_\theta$ iff $\theta(w) \neq w$ iff $\theta(w) \in \bar{P}_\theta$ and hence $\bar{P}_\theta$ is θ-stable. □

Proposition 6 *Let θ be an antimorphic involution. Then both*

1. *P_θ and $\bar{P}_\theta$ are dense.*
2. *P_θ and $\bar{P}_\theta$ are prolongable.*

Proof

1. In order to show that P_θ is dense we need to show that for all $u \in \Sigma^*$ there exist $x, y \in \Sigma^*$ such that $xuy \in P_\theta$. If $u \in P_\theta$ then for $x = y = \lambda$, $xuy \in P_\theta$ and similarly if $u \in \bar{P}_\theta$ then for $x = \lambda$ and $y = \theta(u)$ or $y = \lambda$ and $x = \theta(u)$, $xuy \in P_\theta$.
2. For every $w \in P_\theta$, $w = \theta(w)$. For all $a \in \Sigma$, $aw\theta(a) \in P_\theta$ since $\theta(aw\theta(a)) = a\theta(w)\theta(a) = aw\theta(a)$. For every $w \in \bar{P}_\theta$, $w \neq \theta(w)$ and for all $a, b \in \Sigma$, $awb \notin P_\theta$ since $\theta(awb) = \theta(b)\theta(w)\theta(a) \neq awb$ since $w \neq \theta(w)$. □

In the following Lemma we prove a relation between the set of all non θ-palindromes and θ-unbordered words.

Lemma 8 *Let θ be an antimorphic involution and let Σ be such that for all $a \in \Sigma$, $\theta(a) \neq a$. Then the set of all θ-palindromes P_θ is a proper subset of the set of all θ-bordered words B_θ.*

Proof Let $w \in P_\theta$. Note that $w \neq a$ for all $a \in \Sigma$ since $a \neq \theta(a)$. Since $w = \theta(w)$ we have $w = ax\theta(a)$ for some $x \in P_\theta$ which clearly implies that $w \in B_\theta$. □

We recall the following definition from Kari et al. (2007).

Definition 3 Let θ be either a morphic or an antimorphic involution. A word $u \in \Sigma^*$ is said to be an *(θ, k)-hairpin-free* if $u = xvy\theta(v)z$ or $u = x\theta(v)yvz$ where $x, v, y, z \in \Sigma^*$ implies $|v| < k$.

We denote by *hpf*(θ, k) the set of all (θ, k)-hairpin-free words in Σ^* and note that when $k = 1$ we obtain the set of all hairpin-free words over Σ^*. It was shown in Kari et al. (2007) that the set of all hairpin-free words is closed under insertion. Note that the set of all involution palindromes is a subset of the set of all hairpin-free words. The set of all θ-palindromes is not closed under insertion, i.e., for all $u = u_1u_2 \in P_\theta$, there exists $w \in \Sigma^*$ such that $u_1wu_2 \notin P_\theta$. Note that it was shown in Kari and Mahalingam (2007c) that B_θ, the set of all θ-bordered words, is a proper subset of the set of all hairpin-free words and hence

P_θ is a proper subset of the set of all hairpin-free words. In Kari and Mahalingam (2007a), it was shown that for an antimorphic involution θ, the set of all θ-bordered words is regular. We show using pumping lemma for regular languages that the set of all θ-palindromes is not regular.

Lemma 9 *When θ is an antimorphic involution, the set of all θ-palindrome words is not regular.*

Proof Let θ be an antimorphic involution. Since θ is not the identity function and $|\Sigma| \geq 2$, there exist $a, b \in \Sigma$ such that $a \neq b$, $\theta(a) = b$ and $\theta(b) = a$. Assume that the language P_θ of all θ-palindromes is regular and let n be the constant given by the pumping lemma. Choose $w = a^n b^n$ and note that $w = \theta(w)$ and hence w is a θ-palindrome. Let $w = a^n b^n = xvy$ such that $|xv| \leq n$ and $|v| > 0$. Then $z = xv^i y$ contains more a's than b's for all $i \geq 2$ and hence z is not a θ-palindrome. Thus P_θ is not regular. □

In the following proposition we construct a context-free grammar that generates the set of all θ-palindromes over a finite alphabet set for an antimorphic involution θ.

Proposition 7 *For an antimorphic involution θ, the set P_θ is context-free.*

Proof Let Σ be a finite alphabet set and let $G = (\{X, Y\}, \Sigma, X, \mathcal{R})$ where $\mathcal{R} = \{X \to \lambda$, $Y \to \lambda$, $X \to a_i X \theta(a_i)$ for all $a_i \in \Sigma$ and $X \to b_i Y b_i$, $Y \to b_i Y b_i$ for all $b_i \in \Sigma$ such that $b_i = \theta(b_i)\}$. It is easy to check that G generates the set of all θ-palindromes over Σ and G is context-free. □

In the next lemma we observe that for a morphic involution θ which is not identity for all letters in Σ, a θ-palindrome must be of even length.

Lemma 10 *Let Σ be such that for all $a \in \Sigma$, $\theta(a) \neq a$.*

1. *When θ is a morphic involution, then $P_\theta = \{\lambda\}$.*
2. *When θ is an antimorphic involution, then for all $u \in P_\theta$, the length of u is an even number.*

Proof

1. Let $u = a_1 a_2 \ldots a_n \in P_\theta$ and θ be a morphic involution. Then $u = a_1 a_2 \ldots a_n =$ - $\theta(a_1)\theta(a_2) \ldots \theta(a_n)$ which implies that $\theta(a_i) = a_i$ for all $1 \leq i \leq n$ a contradiction to our assumption. Hence $u = \lambda$.
2. Let u be a θ-palindrome and hence $u = \theta(u)$. Let $u = a_1 a_2 \ldots a_n$ for some $a_i \in \Sigma$. Then $u = a_1 a_2 \ldots a_n = \theta(a_n)\theta(a_{n-1}) \ldots \theta(a_1)$ and hence $a_i = \theta(a_{n-i+1})$ for all $1 \leq i \leq n$. Suppose $n = 2k + 1$, then for $i = k + 1$, $a_{k+1} = \theta(a_{n-i+1}) = \theta(a_{2k+1-k-1+1}) = \theta(a_{k+1})$ which is a contradiction. Thus n has to be even. □

