

Isothermal reactivating Whiplash PCR for locally programmable molecular computation

John H. Reif · Urmi Majumder

© Springer Science+Business Media B.V. 2009

Abstract Whiplash PCR (WPCR; Hagiya et al., in Rubin H, Woods DH (eds) DNA based computers, vol III, pp 55–72. American Mathematical Society, Providence, RI, 1999) is a novel technique for autonomous molecular computation where a state machine is implemented with a single stranded DNA molecule and state transition is driven by polymerase and thermal cycles. The primary difference between WPCR computation and other forms of molecular computing is that the former is based on local, rather than global rules. This allows many (potentially distinct) WPCR machines to run in parallel. However, since each state transition requires a thermal cycle, multi-step WPCR machines are laborious and time-consuming, effectively limiting program execution to only a few steps. To date, no WPCR protocol has been developed which is *both* autocatalytic (self-executing) and isothermal (with no change in temperature). In this paper, we describe some isothermal and autocatalytic protocols that use a combination of strand displacement and DNA polymerization events. Our designs include (1) a protocol where *transition rules cannot be reused in subsequent computing* (2) a protocol where *rules can be reused using an auxiliary strand displacement event but does not prevent back-hybridization* (an event responsible for limiting the program execution to only a few state transitions before the machine stalls), (3) a *reusable rule protocol that prevents back-hybridization*. Furthermore, we show that the third machine which gets rid of thermal cycles and still prevents back-hybridization, is computationally equivalent to the original WPCR machine. We also compute the state transition likelihood and the corresponding rate in this protocol. Finally we present a DNA sequence design of a 3-state isothermal and reactivating WPCR machine along with an experimental verification plan.

Keywords Autonomous molecular computation · Finite state automata · DNA self-assembly · Strand displacement · DNA polymerization ·

J. H. Reif · U. Majumder (✉)
Department of Computer Science, Duke University, Durham, NC, USA
e-mail: urmim@cs.duke.edu

J. H. Reif
e-mail: reif@cs.duke.edu

Programmable molecular machines · Polymerase chain reaction · Isothermal computing · Autocatalytic biomolecular computers · State transition

Abbreviations

PCR	Polymerase chain reaction
WPCR	Whiplash polymerase chain reaction
DNA	Deoxyribo nucleic acid
ds-DNA	Double stranded deoxyribo nucleic acid
PNA	Peptide nucleic acid
IR-WPCR	Isothermal reactivating Whiplash polymerase chain reaction
FRET	Fluorescence resonance energy transfer
bis-PNA	Bi-specific peptide nucleic acid

1 Introduction

1.1 Need for an autocatalytic and isothermal WPCR protocol

A primary challenge in nanoscience is the design of synthetic molecular devices that run autonomously (meaning that the program executes without any external mediation over multiple work cycles) and programmable (meaning that the machine's behavior can be modified without completely redesigning the system). In the last few years, the idea of constructing complex devices at the molecular scale using synthetic materials, such as synthetic DNA, has gone from theoretical concept to experimental reality. One such autonomous molecular computing device is called a Whiplash PCR (WPCR) machine (Hagiya et al. 1999), equivalent to a restricted class of finite automata (namely those that can be represented as a directed acyclic graph). In this machine, the current state is encoded at the 3' end of a DNA single strand while the remainder of the strand encodes the state transition rules. The original machine works as follows: using appropriate thermal cycles, the current state anneals to the correct transition rule and, next, a polymerase extends the 3' end of the strand to copy the next state from the encoded transition rule. The only limitation of this system is that it can execute only a single step before it requires significant environmental changes (thermal cycles) to continue the computation. This paper presents a protocol that converts a WPCR system into an autonomous computing device, thus eliminating the need for external mediation to enable further steps. This capability is important if we wish to use these machines outside fully equipped laboratories.

1.2 Importance of locally programmable molecular computation

Although existing autonomous molecular computing devices e.g., DNA based tiling assembly (Winfree et al. 1998) and restriction enzyme based automata (Benenson et al. 2001) are computationally quite powerful (Soloveichik and Winfree 2005), they are not capable of executing distinct programs in parallel. In contrast, many complex molecular mechanisms found in the cell are more flexible and can perform a diverse set of tasks simultaneously. Whiplash PCR (Matsuda and Yamamura 2002; Nishikawa and Hagiya 1999; Rose et al. 2001; Winfree 1998b; Hagiya et al. 1999; Sakamoto et al. 1999) allows

parallel execution because each machine holds its own program and thus multiple, distinct molecular programs can run simultaneously in the same reaction tube.

1.3 Engineering DNA for biomolecular computation

Before we discuss how DNA can be the building block of a WPCR machine, we must introduce some of the primary methods by which DNA can be engineered and used in the correct execution of a WPCR machine. The three primary techniques are hybridization, strand displacement and polymerization.

DNA Hybridization is the process of combining two complementary, single-stranded DNA molecules into double-stranded molecule. DNA (and RNA) can bind to their complements under normal conditions or by when the strand mixture is cooled. Simple hybridization has been used to make a large variety of complex structures such as tiles and lattices (Mathieu et al. 2005; Shih et al. 2004; Goodman et al. 2004). The opposite of hybridization is *dehybridization*. The latter can be achieved by heating the solution mixture.

Polymerized chain reaction (PCR) is a common method for amplifying a target DNA molecule, which can be a single gene or merely part of it (Fig. 2). Through PCR, a small amount of DNA can be amplified several times and the richness and the fidelity of the product facilitates many applications, such as detecting hereditary diseases, identifying bacterial species, cloning genes, DNA computing and many others. A recent paper used DNA polymerase $\phi 29$ to power a nano-transport device (Sahu et al. 2008).

Branch migration or *strand displacement* is the process by which a single, invading DNA strand extends its partial pairing with its complementary strand as it displaces a resident strand from a DNA duplex (Fig. 2).

1.4 Previous methods for WPCR computing devices and their limitations

Hagiya et al. (1999) first proposed and experimentally demonstrated (only for a limited number of steps) a WPCR machine. It was not easy to increase the number of steps because of a phenomenon called *back-annealing* (also known as back-hybridization), where a hairpin with a longer double stranded (ds) DNA region is preferentially formed over one with a shorter ds-DNA region. Sakamoto et al. (1999) suggested a modified protocol where successive transitions were carried on at the same temperature, thus preventing back-hybridization. They also proposed a protocol for preventing *out-of-frame annealing* which happens because portions of two adjacent sequences constitute the sequences complementary to a third state sequence. However, this work did not significantly increase the number of steps the WPCR machine could execute before stalling at an intermediate step. Later, Rose et al. (2001) proposed a scheme using targeted *PNA₂/DNA* triplex formation. This triplex region destabilized the hairpin structure, thus preventing back-hybridization. Although this protocol can be isothermal, it is not autocatalytic. Further, it has been not yet been implemented and its role in increasing the number of steps a WPCR machine can execute before stalling is not clear. Two recent papers by Rose et al. (2006; Komiya et al. 2008) proposed and experimentally demonstrated isothermal (no thermal cycling required) conditions for the functioning of a WPCR machine using strand displacement techniques. However, one still needs to add a rule protection strand after each polymerization step to drive the computation forward. Hence if we wish to overcome this problem and also allow flexibility of applications, we need to design an autocatalytic (a system that reactivates itself) and isothermal (no thermal cycles required) protocol for WPCR that would allow the

machine to compute for a reasonable number of steps before succumbing to back-hybridization in places other than a laboratory.

1.5 Our contribution

Our main contribution in this paper is the design of a WPCR machine that eliminates the need for any other external mediation and, thus, the resultant machine is *isothermal*, *autocatalytic* and capable of preventing back-hybridization. By isothermal we mean that the temperature of the reaction mixture is constant throughout computation. However, our design requires an additional preparation stage which precedes the computation stage and this step is not isothermal. This protocol is also in contrast with the definition of isothermal by Sakamoto et al. (1999) who defined isothermal as a design where the denaturation of the previous state and annealing of the next state occur at the same temperature. This design still needs external thermal control for multiple state transitions. By autocatalytic, we mean that, once the protein enzyme is introduced in the solution, it drives computing on its own. This protocol can again be contrasted against the protocol for PNA-mediated WPCR (Rose et al. 2001) where, after each polymerization step, the mixture needs to be washed by bis-PNA. Recall that back-hybridization is an event where the WPCR machine stalls because the 3' end of the strand binds to the previous transition rule in preference to the new transition rule (also the correct one) since the former is energetically more stable. Several protocols have been proposed (Sakamoto et al. 1999; Rose et al. 2001, 2006; Komiya et al. 2008) to prevent back-hybridization. However, to the best of our knowledge, there exists no WPCR protocol that is both isothermal and autocatalytic and capable of preventing back-hybridization simultaneously. Our isothermal and reactivating protocol is partially inspired by our previous work where we used a combination of DNA polymerization and strand displacement events to minimize errors in computational tile assembly (Majumder et al. 2008).