Corollary 1 *Let θ be a morphic involution over an alphabet Σ. Then the set of all θ-palindromes, P_θ, is regular and equals Σ'^*, where $\Sigma' \subseteq \Sigma$ and $\theta(a) = a$ for all $a \in \Sigma'$, while $\theta(b) \neq b$ for all $b \in \Sigma \setminus \Sigma'$.*

Lemma 11 *Let θ be an antimorphic involution and let $P_\theta^{(n)}$ be the set of all θ-palindromes of length n. Let Σ be such that $|\Sigma| = m$ and let $\Sigma' \subseteq \Sigma$ be the maximal subset such that for all $a \in \Sigma'$, $a = \theta(a)$ and $|\Sigma'| = r$. Then,*

1. *when $n = 2k + 1$, $|P_\theta^{(n)}| = m^k r$.*
2. *when $n = 2k$, $|P_\theta^{(n)}| = m^k$.*

Proof Let $u \in P_\theta^{(n)}$. When $n = 2k + 1$, then $u = a_1 a_2 \ldots a_{2k+1} = \theta(a_1 a_2 \ldots a_{2k+1}) = \theta(a_{2k+1})\theta(a_{2k}) \ldots \theta(a_2)\theta(a_1)$. Thus $u = a_1 a_2 \ldots a_k\ a_{k+1}\theta(a_1 \ldots a_k)$ with $a_{k+1} = \theta(a_{k+1})$. Hence we have m choices for all the first k positions and r choices for the $k + 1$th position and only one choice for the remaining positions. Hence $|P_\theta^{(n)}| = m^k \times r = m^k r$. The argument is similar when $n = 2k$ and for all $u \in P_\theta^{(2k)}$, $u = a_1 a_2 \ldots a_k\ \theta(a_1 a_2 \ldots a_k)$ and hence we have m choices for the first k positions and only one choice for the remaining positions and thus $|P_\theta^{(n)}| = m^k$. □

Example 1 Let $\Sigma = \{a,b\}$ and let θ be an antimorphic involution such that $\theta(a) = b$ and $\theta(b) = a$. Note that $|\Sigma| = m = 2$. For $n = 4 = 2k$, we have $k = 2$ and the set of all θ-palindromes of length 4 is $P_\theta^{(4)} = \{abab, baba, bbaa, aabb\}$ and $|P_\theta^{(4)}| = 4 = m^k = 2^2$. The number of all non θ-palindromes of length 4 is $2^4 - 4 = 12$.

Example 2 Consider the DNA alphabet $\Delta = \{A,G,C,T\}$ and let θ be an antimorphic involution that maps $A \mapsto T$ and $C \mapsto G$. For $n = 4 = 2k$, we have $k = 2$ and the set of all θ-palindromes of length 4 is given by $P_\theta^{(4)} = \{AATT, ATAT, ACGT, AGCT, CATG, CTAG, CCGG, CGCG, GATC, GTAC, GCGC, GGCC, TATA, TTAA, TCGA, TGCA\}$. It is easy to check that $|P_\theta^{(4)}| = 16 = 4^2 = m^k$.

5 Simultaneous Watson–Crick conjugate equations

In this section we concentrate on simultaneous word equations especially involving words that are WK-conjugates. Even though we concentrate on the WK-involution, our results hold for a general involution mapping which can be either a morphism or an antimorphism. We observe that the solutions of such equations are nothing but a product of θ-palindromes.

In the following Proposition we solve a simultaneous equation concerning a word x such that x is θ-conjugate to its WK complement.

Proposition 8 *Let $x, y \in \Sigma^+$ such that $xy = \theta(y)\theta(x)$ and $x\theta(y) = y\theta(x)$.*

1. *If θ is a morphic involution, then $x = \alpha^m$ and $y = \alpha^n$ for some $\alpha \in P_\theta$.*
2. *If θ is an antimorphic involution, then $x = (\alpha\beta)^m$, $y = \alpha(\beta\,\alpha)^n$ with both $\alpha, \beta \in P_\theta$ and for some $m \geq 1$, $n \geq 0$.*

Proof

1. Let θ be a morphic involution. We first consider the case when $|x| < |y|$. The other case when $|y| \leq |x|$ is similar. Let $|x| < |y|$, then $xy = \theta(y)\theta(x)$ implies that $\theta(y) = xy_1$, $y = y_1\theta(x)$ and $x\theta(y) = y\theta(x)$ implies that $y = x\theta(y_1)$, $\theta(y) = \theta(y_1)\theta(x)$ for some $y_1 \in \Sigma^+$. Thus we can deduce that $x = \theta(x)$, $xy_1 = \theta(y_1)x$ and $x\theta(y_1) = y_1 x$. Then by Proposition 5, either $x = \alpha^i$, $y_1 = \alpha^j$ with $\alpha = \theta(\alpha)$ or $x = [\theta(\alpha)\alpha]^k\theta(\alpha)$, $y_1 = [\alpha\theta(\alpha)]^l$. If $x = [\theta(\alpha)\alpha]^k\theta(\alpha)$, then since $x = \theta(x)$ we deduce that $\alpha = \theta(\alpha)$ and hence $x = \alpha^i$ and $y = \alpha^j$ for some $\alpha \in P_\theta$.
2. Let θ be an antimorphic involution. We first consider the case when $|x| < |y|$. The other case when $|y| \leq |x|$ is similar. Let $|x| < |y|$, then $xy = \theta(y)\theta(x)$ implies that $\theta(y) = xy_1$, $y = y_1\theta(x)$ for some $y_1 \in \Sigma^+$ and $x\theta(y) = y\theta(x)$ implies that $y = x\theta(y'')$, $\theta(y) = \theta(y'')\theta(x)$ for some $y'' \in \Sigma^+$. Thus we can deduce that $y = x\theta(y'') = y_1\theta(x)$ and $y_1 = \theta(y_1)$, $y'' = \theta(y'')$. Let $|x| < |y_1|$, then we have $y_1 = xs_1 = \theta(s_1)\theta(x)$ and $y'' = x\theta(s_1) = s_1\theta(x)$ for some $s_1 \in \Sigma^+$. Hence $y = x^2\theta(s_1) = \theta(s_1)\theta(x)^2$. Note that

from applying Proposition 5 to $\theta(s_1)\theta(x^2) = x^2\theta(s_1)$, we can deduce that $\theta(s_1) \in P_\theta$. Thus $y_1 = s_1\theta(x) = xs_1$ and by Proposition 5 there exist $\alpha, \beta \in P_\theta$ such that $s_1 = \alpha(\beta\alpha)^i$ and $x = (\alpha\beta)^m$ for $m \geq 1$ and $i \geq 0$. Therefore $y = y_1\theta(x) = s_1\theta(x)(-x) = \alpha(\beta\alpha)^n$. If $|x| \geq |y_1|$, then $x = y_1\theta(x_2) = x_1x_2$ where $x_2 \in P_\theta$, $x_1 = y_1$. Also, $y'' = \theta(x_1) \in P_\theta$ which implies $x_1 \in P_\theta$. Hence $x = \alpha\beta$, $y = \alpha\beta\alpha$ where $x_1 = \alpha$, $x_2 = \beta$ and $\alpha,\beta \in P_\theta$. □

Example 3 Consider the DNA alphabet $\Delta = \{A,G,C,T\}$ and let θ be the Watson–Crick involution. Let $x = ATCG$, $y = ATCGAT \in P_\theta$ and $\theta(x) = CGAT$. Then we have $x\theta(y) = \theta(y)\theta(x)$ and $x\theta(y) = y\theta(x)$ with $x = \alpha\beta$ and $y = (\alpha\beta)\alpha$ for $\alpha = AT$ and $\beta = CG$.

The following corollary is similar to that of the above proposition (Proposition 5) and hence we omit the proof. Replacing x with $\theta(y)$ and y with $\theta(x)$ in Proposition 8 we obtain the following.

Corollary 2 *Let $x, y \in \Sigma^+$ such that $xy = \theta(y)\theta(x)$ and $\theta(x)y = \theta(y)x$.*

1. *If θ is a morphic involution then $x = \alpha^m$, $y = \alpha^n$ for some $\alpha \in P_\theta$.*
2. *If θ is an antimorphic involution then $x = \alpha(\beta\alpha)^n$, $y = (\beta\alpha)^m$, $\alpha, \beta \in P_\theta$ and $m \geq 1$, $n \geq 0$.*

Example 4 Consider the DNA alphabet $\Delta = \{A,G,C,T\}$ and let θ be the Watson–Crick involution. Let $x = ATCGAT \in P_\theta$, $y = CGAT$ and $\theta(y) = ATCG$. Then we have $xy = \theta(y)\theta(x)$ and $\theta(x)y = \theta(y)x = ATCGATCGAT$ with $\alpha = AT$, $\beta = CG$ and $x = \alpha(\beta\alpha)$, $y = \beta\alpha$.

Proposition 9 *Let $x, y \in \Sigma^+$ such that $xy = \theta(y)\theta(x)$ and $yx = \theta(x)\theta(y)$. Let θ be either a morphic or an antimorphic involution, then one of the following holds*:

1. *$x = p^m$, $y = p^n$ for $p \in P_\theta$ and m, $n \geq 1$.*
2. *$x = [\theta(p)p]^m\theta(p)$, $y = [p\theta(p)]^n p$, for $p \in \Sigma^+$ and m, $n \geq 0$.*

Proof Let θ be a morphic involution and let $xy = \theta(y)\theta(x)$, $yx = \theta(x)\theta(y)$. If $|x| < |y|$ then $\theta(y) = xy_1$, $y_2 = \theta(x)$ and hence $y = \theta(x)\theta(y_1) = y_1y_2$. Thus $y = y_2\theta(y_1) = y_1y_2$ which implies that y_2 θ-commutes with $\theta(y_1)$. Then by Proposition 5, we have one of the following:

$-y_1 = p^i$, $y_2 = p^m = x$ for $p \in P_\theta$.
$-y_1 = [p\theta(p)]^i$, $y_2 = [p\theta(p)]^m p = \theta(x)$ for some $p \in \Sigma^+$.

Thus either we have $x = p^m$ and $y = p^n$ for $p \in P_\theta$ or $x = [\theta(p)p]^m\theta(p)$ and $y = [p\theta(p)]^n p$, $p \in \Sigma^+$. The case when $|y| \leq |x|$ is similar.

Let θ be an antimorphic involution and let $xy = \theta(y)\theta(x)$, $yx = \theta(x)\theta(y)$. If $|x| < |y|$, then $xy = \theta(y)\theta(x)$ implies that there exists $y_1 \in \Sigma^+$ such that $\theta(y) = xy_1$ and $y_2 = \theta(x)$. Thus we can deduce that $y_1 \in P_\theta$ and $y = y_1\theta(x)$. Substituting this in $yx = \theta(x)\theta(y)$ we obtain $y_1y_2\theta(y_2) = y_2\theta(y_2)y_1$. Let $z = y_2\theta(y_2)$ then $zy_1 = y_1z$ and hence there exists $s \in \Sigma^+$ such that $z = s^i$ and $y_1 = s^j$. Note that $s \in P_\theta$ since $y_1 \in P_\theta$. We have $z = y_2\theta(y_2) = s^i$ and we have either $y_2 = s^{j_1}$, $\theta(y_2) = s^{j_1}$ or $y_2 = s^{i_1}s_1$, $\theta(y_2) = s_2s^{i_2}$ where $s = s_1s_2$. Therefore $y_2 = s^{i_1}s_1 = s^{i_2}\theta(s_2)$. Thus we have $i_1 = i_2$, $s_1 = \theta(s_2) = p$ and $y_1 = [p\theta(p)]^i$, $y_2 = [p\theta(p)]^m p$. Hence either $x = s^l$ and $y = s^m$ for $s \in P_\theta$ or $x = [\theta(p)p]^m\theta(p)$ and $y = [p\theta(p)]^n p$. The case when $|y| \leq |x|$ is similar. □

Example 5 Consider the DNA alphabet $\Delta = \{A,G,C,T\}$ and let θ be the Watson–Crick involution. Let $x = ACTGCAGTACTG$ and $y = CAGT$. Then we have $xy = \theta(y)\theta(x) = ACTGCAGTACTGCAGT$ and $yx = \theta(x)\theta(y) = CAGTACTGCAGTACTG$ with $p = CAGT$, $m = 1$, $n = 0$.

Replacing x with $\theta(x)$ and viceversa in Proposition 9 we obtain a similar result.

Corollary 3 *Let $x, y \in \Sigma^+$ such that $\theta(x)y = \theta(y)x$ and $x\theta(y) = y\theta(x)$. Let θ be either a morphic or an antimorphic involution, then one of the following holds*:

1. $x = p^m$, $y = p^n$ *for* $p \in P_\theta$ *and* $m, n \geq 1$.
2. $x = [p\theta(p)]^m p$, $y = [p\theta(p)]^n p$, *for* $p \in \Sigma^+$ *and* $m, n \geq 0$.