The key idea in our design of an isothermal, reactivating WPCR (IR-WPCR) machine is to use primer extension of a secondary strand to dehybridize the 3' end of the WPCR strand after the next state is copied to it. Thus, the 3' end of the WPCR strand is now free to bind to the complement of the new current state. This action essentially eliminates the thermal cycling required by the original WPCR machine to execute a state transition. We give three versions of IR-WPCR: (1) a protocol where a state transition rule is no longer available for computation after the next state is copied (Sect. 3), (2) a protocol in which a rule can be made "reusable" by using an auxiliary strand that restores the original state of the secondary primer (Sect. 4) and finally (3) a protocol that allows reuse of transition rules while preventing back-hybridization (Sect. 5). The protocol presented in Sect. 5 is the most significant protocol in this paper: isothermal reactivating WPCR where the states are reusable and yet it is capable of preventing back-hybridization using a type of WPCR that we call *folding WPCR*. A primary assumption in all the above-mentioned protocols is that the concentration of the WPCR strand is such that two or more copies of WPCR strands do not interact among each other. We also prove that this machine is computationally equivalent to the original WPCR machine (Sect. 6). We further compute the state transition rate for an IR-WPCR machine and the probability of preventing back-hybridization at every step (Sect. 7). We also estimate the rate at which states are made reusable in the reusable rule protocol. Additionally, we present a DNA sequence design of a 3 state IR-WPCR machine and an experimental verification plan (Sect. 8). It should be remembered, however, that this example is meant to demonstrate the isothermal and autocatalytic aspects of the protocol and not the computational power of a WPCR device.

1.6 Organization of the paper

In Sect. 1, we briefly described a WPCR machine and its role as a autonomous and programmable molecular device. In particular, we discussed the advantages of the ability of a WPCR device to hold programs in the device itself. Furthermore, we discussed some of the previous approaches for realizing the concept of a WPCR machine and their shortcomings. In Sect. 2, we briefly describe the original Whiplash PCR machine and subsequently in Sect. 3 we present an isothermal and autocatalytic WPCR system. However, the protocol presented in Sect. 3 does not allow transition rules to take part in computation once the next state is copied from that site. Hence in Sect. 4 we discuss another protocol where we can make a transition rule reusable using an additional strand displacement operation. Yet this protocol is not suitable for preventing back-hybridization. Thus in Sect. 5 we describe a protocol for handling back-hybridization without change in temperature. Next in Sect. 6 we prove how an IR-WPCR machine capable of preventing back-hybridization can simulate a restricted class of finite state automata. In Sect. 7 we present a continuous time Markov Chain model for the working of an IR-WPCR machine and estimated the rate of state transition and the likelihood of preventing back-hybridization in each computational step. In Sect. 8 we propose a DNA design of a 3 state machine and describe how a reusable rule WPCR device can implement it. Finally, in Sect. 9 we conclude the paper with a summary of our contributions and future directions.

2 Original Whiplash PCR system

In the original WPCR machine, the transition table is encoded on a single stranded DNA, W as $S - a_1 - b_1 - S - a_2 - b_2 - \dots - S - a_n - b_n$ where each pair $a_i - b_i$ represents the transition from state a_i to state b_i . Here and in the rest of the paper, any symbol s encodes for a DNA sequence and s^* encodes for its complementary sequence. The stopper sequence S isolates one state transition rule from another. The 3' end of the same strand encodes the current state. For the description of the rest of the protocol, refer to Fig. 1. We represent the stopper sequence S as a black square in the figure. Without loss of generality (w.l.o.g.), let us assume that the current state of the machine is a_i^* . Following the transition table, a_i can transition to b_i . Once a_i^* hybridizes with a_i (Fig. 1: State S1) in W , polymerase extends the 3' end of W to copy b_i (Fig. 1: State S3). The polymerase halts after transcribing the bases complementary to b_i because of S which is generally implemented by emitting one of the bases in the solution. Using appropriate thermal cycling, W is then denatured. Consequently, it loses the hairpin structure (Fig. 1: State S4). Once the mixture is cooled, the 3' end of newly extended W (now bearing b_i^* as the current state) hybridizes with another section of itself which encodes the appropriate transition rule (in this rule $a_j = b_i$ is the current state and b_j is the next state; Fig. 1: State S5). Although input is not part of the description of the WPCR machine, it has been suggested that input be provided as part of the initial state and the encoding of the transition table updated to include inputs in the manner $S - a_i - I_i - b_i$ for the i th transition rule.

As mentioned in Sect. 1, back-hybridization is a serious limitation of the original Whiplash PCR system. Recall that back-hybridization refers to the phenomenon where a longer hairpin is preferentially formed over a shorter hairpin. As a result, the machine instead of binding to a rewrite rule whose current state matches with the new current state of the machine (also the next state of the previous rewrite rule) favorably binds to the

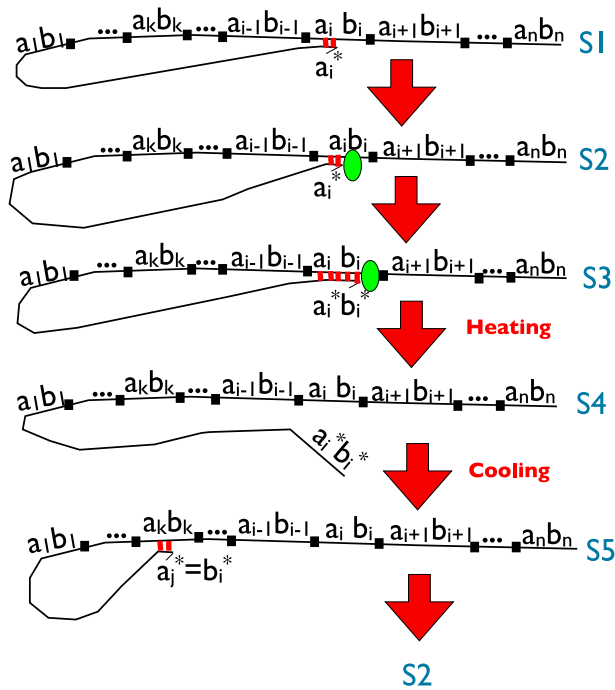


Fig. 1 Schematic of the protocol for the original Whiplash PCR machine: $S1$: initial state of the WPCR strand W with current state being a_i^* . $S2$: polymerase binds to the 3' end of W (bearing the current state). $S3$: next state b_i^* is copied at the head of W by primer extension. $S4$: the mixture is heated so that W loses its hairpin structure. $S5$: the solution is cooled so that the head of W can bind to the new current state $b_i^* = a_j^*$ encoded at the 3' end of the strand and the whole state transition repeats again beginning with State S2

rewrite rule from the previous step and, hence, it is stuck in this particular transition, since there are no new states to copy. This phenomenon has been illustrated in Fig. 4.

3 IR-WPCR machine with non-reusable rules

In IR-WPCR with non-reusable rules, computation comprises of the following steps after the 3' end of W binds to current state in rule R_i : (a) as with the original WPCR protocol, copying the next state at the 3' end of the WPCR strand W , (b) dislodging a secondary primer sequence P_i , which is specific to the transition rule R_i from its initial position triggered by the primer extension on W , (c) subsequent hybridization of P_i to its final position in rule R_i and (d) dislodging of 3' end of W by primer extension of P_i , allowing the 3' end of W to bind to the new transition rule. Observe that (b) and (d) act like a logical toggle switch allowing for an isothermal, autocatalytic reaction.

In this version of WPCR, each rule is encoded as a 7-tuple $\langle x_i, y_i, z_i, a_i, b_i, w_i, y_i \rangle$ where a_i still represents the current state and $b_i w_i y_i$ represents the next state where the b_i in IR-WPCR is not the same as b_i in the original WPCR strand. Rather, the original b_i is now divided into 3 subsequences b_i , w_i and y_i . The other regions in this tuple are required for destabilizing the 3' end of the strand once the next state is copied at the end of it. In this machine, the transition table with n rules is encoded on a single stranded DNA as

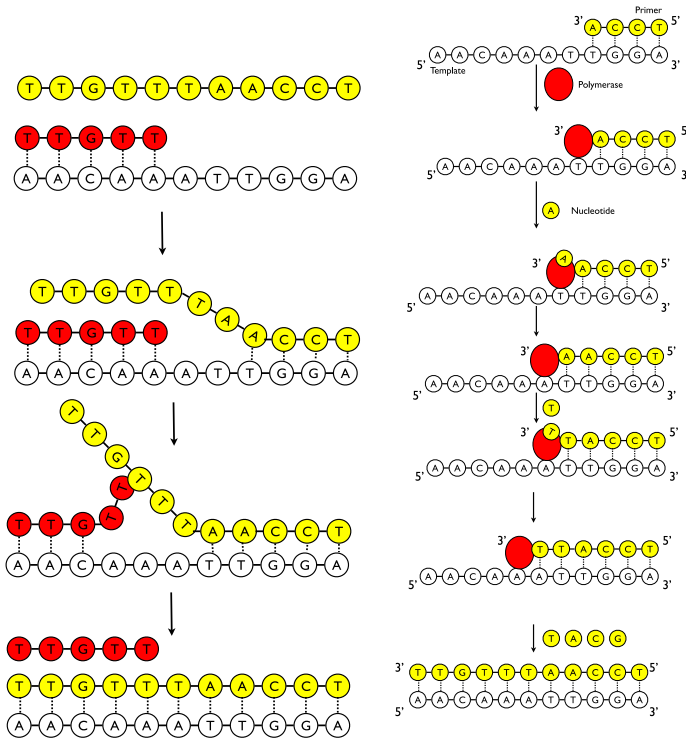


Fig. 2 *Left*: branch migration; *Right*: extension of primer strand bound to the template by DNA polymerase

$S - x_1 - y_1 - z_1 - a_1 - b_1 - w_1 - y_1 - \dots - S - x_i - y_i - z_i - a_i - b_i - w_i - y_i - \dots - S - x_n - y_n - z_n - a_n - b_n - w_n - y_n$. The 3' end of the single strand still encodes the current state as in original WPCR. We also tether the transition table portion of W to another stable nanostructure to prevent formation of any undesired secondary structure (Figs. 2, 3). The latter is mostly a double stranded DNA intercepted by sections of DNA that is bound to each transition rule. Since the rigidity of a double-stranded DNA is well known we use this particular nanostructure as a support in our designs.

3.1 Computing with a non-reusable rule IR-WPCR strand

Suppose we have the single strand in the form shown in Fig. 3 prior to the addition of polymerase. In Sect. 2, we will discuss how we can obtain this particular secondary structure. W.l.o.g we will assume that the 3' end of the single strand encodes for the complement of the current state a_i in rule R_i . For clarity, we will refer to a figure that focuses only on the events at R_i (Figs. 4, 5).

Once a_i^* binds to a_i in R_i (Fig. 5: State S1) in presence of polymerase (Fig. 5: State S2), the next state $b_i w_i y_i$ is copied at the 3' end of W , thus dehybridizing the $(w_i y_i)^*$ portion of the protection strand P_i encoded as $(x_i p_i w_i y_i)^*$ (Fig. 5: State S3). The y_i^* portion of P_i is now free to hybridize with the y_i portion on the rule R_i that is closer to x_i (Fig. 5: State S4). The 3' end of P_i , in presence of polymerase (Fig. 5: State S5), then extends up to the stopper sequence S (shown in black filled squares in the figure), thus displacing the 3' end

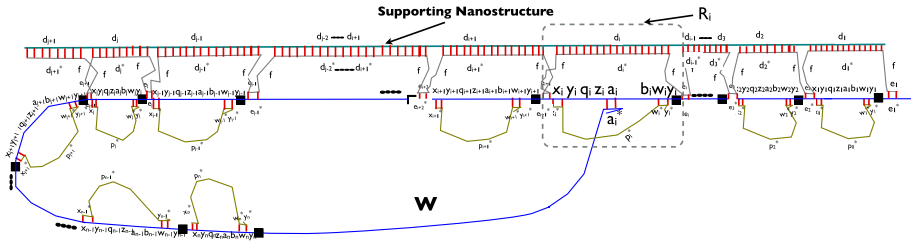


Fig. 3 Complete WPCR Strand for isothermal and autocatalytic program execution (Rule R_i on focus). Although details are provided in this figure, the emphasis is on the layout of the overall strand. In particular, note that most of the strand representing the transition rules is stabilized using a supporting DNA nanostructure and only the current state of the machine is allowed to freely bind to an appropriate rewrite rule using a lag region W

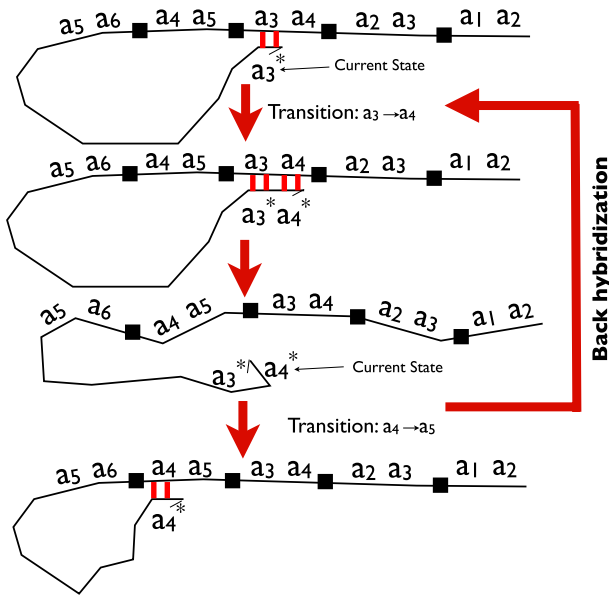


Fig. 4 Back-hybridization: transition from state a_3 to state a_4 happens as usual but for the next transition a_4 to a_5 , the 3' end of the machine preferentially binds with the old transition rule. This is because a_3^* along with a_4^* at the 3' end of the machine has a longer hybridization region when bound with rewrite rule $a_3 \rightarrow a_4$ compared to when only a_4^* binds with the current state of the rewrite rule $a_4 \rightarrow a_5$. Consequently, the machine is stuck in state a_4

of W (Fig. 5: State S6). This rule site is now completely unavailable for further hybridization and hence this protocol is called non-reusable rules IR-WPCR. The new current state $a_j^* = (b_i w_i y_i)^*$ at the 3' end of W then binds to a_j which is the current state for transition rule R_j (Fig. 5: State S7). At this stage, the next state transition begins with the polymerase binding to the head (3' end) of W , encoding the current state a_j^* (Fig. 5: State S2). Hence the state machine operates without thermal cycles and uses only polymerase to facilitate denaturation of the 3' of W from the old rule. Moreover, each rule allows copying