Lemma 12 *Let θ be either a morphic or an antimorphic involution and let $x, y \in \Sigma^+$. Then $xu = \theta(u)y$ and $x\theta(u) = uy$ iff $x^{2k+1}u = \theta(u)y^{2k+1}$ and $x^{2k}u = uy^{2k}$ for all $k \geq 0$.*

Proof Assume $xu = \theta(u)y$ and $x\theta(u) = uy$. Then $x^{2k+1}u = x^{2k} \cdot xu = x^{2k}\theta(u)y = x^{2k-1}uy^2 = \cdots = \theta(u)y^{2k+1}$. Similarly we can show that $x^{2k}u = uy^{2k}$. Conversely, let $x^{2k+1}u = \theta(u)y^{2k+1}$ and $x^{2k}u = uy^{2k}$. Then $x \cdot x^{2k+1}u = x \cdot \theta(u)y^{2k+1}$ and $x \cdot x^{2k+1} u = x^{2k+2}u = uy^{2k+2}$. Thus we have $x\theta(u)y^{2k+1} = uy \cdot y^{2k+1}$ and hence by length argument we have that $x\theta(u) = uy$. Substituting $k = 0$ in $x^{2k+1}u = \theta(u)y^{2k+1}$ we get $xu = \theta(u)y$. □

Proposition 10 *Let θ be either a morphic or an antimorphic involution and let $x, y, u \in \Sigma^+$ such that $xu = \theta(u)y$ and $x\theta(u) = uy$. Then $x = (\alpha\beta)^m$, $y = (\beta\alpha)^m$ and $u = (\alpha\beta)^n\alpha \in P_\theta$ for some $\alpha, \beta \in P_\theta$, $m \geq 1$ and $n \geq 0$.*

Proof If $|x| = |u|$, then $x = u = y = \theta(u)$.

Let θ be a morphic involution and suppose $|x| < |u|$, then $\theta(u) = xu_1 = \theta(u_1)y$ and $u = u_1y = x\theta(u_1)$ for some $u_1 \in \Sigma^+$. Thus we can deduce that $x = \theta(x)$ and $y = \theta(y)$. We have $u = x\theta(u_1) = u_1y$ and hence from Proposition 3 there exist $s, t \in \Sigma^*$ such that $x = st$, $y = ts$ and $u_1 = (st)^i s$ with $s, t \in P_\theta$ since $x, y \in P_\theta$. If $|x| > |u|$ then there exists $y_1 \in \Sigma^+$ such that $x = \theta(u)y_1$, $y = y_1u$ and $x = uy_1$, $y = y_1\theta(u)$ and we can deduce that $u \in P_\theta$. Thus the equation $xu = \theta(u)y$ becomes $xu = uy$ and from Proposition 1 we have $x = \alpha\beta$, $y = \beta\alpha$ and $u = (\alpha\beta)^n\alpha$. Since $u \in P_\theta$, both $\alpha, \beta \in P_\theta$.

Let θ be an antimorphic involution. If $|x| > |u|$ then we have $x = \theta(u)y_1 = uy_1$ and $y = y_1u = y_1\theta(u)$ and hence $u = \theta(u)$. Thus we can deduce, $xu = uy$ and hence from Proposition 1 we get $x = \alpha\beta$, $y = \beta\alpha$ and $u = (\alpha\beta)^n\alpha$. Suppose $|x| < |u|$, then $\theta(u) = xs = s_1y$ and $u = xs_1 = s\ y$ for some $s, s_1 \in \Sigma^+$. Thus we can deduce that $s_1 = \theta(s_1)$, $s = \theta(s)$ and $x = \theta(y)$ and hence we have $u = sy = s\theta(x) = xs_1$. Then from Proposition 3 either $s = \theta(s_1)$ or $s = pq$, $s_1 = q\theta(p)$ and $x = p$. If $s = \theta(s_1)$ then we have $s = s_1$ since $s_1 \in P_\theta$ and $u = s\theta(x) = xs$ and by Proposition 5, there exist $\alpha, \beta \in P_\theta$ such that $s = \alpha(\beta\alpha)^i$ and $x = (\alpha\beta)^j$ and hence $y = (\beta\alpha)^j$, $u = \alpha(\beta\alpha)^n$. If $s = pq$, $s_1 = q\theta(p)$ and $x = p$ holds, then we have $s = pq = \theta(q)\theta(p)$ and $s_1 = q\theta(p) = p\theta(q)$ and hence from Proposition 8 there exist $\alpha, \beta \in P_\theta$ such that $p = (\alpha\beta)^i$ and $q = \alpha(\beta\alpha)^j$. Then we have $x = (\alpha\beta)^i$, $y = \theta(x) = (\beta\alpha)^i$ and $u = sy = (\alpha\beta)^k\ \alpha$. □

Example 6 Consider the DNA alphabet $\Delta = \{A,G,C,T\}$ and let θ be the Watson–Crick involution. Let $x = ATCG$, $y = CGAT$ and $u = ATCGAT \in P_\theta$. Then we have $xu = \theta(u)y$ and $x\theta(u) = uy$ where $\alpha = AT$ and $\beta = CG$.

6 Properties of Watson–Crick palindromes

In this section we concentrate on several basic algebraic and closure properties of set of all θ-palindromes over a given alphabet Σ where θ is an antimorphic involution. In particular we concentrate on WK-palindromes. As we will see, for an antimorphic involution the set of θ-palindromes is not in general closed under catenation (Lemma 13 and related observations) nor under insertion (Lemma 15 and related observations). This would imply that, in the case of DNA, unwanted WK-palindromes can be easily disposed of by simple DNA manipulations. Lemma 19 provides a connection between θ-palindromes and primitive words in the case of an antimorphic involution. It turns out that a word u θ-commutes with v if and only if both v and its primitive root can be written as a product of two θ-palindromes. Finally, we show that for an antimorphic involution, any θ-palindrome that cannot be written as a product of two nonempty θ-palindromes must be primitive (Corollary 6).

Observe that the set of all WK-palindromes is not necessarily closed under concatenation. For example consider the DNA alphabet $\{A, C, G, T\}$ and let $u = ATAT$ and $v = CGCG$ with both $u, v \in P_\theta$ since $\theta(u) = ATAT = u$ and $\theta(v) = CGCG = v$. But $uv = ATATCGCG$ and $\theta(uv) = CGCGATAT \neq uv$ which implies that $uv \notin P_\theta$. In the following lemma we provide with necessary and sufficient condition for $uv \in P_\theta$ provided $u, v \in P_\theta$.