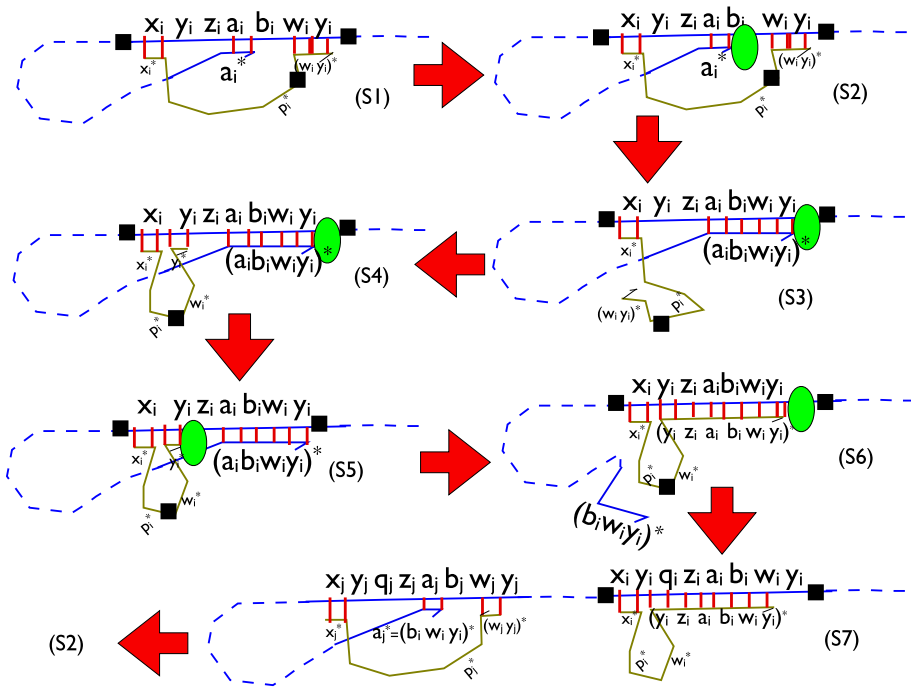


Fig. 5 Evaluation stage for non-reusable rules IR-WPCR protocol with the focus being only on the transition rule R_i to which the current state is hybridized: $S1$ WPCR strand W with protection strand P_i encoded as $(x_i p_i y_i)^*$ partially hybridized with rule R_i . Also the 3' end of W , bearing the current state a_i^* is hybridized to a_i of R_i . $S2$: polymerase binds to the 3' end of W . $S3$: polymerase extends a_i^* to copy $b_i w_i y_i$, thus displacing $w_i^* y_i^*$ of P_i from $w_i y_i$ of rule R_i located further away from x_i in R_i . $S4$: y_i^* of P_i binds to y_i located next to x_i in R_i . $S5$: polymerase binds with the 3' end of P_i . $S6$: 3' end of P_i is extended by the polymerase to copy $z_i a_i b_i w_i y_i$, thus displacing 3' end of W which has the new current state $a_j = b_i w_i y_i$. $S7$: 3' end of W bearing a_j^* binds to the a_j in rule R_j and the process repeats starting with the polymerase binding to the 3' end of W as shown in State S2

of next state *exactly once* and hence we call this machine *non-reusable rule IR-WPCR machine*.

3.2 Preparing a non-reusable rule IR-WPCR strand for computation

This section will describe how to obtain the secondary structure of the WPCR strand W as shown in Fig. 3. In order to ensure that the hairpin structure is stable, we tether the single strand with another nanostructure which has extended ds-DNA regions. For clarity, we again focus only on rule R_i for the description of the preparation stage (Fig. 6). We can use either a simple or a multi-step complex preparation protocol to ensure that each rule in W is properly “protected”.

In the *simple preparation protocol*, once we have guaranteed that the secondary structure of W is that of a hairpin, we directly introduce the protection strands P_i for each rule R_i into the solution. Since the $w_i y_i$ section is longer than the y_i region near x_i for each R_i the protection strand for R_i acquires the configuration shown in Fig. 5: State S1 with high probability. For more guaranteed hybridization of the protection strand with its

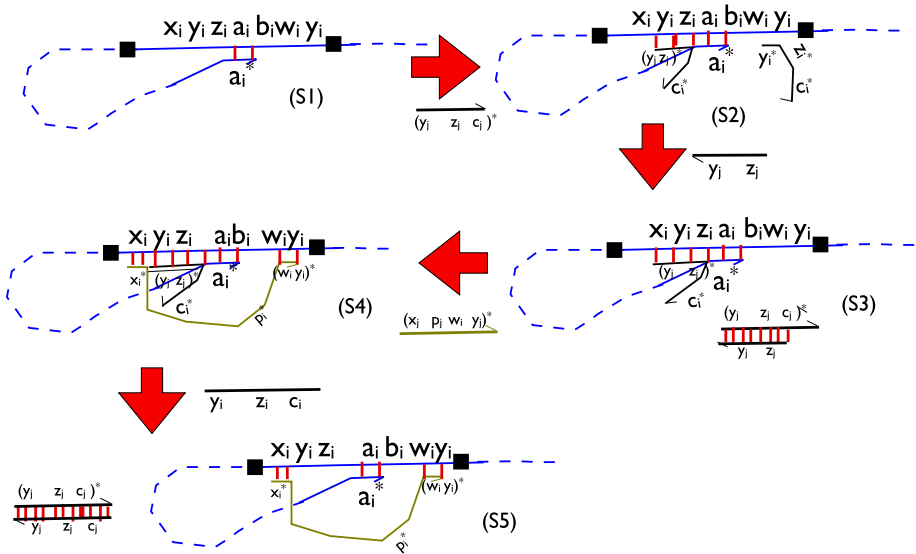


Fig. 6 Complex preparation protocol with respect to only rule R_i : S1 WPCR strand W tethered to support (not shown in the Figure). S2: $(y_i z_i c_i)^*$ is added to the solution. One copy binds to the y_i near x_i and another binds to y_i further away from it. S3: the copy of $(y_i z_i c_i)^*$ that binds to the y_i in R_i further away from x_i is removed by the addition of $y_i z_i$. The duplex thus formed is then removed from the solution using magnetic beads (not shown here). S4: Protection strand P_i encoded as $(x_i p_i w_i y_i)^*$ is introduced and it hybridizes with the x_i and free $w_i y_i$ of rule R_i . S5: the copy of $(y_i z_i c_i)^*$ that is bound to the y_i in R_i nearer to x_i is removed by the addition of $y_i z_i c_i$. Here too, the duplex is later removed using magnetic beads

corresponding transition rules, one may adhere to the *complex preparation protocol* which is presented in the following paragraph.

The Complex preparation protocol (Fig. 6) comprises of the following steps: once W is tethered to the supporting nanostructure, strand $(y_i z_i c_i)^*$ is added to the solution and the resultant nanostructure is shown in Fig. 6 (State S2). However, in order to attain the state shown in Fig. 5: State S1, we need to remove the copy of $(y_i z_i c_i)^*$ that is bound to y_i further away from x_i for each rule R_i . Hence we add $y_i z_i$ for each rule R_i and the toehold z_i in $y_i z_i$ facilitates the displacement of $(y_i z_i c_i)^*$ (Fig. 6: State S3). If $y_i z_i$ is biotin labeled, then we can easily remove it from the solution using streptavidin coated magnetic beads. This protocol ensures that $(y_i z_i c_i)^*$ that is bound to y_i further away from x_i is permanently removed from the solution and would not interfere with the final desired secondary structure of W . In the second last step of this protocol we add the protection strand P_i encoded as $(x_i p_i w_i y_i)^*$ and when it is partially hybridized to W , the latter looks like the one shown in Fig. 6 (State S4). Once P_i is in the desired location we add $y_i z_i c_i$ to remove $(y_i z_i c_i)^*$ hybridized with the y_i region near x_i by strand displacement using c_i as the toehold (Fig. 6: State S5). By including a biotin in $y_i z_i c_i$ the double stranded complex can be easily removed from the solution using streptavidin coated magnetic beads. It is important to add $y_i z_i c_i$ after P_i so that the latter does not hybridize with y_i closer to x_i . If it does so then as soon as polymerase is added, P_i will be extended, making that particular rule completely unavailable even before it can take part in the computation. Consequently, we have W in the desired state (Fig. 3), ready to start computing as soon as polymerase is added.

3.2.1 Benefits and limitations

The non-reusable rule IR-WPCR machine works very well for reducing back-hybridization since each rule can only be used once and is concealed thereafter. Consequently the 3' end of the WPCR machine cannot hybridize with the old rule and is forced to bind with the correct transition rule, thus preventing back-hybridization. However, inability to reuse a transition rule restricts the computational power of the machine.

The only limitation of this machine is that a rule can only be used once. One way to avoid this problem is to have several redundant copies of each transition rule encoded in W . However, we propose a more elegant solution to address this problem of a rule reusability using strand displacement in Sect. 4.

4 IR-WPCR machine with reusable rules

To address the limitation of a non-reusable rule IR-WPCR machine, this section describes a protocol that uses an additional strand A_i to displace P_i after primer extension. Refer to Fig. 7 for the description of the protocol with respect to transition rule R_i . This protocol comprises of the same steps: (a)–(d) as the non-reusable rule IR-WPCR protocol (See Sect. 3). Additionally it has a final step (e) where an auxiliary strand A_i (present in the reaction mixture) changes the secondary structure of P_i . This strand A_i can partially hybridize with the extended section of P_i and force P_i to return to its original “protection” state, permitting further computation using this transition rule. In a manner, the rule is “reset” after the state transition takes place. Hence this protocol is called *reusable rule 2IR-WPCR*. In the following section we describe how a state transition occurs in this new protocol. The preparation stage for this protocol is basically the same as that of non-reusable rules IR-WPCR (See Sect. 2). The only addition is that the auxiliary strands are present in the solution before the polymerase is introduced.

4.1 Computing with a reusable rule IR-WPCR strand

The first part of evaluation is the same as that in non-reusable rules IR-WPCR protocol (Fig. 7(Right): State S1–State S6). However, unlike the other protocol, in this method, the solution additionally contains a large concentration of A_i , which is encoded as $w_i y_i z_i a_i b_i$. Once P_i is extended, its w_i^* region is used as a toehold by A_i to displace the former. However, the $(w_i y_i)^*$ portion at the end of extended P_i is still free to hybridize with its complementary region on R_i (Fig. 7(Right): State S7). This step ensures that secondary structure of P_i is “reset” enabling R_i to participate in computation again. The last stage of evaluation is the same as in non-reusable rule IR-WPCR where the new state corresponds to $b_i w_i y_i = a_j$ for some j and hence the 3' end of the WPCR strand W binds to rule R_j (Fig. 7(Right): State S8).

This additional step of resetting the transition rule adds a number of new behaviors to the computation. For example, after each state transition, the portion on P_i between the ends hybridized with R_i has an additional ds-DNA region encoded as $w_i y_i z_i a_i b_i$. In other words, P_i becomes longer each time R_i is used. Moreover, note that a copy of A_i is consumed each time R_i is used so the solution must have an excess of A_i . However, at the beginning of the program execution, each P_i binds with R_i in W in preference to A_i since x_i as well as $w_i y_i$ hybridization in $P_i R_i$ complex is much more favorable than just $w_i y_i$

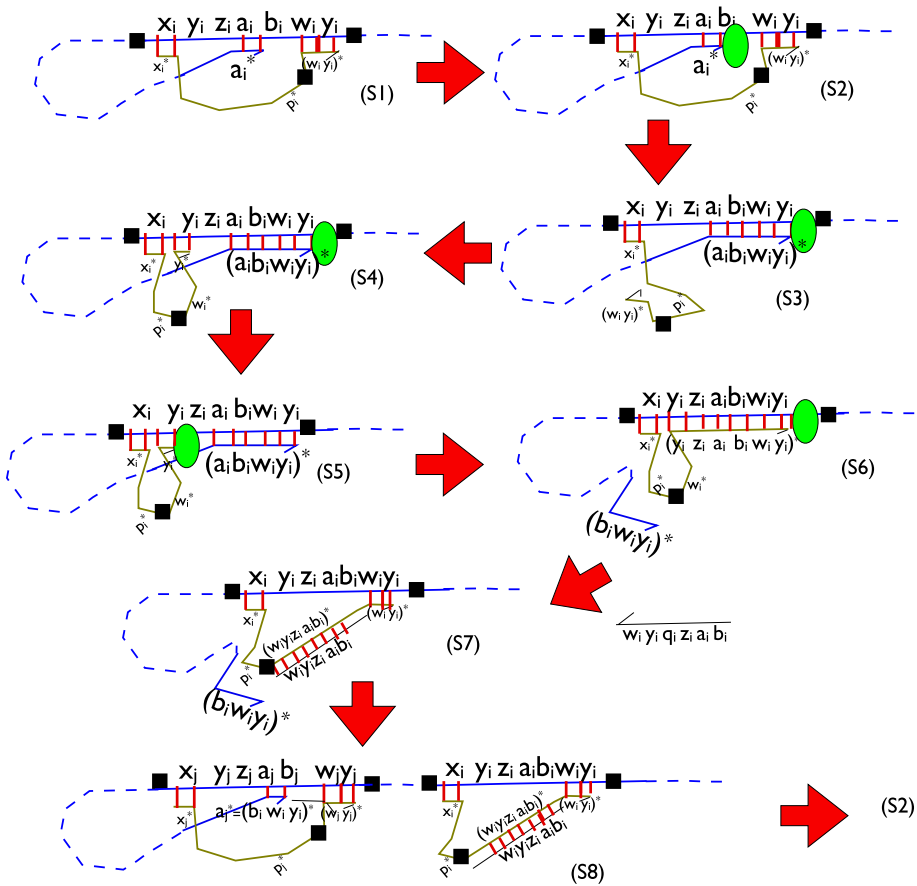


Fig. 7 Evaluation Stage in reusable rule IR-WPCR with the focus being only on the transition rule to which the current state is hybridized: *S1*: WPCR strand W with protection strand P_i encoded as $(x_i p_i y_i)^*$ partially hybridized with rule R_i . Also the 3' end of W , bearing the current state a_i^* is hybridized to a_i of R_i . *S2*: polymerase binds to the 3' end of W . *S3*: Polymerase extends a_i^* to copy $b_i w_i y_i$, thus displacing $(w_i y_i)^*$ of P_i from $w_i y_i$ of rule R_i located further away from x_i in R_i . *S4*: y_i^* of P_i binds to y_i located next to x_i in R_i . *S5*: Polymerase binds with the 3' end of P_i . *S6*: 3' end of P_i is extended by the polymerase to copy $z_i a_i b_i w_i y_i$, thus displacing 3' end of W which has the new current state $a_j = b_i w_i y_i$. *S7*: A_i encoded $(w_i y_i z_i a_i b_i)^*$ present in the solution displaces $(w_i y_i z_i a_i b_i)^*$ region of the protection strand P_i so that the configuration of the latter can be reset. *S8*: 3' end of W bearing a_j binds to the a_j in rule R_j and the process repeats again starting with the polymerase binding to the 3' end of W as shown in State *S2*

hybridization in $P_i A_i$ complex. P_i must also contain a stopper sequence to prevent P_i from being extended in an undesired direction.

4.1.1 Benefits and limitations

A reusable rule IR-WPCR machine is computationally equivalent to the original WPCR machine. Consequently, reusable rules IR-WPCR machines suffers from all the limitations of the original WPCR, such as back-hybridization and out-of-frame annealing. Consequently, previously mentioned protocols (Sakamoto et al. 1999; Rose et al. 2001) to avoid these problems (e.g., PNA mediation for back-hybridization) can be incorporated into our

system at the expense of losing full autonomy. However, in the following section we describe a relatively more complex IR-WPCR machine that can reuse its rules and still prevent back-hybridization.

5 Reusable rule IR-WPCR machine that prevents back-annealing using folding WPCR

5.1 Folding Whiplash PCR system

Folding WPCR using thermal cycling is a novel technique for preventing back-annealing while computing with a WPCR machine. It also serves as an important step for understanding the relatively complicated IR-WPCR machine with reusable states that prevents back-hybridization using folding WPCR. Refer to Fig. 8 for the rest of the discussion about folding WPCR. The current state for the i th transition is still a_i and the next state is b_i . However, the encoding of a transition rule in this protocol is $a_i x(a_i)^* b_i S(b_i)^* a_i S$. When cooled, in each transition rule, the $(a_i)^* b_i S(b_i)^* a_i$ section forms a hairpin loop. Thus only the current state of each transition rule is available for binding with the 3' end of the machine. This is the primary trick to preventing back-hybridization. Since the rest of the transition is always hidden in a tight hairpin loop, the machine cannot preferentially form longer step loops over shorter ones as is the case with the original protocol. The detailed protocol for folding WPCR is as follows: (i) without loss of generality, let us assume that the 3' end of the machine encodes a_i^* . When the system is cooled, it binds to the current state of rule R_i . (ii) Polymerase next binds to this end. With high strand displacement capability, it opens up the stem loop and copies the encoding on the rule until the first stopper sequence S . (iii) When heated, the strand denatures and 3' end of the strand bears $(a_i)^* x^* a_i (b_i)^*$. (iv) When cooled, $(a_i)^* x^* a_i$ part of it, forms a stem loop and 3' end now has only $(b_i)^*$ available for binding. This encodes for the next current state a_j which preferentially binds to rule R_j and not R_i as in the original protocol where the 3' end bore $(a_i b_i)^*$ and hybridized with rule R_i , since longer the hybridization, the more stable is the secondary structure. On the contrary, in the folding WPCR protocol, neither $(a_i)^*$ is available at the 3' end of WPCR strand nor b_i in rule R_i , since each is hidden in a stable hairpin loop. It should be remembered that the next state b_i , in each rule R_i is available only when it is copied to the 3' end of the WPCR strand. For the remainder of the time it remains hidden in a stem loop. Hence, neither b_i from R_i nor a_i^* from the 3' end of the strand is available to facilitate the hybridization of rule R_j with two successive encodings of the current states (a_i^* and a_j^*) at the 3' end of the WPCR strand W . Consequently this technique prevents back-hybridization.