Lemma 13 *Let θ be an antimorphic involution and let $u, v \in P_\theta$. Then $uv \in P_\theta$ iff u and v are powers of a common palindromic word.*

Proof Assume that $uv \in P_\theta$. Then $uv = \theta(uv) = \theta(v)\theta(u) = vu$ which implies that u and v are powers of a common word, i.e., $u = s^i$ and $v = s^j$ with $s \in P_\theta$ since $u, v \in P_\theta$. The converse is straightforward. □

Lemma 14 *Let θ be an antimorphic involution and let $x \in \Sigma^+$ such that $x \in P_\theta$. Let $x = uv$ with $u, v \in \Sigma^+$. Then,*

1. *$u \in P_\theta$ iff $uv^k \in P_\theta$ for all $k \geq 2$.*
2. *$v \in P_\theta$ iff $u^k v \in P_\theta$ for all $k \geq 2$.*

Proof

1. Assume $u \in P_\theta$. We show that $uv^k \in P_\theta$ for all $k \geq 2$. Since $x = uv \in P_\theta$, $uv = \theta(uv) = \theta(v)\ \theta(u) = \theta(v)u$. Then by Proposition 5 we have $u = \alpha(\beta\alpha)^n$ and $v = (\beta\alpha)^m$ for $\alpha, \beta \in P_\theta$. Then we have $uv^k = \alpha(\beta\alpha)^n(\beta\alpha)^{mk} = \alpha(\beta\alpha)^i \in P_\theta$. Conversely, let $uv^k \in P_\theta$ for all $k \geq 2$. Given that $uv \in P_\theta$, then $uv^k = \theta(uv^k) = \theta(v^{k-1})$ $\theta(v)\theta(u) = \theta(v^{k-1})uv$. Thus we have $uv^{k-1} = \theta(v^{k-1})u$ and from Proposition 5 we have $u = \alpha(\beta\alpha)^n \in P_\theta$ since $\alpha, \beta \in P_\theta$.
2. Similar. □

Example 7 Consider the DNA alphabet $\Delta = \{A, G, C, T\}$ and let θ be the Watson−Crick involution. Let $u = ATCGAT$ and $v = CGAT$. Then for $x = uv = ATCGATCGAT$ we have both $x, u \in P_\theta$. Observe that $uv^k \in P_\theta$ for all $k \geq 0$ and $u^k v \notin P_\theta$ for all $k \geq 2$.

Corollary 4 *Let θ be an antimorphic involution and let $uv \in P_\theta$, then $u, v \in P_\theta$ iff $u^+v^+ \in P_\theta$.*

It was shown in Kari et al. (2007) that the set of all hairpin-free words is closed under insertion. Observe that neither the set of all θ-bordered words nor the set of all

θ-palindromes are closed under insertion. For example consider the DNA alphabet $\{A, G, C, T\}$ and let $u = ATAT = u_1u_2 \in P_\theta$ and let $w = CGA$. Then $u_1wu_2 = ACGATAT \notin P_\theta$.

The following lemma provides conditions under which the insertion into a θ-palindrome results in θ-palindromic words.

Lemma 15 *Let θ be an antimorphic involution and let $x, v, y \in \Sigma^+$ such that $xy \in P_\theta$. If $xvy \in P_\theta$ then v can be written as a product of two palindromes.*

Proof Given that $xy, xvy \in P_\theta$ and let $|x| = |y|$. Then $xvy = \theta(y)\theta(v)\theta(x)$ and $xy = \theta(y)\theta(x)$. Since $|x| = |y|$ we have $x = \theta(y)$ and $v = \theta(v)$. If $|x| < |y|$ such that $|y| \leq |xv|$ then $\theta(y) = xy_1$, $y_2 = \theta(x)$ where $y = y_1y_2$, which implies that $y_1 \in P_\theta$. Also, $xvy = \theta(y)\theta(v)\theta(x)$ implies that $\theta(y) = xv_1$, $\theta(v)\theta(x) = v_2y$ and hence $v_2 \in P_\theta$ and $y_1 = \theta(v_1) \in P_\theta$. Thus $v = v_1v_2$ with $v_1, v_2 \in P_\theta$. If $|x| < |y|$ such that $|y| > |xv|$ then $xy \in P_\theta$ implies that $\theta(y) = xy_1 = \theta(y_2)\theta(y_1)$, $y_2 = \theta(x)$ and $xvy \in P_\theta$ implies that $\theta(y) = xvy' = \theta(y'')\theta(y')$ and $y'' = \theta(v)\theta(x)$ with $y = y'y'' = y_1y_2$. Thus we have $y_1, y' \in P_\theta$ and $xy = xy'y'' = xy'\theta(v)\theta(x)$. Since $xy \in P_\theta$, $xy = xy'\theta(v)\theta(x) = xvy'\theta(x)$ which implies that $y'\theta(v) = vy'$. Then by Proposition 5 there exist $\alpha, \beta \in P_\theta$ such that $y' = \alpha(\beta\alpha)^i$ and $v = (\alpha\beta)^j = (\alpha\beta)^{j_1}\alpha \cdot \beta(\alpha\beta)^{j_2}$ with $(\alpha\beta)^{j_1}\alpha, \beta(\alpha\beta)^{j_2} \in P_\theta$. The case when $|y| < |x|$ is similar. □

The converse of the above Lemma does not hold in general. For example consider $xy = ATCGAT \in P_\theta$ and $v = ATCG$ such that $AT, CG \in P_\theta$. But $xvy = ATC \cdot ATCG \cdot GAT = ATCATCGGAT \notin P_\theta$ where $x = ATC$ and $y = GAT$.

Lemma 16 *Let θ be an antimorphic involution and let $x, y \in P_\theta$. If there exists a $z \in \Sigma^*$ such that $|xz| \geq |y|$, $|yz| \geq |x|$ and $xzy \in P_\theta$ then $x = \alpha(\beta\alpha)^i$, $y = \alpha(\beta\alpha)^j$ for some $i, j \geq 0$ with $\alpha, \beta \in P_\theta$.*

Proof Let $x, y, xzy \in P_\theta$. Then $xzy = \theta(xzy) = y\theta(z)x$. If $z = \lambda$ then we have $xzy = xy = yx$. Since $|xz| \geq |y|$ and $|yz| \geq |x|$, we have that $x = y$ and the statement of the Lemma holds.

Assume that $z \neq \lambda$. If $|x| < |y|$ then there exists $z_1 \in \Sigma^+$ such that $y = xz_1$, $z = z_1z_2$, $z_2y = \theta(z_2)\theta(z_1)x$. Thus we can deduce that $z_2 \in P_\theta$ and $y = xz_1 = \theta(z_1)x$. Thus by Proposition 5 there exist $\alpha, \beta \in P_\theta$ such that $x = \alpha(\beta\alpha)^i$ and $z_1 = (\beta\alpha)^j$ and hence $y = \alpha(\beta\alpha)^k$. If $|x| \geq |y|$ then there exists $z_2 \in \Sigma^*$ such that $x = y\theta(z_2)$, $z = z_1z_2$, $zy = \theta(z_1)x$. Thus we can deduce that $z_1 \in P_\theta$ and $x = z_2y = y\theta(z_2)$. Again using Proposition 5, we can find an $\alpha, \beta \in P_\theta$ such that $y = \alpha(\beta\alpha)^i$ and $\theta(z_2) = (\beta\alpha)^j$. Then we have $x = \alpha(\beta\alpha)^k$. □

The above lemma doesn't hold when $|xz| < |y|$ or $|yz| < |x|$. For example let $x = ATCGAT$, $y = ATCGATACGTATCGATCGATACGTATCGAT$ and $z = ACGTATCG$. Note that $x, y, xzy \in P_\theta$ and $|xz| < |y|$. But $x = \alpha(\beta\alpha)$ for $\alpha = AT$, $\beta = CG$ with $\alpha, \beta \in P_\theta$ and $y = [ATCGATACGTATCG\,]^2AT \neq \alpha(\beta\alpha)^i$ for all $i \geq 0$. Also $y \neq x\,(px)^j$ for $p \in P_\theta$.

In the following lemma we use some of the various simultaneous conjugate equations from Sect. 5 to show other properties of palindromic words.

Lemma 17 *Let θ be an antimorphic involution and let $u, v \in \Sigma^+$.*

1. *If $uv, \theta(u)v \in P_\theta$ then $u \in P_\theta$.*
2. *If $uv, u\theta(v) \in P_\theta$ then $v \in P_\theta$.*

Proof

1. Given $uv, \theta(u)v \in P_\theta$ then from Corollary 2 we have $u = \alpha(\beta\alpha)^n$, $\alpha, \beta \in P_\theta$ and hence $u \in P_\theta$.

2. Given $uv, u\theta(v) \in P_\theta$, then by Proposition 8 we have $v = \alpha(\beta\alpha)^n$ with $\alpha, \beta \in P_\theta$ and hence $v \in P_\theta$. □

Lemma 18 *Let θ be an antimorphic involution and let $u, v \in \Sigma^+$ such that $u \in P_\theta$ and either $uv \in P_\theta$ or $vu \in P_\theta$ then v is a product of two palindromes.*

Proof We have $uv = \theta(uv) = \theta(v)\theta(u) = \theta(v)u$ since $u \in P_\theta$. Thus u θ-commutes with v and by Proposition 5 we have $u = \alpha(\beta\alpha)^i$, $v = (\beta\alpha)^j$ with $\alpha, \beta \in P_\theta$. Thus $v = (\beta\alpha)^{i_1}\beta \cdot \alpha(\beta\alpha)^{i_2}$ with $i_1 + i_2 = j - 1$ and $(\beta\alpha)^{i_1}\beta, \alpha(\beta\alpha)^{i_2} \in P_\theta$.

Recall that a word $u \in \Sigma^+$ is called primitive if it is not a power of another word, i.e., there exists no word s such that $u = s^k$ for some $k > 1$. If $u = s^k$ for some $k \geq 2$, and s is minimal in length then we call s to be the primitive root of u. We show in the following Lemma that the primitive root of a non-palindromic word v can be written as a product of two Watson–Crick palindromes iff there exists another word that θ-commutes with v.

Lemma 19 *Let θ be an antimorphic involution and let $v \in \Sigma^+ \setminus P_\theta$. Then the primitive root of v (written as $\sqrt{v}$) is the product of two non-empty Watson–Crick palindromes iff there exists a non-empty $u \in P_\theta$ such that u θ-commutes with v.*

Proof Assume that there exists a $u \in P_\theta$ such that u θ-commutes with v. Then $uv = \theta(v)u$ and by Proposition 5 there exist $\alpha, \beta \in P_\theta$ such that $u = \beta(\alpha\beta)^j$ and $v = (\alpha\beta)^i$. Observe that α and β cannot be simultaneously empty since $u, v \in \Sigma^+$. If one of α or β is empty then $u = \alpha^i$ and $v = \alpha^j$ or $u = \beta^i$ and $v = \beta^j$. Both cases imply that $v \in P_\theta$ which is a contradiction to our assumption. Hence both $\alpha, \beta \in \Sigma^+$. Note that from Lemma 3, $\sqrt{v} = xy$ where x, y are the antimorphic twin-roots of v and thus non-empty θ-palindromes. Conversely, let $\sqrt{v} = \alpha\beta$ for some $\alpha, \beta \in P_\theta \cap \Sigma^+$. Then $v = (\alpha\beta)^i$ for some $i \geq 1$ and hence for $u = \beta(\alpha\beta)^j$ for some $j \geq 0$ we have $uv = \beta(\alpha\beta)^j(\alpha\beta)^i = \theta(v)u$. □

Lemma 20 *Let θ be an antimorphic involution and let $u = xy$ be a primitive word such that $x, y \in P_\theta \cap \Sigma^+$. Then $u \notin P_\theta$ and the factorization of u such that u is a product of two non empty θ-palindromes is unique.*

Proof Given $u = xy$ with $x, y \in P_\theta \cap \Sigma^+$. Suppose $u \in P_\theta$ then $u = xy = \theta(u) = \theta(y)\theta(x) = yx$ which implies that $x = s^i$ and $y = s^j$ for some $s \in \Sigma^+$ and $i, j \geq 1$. Note that $s \in P_\theta$ since both $x, y \in P_\theta$. Thus we have $u = s^{i+j}$ a contradiction since u is primitive. Hence $u \notin P_\theta$.