5.2 Reusable rule IR-WPCR machine that prevents back-hybridization using folding WPCR

Refer to Fig. 9 for the reusable rule IR-WPCR machine that uses folding WPCR to prevent back-hybridization.¹ In the previously mentioned reusable rule IR-WPCR machine, recall

¹ Although Fig. 9 is very complex, it should be remembered that it is essentially a cross between Fig. 7 and 8. The hairpins in the computational rewrite rule and the 3' end of the machine are taken from Fig. 8 while the rest of the protocol comes from Fig. 7. Keeping this in mind, the reaction pathway in Fig. 9 is much easier to understand.

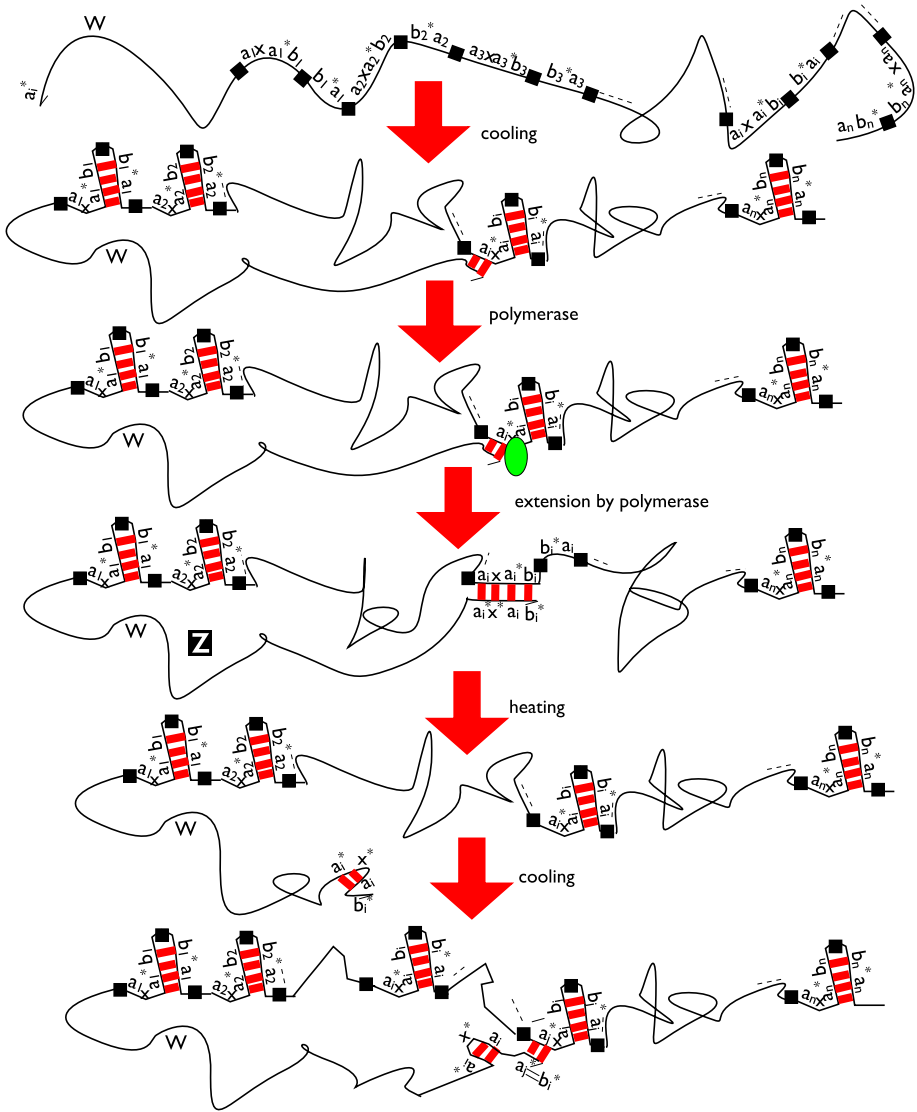


Fig. 8 Schematic of the protocol for the folding Whiplash PCR machine: *S1*: initial state of the WPCR strand *W*. *S2*: the solution is heated such that the next state in each rule hidden in a hairpin loop with current state of the machine being a_i . *S3*: polymerase binds to the 3' end of *W* (bearing the current state). *S4*: next state b_i^* is copied at the head of *W* by primer extension and hairpin loop is opened. *S5*: the mixture is heated so that *W* loses its hairpin structure (It may even open up the individual hairpin loops in each rule, not shown here). *S6*: the solution is cooled so that the head of *W* can bind to the new current state $b_i^* = a_i^*$ encoded at the 3' end of the strand and the whole state transition repeats again beginning with State *S2*. Note that the next state in each rule is hidden in a stem loop as is the old current state encoded at the 3' end of the WPCR strand. These two stem loop formations are key to preventing back-hybridization in this protocol

that for rule R_i , the current state is a_i while the next state is $b_i w_i y_i$. However, in the reusable rule IR-WPCR machine that also uses folding WPCR to prevent back-hybridization, the state encoding is $x_i y_i z_i a_i x(a_i)^* b_i w_i y_i S(b_i)^* a_i$. Under suitable temperature, $(a_i)^* b_i w_i y_i S(b_i)^* a_i$