Suppose the factorization of $u = xy$ is not unique. Then there exist $\alpha, \beta \in P_\theta \cap \Sigma^+$ such that $u = xy = \alpha\beta$. If $|y| < |\beta|$, then there exists $s \in \Sigma^+$ such that $\beta = sy = \theta(y)\theta(s) = y\theta(s) = \theta(\beta)$ and $x = \alpha s = \theta(s)\theta(\alpha) = \theta(s)\alpha = \theta(x)$. Thus $u = xy = \alpha sy = \theta(s)\alpha y = \alpha\beta = \alpha y\theta(s)$ and αy commutes with $\theta(s)$. Thus there exists $r \in \Sigma^+$ such that $\alpha y = r^i$ and $\theta(s) = r^j$. Hence $u = r^{i+j}$ a contradiction since u is primitive. Thus we have $\alpha = x$ and $\beta = y$. Thus the factorization of $u = xy$ is unique. □

In the next result we show that a word u which is not a WK-palindrome can be written as a product of two WK-palindromes iff the primitive root of u can also be written as a product of two WK-palindromes.

Lemma 21 *Let θ be an antimorphic involution. A non θ-palindrome u is a product of two θ-palindromes p, q if $\sqrt{u}$ is a product of two θ-palindromes.*

Proof Let $u = pq$ with $p, q \in P_\theta \cap \Sigma^+$ and $u \notin P_\theta$. Then by Proposition 4 and (Kari–Mahalingam–Seki) there uniquely exist $x, y \in P_\theta$ such that $u = (xy)^n$ and $\sqrt{u} = xy$. Conversely, if $\sqrt{u} = \alpha\beta$, then $u = (\alpha\beta)^n = \alpha\beta(\alpha\beta)^{n-1}$ with $\alpha, \beta(\alpha\beta)^{n-1} \in P_\theta$. □

Note that not all non-θ-palindromes can be written as a product of two θ-palindromes. For example consider the words *aba* and *abaa* over the alphabet set $\{a, b\}$. Let θ be an antimorphic involution that maps a to b and viceversa. Then both $aba, abaa \notin P_\theta$. Also $aba \neq pq$ and $abaa \neq st$ for all $p, q, s, t \in P_\theta$. Based on the previous results we have the following observation.

Corollary 5 *Let $u \in \Sigma^+$ such that $u \notin P_\theta$. Then the following are equivalent.*

1. *$\sqrt{u}$ is the product of two non-empty WK-palindromes.*
2. *There exists $v \in P_\theta \cap \Sigma^+$ such that v θ-commutes with u.*
3. *u is a product of two non-empty WK-palindromes.*

Let u be a θ-conjugate of w. In the following lemma we find necessary and sufficient conditions under which $w \in P_\theta$ whenever $u \in P_\theta$. In order to prove the following Lemma we use the result in Proposition 8.

Lemma 22 *Let θ be an antimorphic involution and let u be a θ-conjugate of w and let $u \in P_\theta$. Then $w \in P_\theta$ iff $u = w$.*

Proof Let u be a θ-conjugate of w and let $u \in P_\theta$. Assume that $w \in P_\theta$. Since θ is an antimorphism, by Proposition 3, we have either $u = \theta(w)$ or $u = xy$ and $w = y\theta(x)$. The case when $u = \theta(w)$ implies that $u = w$ since $w \in P_\theta$. Assume that $u = xy$ and $w = y\theta(x)$. Since both $u, w \in P_\theta$ we have that $u = xy = \theta(y)\theta(x)$ and $w = y\theta(x) = x\theta(y)$ and by Proposition 8 we have that $u = \alpha(\beta\alpha)^n$ and $w = \alpha(\beta\alpha)^n$ and hence $u = w$. The converse is straightforward. □

Lemma 23 *Let θ be either a morphic or an antimorphic involution and let $u, w \in \Sigma^+$ such that $u \in P_\theta$ and $w = xu = uy$ for some $x, y \in \Sigma^+$ then either $w \in P_\theta$ or $w \in B_\theta$ such that $w = pqp$ with $p \in P_\theta$.*

Proof Given that $w = xu = uy$ for some $x, y \in \Sigma^+$. Then by Proposition 1 there exist $p, q \in \Sigma^+$ such that $x = pq$, $y = qp$ and $u = p(qp)^i$ for some $i \geq 0$. If $|u| \geq |x|$ then $i \geq 1$ and since $u \in P_\theta$ we have both $p, q \in P_\theta$. Hence $w = xu = p(qp)^{i+1} \in P_\theta$. If $|u| < |x|$, then $i = 0$ and $u = p$. Since $u \in P_\theta$ we have $p \in P_\theta$ and $w = pqp$ with $p \in P_\theta$. □

Lemma 24 *Let θ be an antimorphic involution and let $P_{\theta_+} = P_\theta \setminus \{\lambda\}$. Then $P^2_{\theta_+} \cap P_{\theta_+} = \{s^i | i \geq 2; s \in P_{\theta_+}\}$.*

Proof Let $x \in P^2_{\theta_+} \cap P_{\theta_+}$. Then $x = \alpha\beta$ with $\alpha, \beta, x \in P_{\theta_+}$ and hence $\alpha\beta = \theta(\beta)\theta(\alpha) = \beta\alpha$ which implies that $\alpha = s^i$ and $\beta = s^j$ for $s \in P_{\theta_+}$. Thus $x = s^k$ for $k \geq 2$ and $s \in P_{\theta_+}$ and hence $P^2_{\theta_+} \cap P_{\theta_+} \subseteq \{s^i | i \geq 2; s \in P_{\theta_+}\}$. Conversely, let $x = s^i$ with $i \geq 2$, $s \in P_{\theta_+}$. Then for $\alpha = s^m$ and $\beta = s^n$, $m + n = i$ and $m, n \geq 1$, we have both $\alpha, \beta \in P_\theta$ and $x \in P_{\theta_+}$. Thus $x \in P^2_{\theta_+} \cap P_{\theta_+}$. Thus we have $P^2_{\theta_+} \cap P_{\theta_+} = \{s^i, i \geq 2 : s \in P_{\theta_+}\}$. □

Let Q denote the set of all primitive words.

Corollary 6 *Let θ be an antimorphic involution. Then $P_\theta \cap Q = P_\theta \backslash P^2_\theta$.*

7 Conclusions

In this paper we gave an overview of the existing approaches to the problem of finding optimal DNA encodings for biocomputations and focussed on the study, from an algebraic perspective, of a specific concept: the Watson–Crick palindrome. We obtained several properties of Watson–Crick palindromes that are relevant from a biocomputational perspective, such as the fact that both the set of WK-palindromes and the set of non-WK-palindromes are dense, and the fact that a WK-palindrome can in general be easily changed into a non-WK-palindrome by simple operations such as catenation and insertion.

In addition, we obtained some properties that link the WK palindromes to classical notions such as that of primitive words. For example we showed that, for an antimorphic involution, the set of θ-palindromic words that cannot be written as the product of two nonempty θ-palindromes equals the set of primitive θ-palindromes.

Future work includes the study of more complex systems of Watson–Crick equations, such as the ones studied in this paper and that resulted in WK-palindromic solutions, as well as the investigation of other extensions of classical notions in combinatorics of words such as a generalized notion of θ-primitivity.

Acknowledgements Research supported by Natural Sciences and Engineering Research Council of Canada Discovery Grant and Canada Research Chair Award for Lila Kari.

References

Adleman L (2000) Towards a mathematical theory of self-assembly. Technical Report 00-722, Department of Computer Science, University of Southern California

Daley M, McQuillan I (2006) On computational properties of template-guided DNA recombination. In: Carbone A, Pierce N (eds) Proceedings of the DNA computing 11. LNCS, vol 3892. Springer, Berlin, pp 27–37

de Luca A (2006) Pseudopalindrome closure operators in free monoids. Theor Comput Sci 362:282–300

Domaratzki M (2006) Hairpin structures defined by DNA trajectories. In: Mao C, Yokomori T (eds) Proceedings of the DNA computing 12. LNCS, vol 4287. Springer, Berlin, pp 182–194

Feldkamp U, Banzhaf W, Rauhe H (2000) A DNA sequence compiler. In: Condon A, Rozenberg G (eds) Pre-proceedings of the DNA-based computers 6. Leiden, Netherlands

Feldkamp U, Saghafi S, Banzhaf W, Rauhe H (2001) DNA sequence generator: a program for the construction of DNA sequences. In: Jonoska N, Seeman N (eds) Proceedings of the DNA-based computers 7. LNCS, vol 2340. Springeer, Berlin, pp 23–32

Garzon MH, Oehman C (2001) Biomolecular computation in virtual test tubes. In: Jonoska N, Seeman N (eds) Proceedings of the DNA-based computers 7. LNCS, vol 2340. Springer, Berlin, pp 117–128

Garzon M, Phan V, Roy S, Neel A (2006) In search of optimal codes for DNA computing. In: Mao C, Yokomori T (eds) Proceedings of the DNA computing 12. LNCS, vol 4287. Springer, Berlin, 143–156

Hartemik J, Gifford DK, Khodor J (1999) Automated constaint-based nucleotide sequence selection for DNA computation. In: Kari L, Rubin H, Wood D (eds) Proceedings of the DNA based computers 4. Biosystems 52(1–3):227–235

Hartemink J, Gifford DK (1999) Thermodynamic simulation of deoxyoligonucleotide hybridization for DNA computation. In: Rubin H, Wood D (eds) Proceedings of the DNA-based computers 3. DIMACS series in discrete mathematics and theoretical computer science. AMS Press, Providence, pp 25–38

Hopcroft J, Ullman J, Motwani R (2001) Introduction to automata theory, languages and computation, 2nd edn. Addison Wesley, Boston

Hussini S, Kari L, Konstantinidis S (2003) Coding properties of DNA languages. Theor Comput Sci 290:1557–1579

Jonoska N, Mahalingam K, Chen J (2005) Involution codes: with application to DNA coded languages. Nat Comput 4(2):141–162

Jonoska N, Kari L, Mahalingam K (2006) Involution solid and join codes. In: Ibarra O, Dang Z (eds) Developments in language theory: 10th international conference. LNCS, vol 4036. Springer, Berlin, pp 192–202

Kari L, Mahalingam K (2007a) Involutively bordered words. Int J Found Comput Sci 18:1089–1106
Kari L, Mahalingam K (2007b) Watson–Crick conjugate and commutative words. In: Garzon M, Yan H (eds) Preproceedings of the DNA computing 13. Springer, Berlin, pp 75–87
Kari L, Mahalingam K (2007c) Watson–Crick bordered words and their syntactic monoid. In: Domaratzki M, Salomaa K (eds) International workshop on language theory in biocomputing, Kingston, Canada, pp 64–75
Kari L, Konstantinidis S, Losseva E, Wozniak G (2003) Sticky-free and overhang-free DNA languages. Acta Inf 40:119–157
Kari L, Konstantinidis S, Losseva E, Sosik P, Thierrin G (2005a) Hairpin structures in DNA words. In: Carbone A, Pierce N (eds) Proceedings of the DNA computing 11. LNCS, vol 3892. Springer, Berlin, pp 158–170
Kari L, Konstantinidis S, Sosik P (2005b) Bond-free languages: formalizations, maximality and construction methods. Int J Found Comput Sci 16: 1039–1070
Kari L, Mahalingam K, Thierrin G (2007) The syntactic monoid of hairpin-free languages. Acta Inf 44(3):153–166
Kari L, Mahalingam K, Seki S (2009) Twin-roots of words and their properties. Theor Comput Sci 410 (24–25):2393–2400
Lothaire M (1997) Combinatorics of words. Cambridge University Press, Cambridge
Lyndon RC, Schutzenberger MP (1962) On the equation $a^M = b^N c^P$ in a free group. Mich Math J 9:289–298
Marathe A, Condon A, Corn R (1999) On combinatorial DNA word design. In: Winfree E, Gifford D (eds) Proceedings of the DNA based computers 5. DIMACS series in discrete mathematics and theoretical computer science. AMS Press, Providence, pp 75–89
Shyr HJ (2001) Free monoids and languages. Hon Min Book Company, Taiwan
Soloveichik D, Winfree E (2006) Complexity of compact proofreading for self-assembled patterns. In: Carbone A, Pierce N (eds) Proceedings of the DNA computing 11. LNCS, vol 3892. Springer, Berlin, pp 305–324
Tulpan D, Hoos H, Condon A (2003) Stochastic local search algorithms for DNA word design. In: Hagiya M, Ohuchi A (eds) Proceedings of the DNA-based computers 8. LNCS, vol 2568. Springer, Berlin, pp 229–241
Yu SS (1998) d-minimal languages. Discret Appl Math 89:243–262
Yu SS (2005) Languages and codes. Lecture notes. Department of Computer Science, National Chung-Hsing University, Taichung