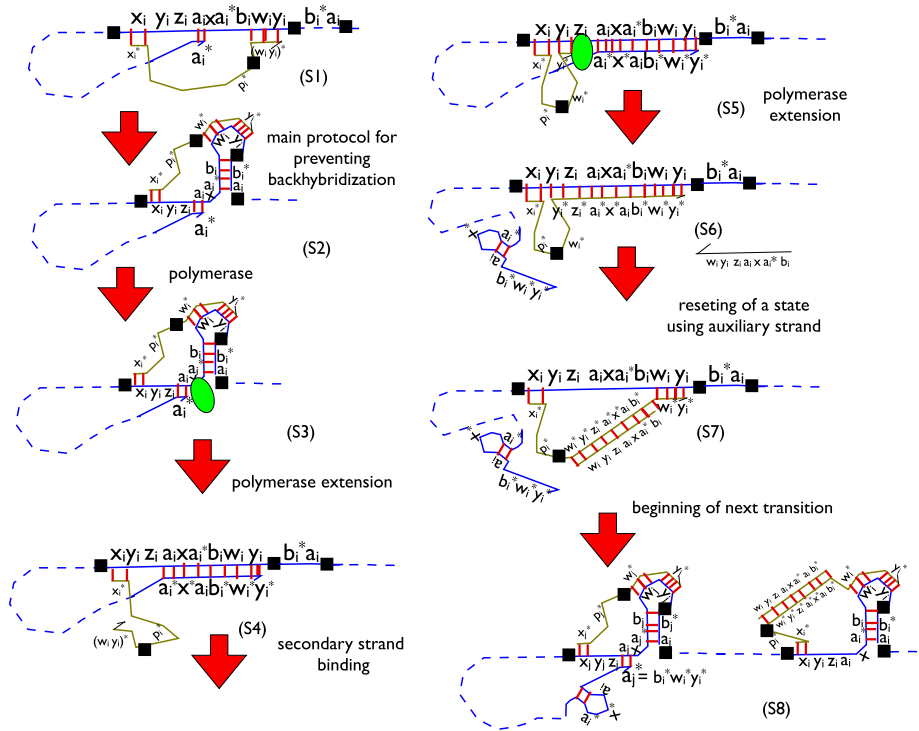


Fig. 9 Evaluation Stage for reusable rule IR-WPCR protocol that prevents back-hybridization using folding PCR with the focus being only on the transition rule R_i to which the current state is hybridized: *S1*: WPCR strand W with protection strand P_i encoded as $(x_i p_i y_i)^*$ partially hybridized with rule R_i . Also the 3' end of W , bearing the current state a_i^* is hybridized to a_i of R_i . *S2*: the temperature of the solution is such that the b_i part of the next state $b_i w_i y_i$ forms part of a stem loop. *S3*: polymerase binds to the 3' end of W . *S4*: polymerase extends a_i^* to copy $b_i w_i y_i$, thus displacing $w_i^* y_i^*$ of P_i from $w_i y_i$ of rule R_i located further away from x_i in R_i . Furthermore it opens the stem loop in which part of the next state was hidden. *S5*: y_i^* of P_i binds to y_i located next to x_i in R_i . Polymerase binds with the 3' end of P_i . *S6*: 3' end of P_i is extended by the polymerase to copy $z_i a_i x(a_i)^* b_i w_i y_i$, thus displacing 3' end of W which has the new current state $a_j = b_i w_i y_i$. *S7*: A_i encoded $(w_i y_i z_i a_i x a_i^* b_i)^*$ present in the solution displaces $(w_i y_i z_i a_i x a_i^* b_i)^*$ region of the extended protection strand P_i so that the configuration of the latter can be reset. Furthermore, at the 3' end of the WPCR strand the old state a_i forms part of a hairpin loop $(a_i)^* x^* a_i$ because of the solution temperature. *S8*: The next state of R_i is reset to its stem loop configuration as well. Additionally, 3' end of W bearing a_j^* binds to the a_j in rule R_j . At this stage, the process repeats starting with the polymerase binding to the 3' end of W as shown in State *S2*

forms a stem loop and the protection strand P_i hybridizes with x_i and $w_i y_i$. W.l.o.g., assume that the current state is a_i and thus the 3' end of the WPCR strand binds to rule R_i . When the polymerase binds to the 3' end of the WPCR strand, it opens up the stem loop hiding part of the next state encoding in this rule. The 3' end of the WPCR strand is extended as far as the first stopper sequence in rule R_i . This event, in turn, displaces the 3' end of the protection strand P_i . The latter now binds to its second best match in rule R_i namely y_i . Polymerase now binds to the 3' end of P_i and extends it to displace the 3' end of the WPCR strand (this end has the new current state of the machine, $b_i w_i y_i$ encoded in it). This event marks the completion of a state transition. An auxiliary strand already in solution, then resets rule R_i to its original configuration. However, the only difference between this protocol and the reusable rule IR-WPCR machine that cannot prevent back-hybridization is that the former uses folding

WPCR to minimize back-annealing. In other words, at the end of any state transition (say R_i for instance), the 3' end of the WPCR strand not only has $(b_i w_i y_i)^*$ encoded in it, but also has $(a_i)^* x a_i$ preceding the $(b_i w_i y_i)^*$ encoding. This $(a_i)^* x a_i$ encoding folds into a loop and hides the previous current state $(a_i)^*$. Similarly, in rule R_i , b_i portion of the next state $b_i w_i y_i$ is hidden in a loop after being copied at the 3' end of the WPCR strand and $w_i y_i$ of the same next state is bound to P_i . The simultaneous unavailability of previous current state at the 3' end of W and the next state in the previously copied rule R_i prevents longer stem loop formation (between the 3' end of the WPCR strand and the previous rule R_i) and encourages the 3' end of the WPCR strand to bind with rule R_j instead, thus minimizing back-hybridization.

6 Simulation of a finite state automata by a reusable rule IR-WPCR that prevents back-hybridization using folding WPCR

In this section we first present a proof by induction on the computational capability of the original WPCR machine since it is the simplest case. Next we show that the isothermal reactivating WPCR machine with reusable states that prevents back-hybridization using folding WPCR is computationally equivalent to the original WPCR machine. Combining these two results we can show that isothermal reactivating WPCR machine with reusable states that prevents back-hybridization using folding WPCR can simulate a restricted class of finite state machine. This restricted class of finite state machines refers to those machines that can be represented as a directed acyclic graph. In other words, input string is of finite length and, for each input, each state is visited at most once. One important application of such a finite state machine could be in molecular disease detection systems. With the disease symptoms representing inputs to the automata, the accept state of the machine will be the one that detects all the symptoms in sequence (meaning the finite state automaton computes the conjunction of all the symptoms present).

Lemma *A WPCR machine with n transition rules can simulate a finite state machine $M = (Q, \Sigma, \delta, q_0)$ where $Q = \{a_1, b_1, \dots, a_i, b_j, \dots, a_j, b_j, \dots, a_n, b_n\}$ is the set of states, $\Sigma = \{\epsilon\}$ is the input set, $q_0 = \{a_1\}$ is the start state and $\delta = \{(a_1, b_1), \dots, (a_i, b_i), \dots, (a_j, b_j), \dots, (a_n, b_n)\}$ is the set of transition functions and $|\delta| = n$.²*

Proof Since computation takes place by DNA hybridization, x^* is the complement of x and they indicate the same state. The transition table is encoded in a single strand of DNA as $a_1 b_1 S a_2 b_2 S \dots S a_n b_n$, where S indicates the separator between two states. Since the only input is the empty string ϵ , we neither have included it in the encoding of the transition rules or at the 3' end of the WPCR strand that encodes the current state of the machine. We will do induction on the number of transitions, k , the machine has undergone. □

Base case $k = 1$. a_1^* is encoded at the 3' end of the WPCR strand which represents the current state of the machine. Recall that the machine has n transition $(a_1, b_1), (a_2, b_2) \dots (a_n, b_n)$ encoded as $a_1 b_1 S a_2 b_2 S \dots a_n b_n S$ at the 5' end of the WPCR strand. S is a stopper

² To be consistent with the description of the original WPCR machine we use only empty input string for the simulation of this finite state machine. It should be remembered, if the original finite state machine can be represented as a directed acyclic graph, we can incorporate input as part of the transition rules and the initial input symbol as part of the current state at the beginning of program execution as described in Sect. 2.

sequence that is essential for the biochemical implementation of a state transition using polymerase. It is also a separator for two adjacent state transition encodings as we have mentioned before. The solution is cooled and since $(a_1)^*$ is complement to the current state a_1 of the transition rule (a_1, b_1) , it hybridizes with it, thus forming a stem loop. A polymerase present in solution next extends $(a_1)^*$ to copy the next state b_1 from the transition rule and stops because of the presence of S , the stopper sequence. The solution is then heated, so that the strand is denatured. This marks the end of the state transition (a_1, b_1) since the 3' end of the WPCR strand now bear $(b_1)^*$, the new current state.

Hypothesis Assume that the machine has undergone $k = m$ transitions. At the end of it, the current state of the machine is a_i . We will now show how it can simulate the $(m + 1)^{st}$ transition.

Induction The 3' end of the WPCR strand encodes for the current state a_i in the form of $(a_i)^*$. When the solution is cooled, it binds to the current state a_i of transition rule (a_i, b_i) encoded as $a_i b_i S$. Next, in presence of polymerase, it is extended to copy only the next state b_i and none of the encoding beyond it because of S . The solution when heated, the hairpin is converted into a single strand without any secondary structure. This marks the end of the $(m + 1)^{st}$ state transition with the new current state b_i encoded at the 3' end of the WPCR strand as $(b_i)^*$. \square

Lemma A reusable rule IR-WPCR machine that prevents back-hybridization using folding WPCR is computationally equivalent to the original WPCR machine that uses thermal cycling to execute a state transition.

Proof The approach for this proof involves showing that a state transition in a reusable rule IR-WPCR machine that prevents back-hybridization using folding WPCR, say machine M_1 can be simulated by the original WPCR machine, say M_2 and vice versa. In other words, we prove that when M_2 with current state encoding $(a_i)^*$ undergoes state transition a_i to b_i (rule R_i), M_1 undergoes the transition a_i (current state in rule R_i) to $b_i w_i y_i$ (next state in rule R_i). The other direction holds as well. Recall that in machine M_2 rule R_i is encoded as $a_i b_i S$ while in M_1 , it is $x_i y_i z_i a_i x (a_i)^* b_i w_i y_i S (b_i)^* a_i$. The difference in next state encoding arises from the fact that M_1 and M_2 use two very different protocols for executing a state transition.

While in M_2 , the main driving force is polymerase and thermal cycling, M_1 is mostly driven by strand displacement along with polymerization. Hence, when we cool the solution with M_2 in it, the 3' end of the WPCR strand binds to a_i in rule R_i . However, there is no external mediation involved in the running of M_1 , once the polymerase is introduced in the solution containing M_1 . In fact, we maintain the temperature of this solution such that 3' end of M_1 bearing a_i^* can bind to a_i in rule R_i . The next step is the same in the program execution with both machines when polymerase binds to the 3' end of each machine. However, in the extension phase, while in M_2 , b_i is copied directly at the 3' end of the strand, in M_1 , the next state is hidden in a hairpin loop and a polymerase with good strand displacement capability is needed in order to open the stem loop and copy the next state $b_i w_i y_i$. Some additional sequences are copied too, such as $x^* a_i$. These sequences near the 3' end of M_1 are essential for preventing back-hybridization that is a major drawback with M_2 . We will discuss this difference shortly.

In the context of highlighting the differences in the two protocols, once the next state is copied, M_2 depends on external energy such as heating to denature the strand so that the automaton can execute the next computational step (next state transition). On the other

hand, in M_1 , the protection strand P_i of R_i is displaced from its $w_i y_i$ binding with R_i when the next state $b_i w_i y_i$ is copied and it binds to the second best complementary sequences y_i adjacent to x_i in the same rule. Polymerase next binds to P_i 's 3' end and extends it to the first stopper sequence S , thus displacing the 3' end of M_1 . Hence the 3' end of the WPCR strand is now free to take part in the next state transition. Additionally to ensure that the transition rule R_i is reusable, the extended protection strand P_i is reset to its original configuration using an auxiliary strand A_i encoded as $w_i y_i z_i a_i x(a_i)^* b_i$.

At this time, b_i of the next state $b_i w_i y_i$ can be concealed in a stem loop to prevent back-hybridization. The next state $b_i w_i y_i$ remains hidden at all times except for when it is copied. Now recall that there was some additional sequences at the 3' end of the strand namely, $x^* a_i$ from state transition R_i . This along with the previous current state $(a_i)^*$ forms a tight loop, thus hiding a_i^* and preventing back-annealing in M_1 . Back-hybridization, however, is very a common occurrence in the running of M_2 since after state transition R_i , $(a_i b_i)^*$ encoded at the 3' end of M_2 , hybridizing with transition rule R_i , is energetically much more favorable than simply $(b_i)^*$ binding with $a_j = b_i$, the current state of transition rule R_j (this is also the expected transition). However, with M_1 , $(a_i)^*$ at the 3' end of the strand is hidden in a hairpin loop while b_i in rule R_i is also concealed in a distinct stem loop. Additionally $w_i y_i$ that encodes for the rest of the next state of rule R_i is bound to the protection strand P_i and hence the probability of a back-hybridization event is almost negligible. □

Theorem *A reusable rule IR-WPCR machine that prevents back-hybridization using folding WPCR can simulate any finite state automata that is implementable by the original machine.*

Proof Combining the above two lemmas we can show that a reusable rule IR-WPCR machine that prevents back-hybridization using folding WPCR can simulate the same restricted class of finite state machines as the original WPCR machine. □

7 Probabilistic analysis of IR-WPCR protocol

For the analysis of the stochastic behavior of a IR-WPCR machine, we describe the steps involved in a rate transition diagram. Based on this continuous time Markov Chain in Fig. 10, we can then compute the state transition probability and rate. We can also compute the probability and rate of “resetting” a state. We first need to explain all the rates shown in the Markov Chain.

7.1 Rate of polymerization

In Fig. 10, r_{poly} corresponds to the rate at which the next state of length l_1 bases is copied and r'_{poly} corresponds to the effective rate at which P_i (l_2 bases) is extended. If the polymerase can extend N_{bases} bases/sec, then $r_{poly} = \frac{N_{bases}}{l_1}$ while $r'_{poly} = \frac{N_{bases}}{l_2}$. r_{wait} is the mean DNA polymerase and strand encounter rate, given by $r_{wait} = \frac{N_{poly}}{N_{strands}} v_t$ where N_{poly} is the number of units of polymerase in the solution and $N_{strands}$ corresponds to the total number of strands (each P_i (corresponding to R_i) in a W and each A_i (corresponding to R_i) in the solution count for a separate strand). Additionally $v_t = \frac{1}{r_{poly} + r'_{poly}}$ denotes the mean number of distinct extensions/sec by 1 unit of polymerase under optimal conditions using excess target and primer (Rose et al. 2001). Furthermore, r_{exo} (or r'_{exo} for P_i) is the effective rate

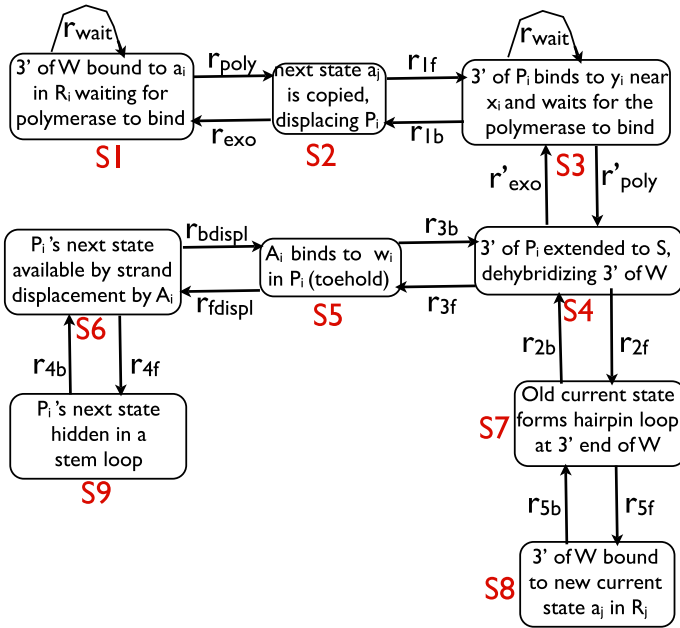


Fig. 10 Continuous time Markov Chain for rule R_i in the reusable rules IR-WPCR protocol that prevents back-hybridization using folding WPCR

of exonuclease activity. It is generally very small and is given by $r_{exo} = \frac{k_{exo} N_{poly}}{I_1 V}$ and $r'_{exo} = \frac{k_{exo} N_{poly}}{I_2 V}$ where k_{exo} for $\phi 29$ (our chosen polymerase) can be calculated from primer extension experiments (Saturno et al. 1995). Here, V is the total volume of the solution. Since target/primer is in excess, the exonuclease activity is limited by the concentration of the polymerase.

7.2 Rate of hybridization

For hybridization events, the rates r_{1f} , r_{2f} , r_{3f} , r_{4f} and r_{5f} are not concentration dependent since all the components are part of the same nanostructure. Hence $r_{if} \propto \sqrt{hl_i}$ where hl_i is the length of the hybridization segment (Hames and Higgins, 1995) for $i = 1, 2, 3, 4, 5$ under optimal conditions, thus neglecting effect of temperature and salt concentration. The rate of dehybridization r_{ib} (r_{1b} , r_{2b} , r_{3b} , r_{4b} , r_{5b}) is given by $k_{fe}^{-G_{se_i}}$ where $G_{se_i} = (\frac{4000}{T} - 11)hl_i$ (T is the temperature of the solution in K) and k_f is the forward rate constant (Winfree 1998a).

7.3 Rate of strand displacement

Finally, for estimating r_{fdispl} and r_{bdispl} we model strand displacement as a 1D random walk. After the toehold hybridization, let G denote the free energy of the 3 strand complex, G_l the free energy after one base pair migration to the left and G_r the same for migration to the right. Moreover, let $\Delta G_r = G_r - G$ and $\Delta G_l = G_l - G$ where the change in free energy can be computed from the nearest neighbor model. If p_r and p_l are the probability for right and left directional migration, then $p_r \propto \exp(\frac{-\Delta G_r}{RT})$ and $p_l \propto \exp(\frac{-\Delta G_l}{RT})$. Further, the mean

time for a single base migration is about $100 \mu \text{ sec}$ (Thompson et al. 1976). Hence, $r_{fdisl} = \frac{10000}{\sum_{i=1}^{l_{fdisl}} \frac{p_{r_i}}{p_{r_i} + p_{l_i}}}$ and $r_{bdisl} = \frac{10000}{\sum_{i=1}^{l_{bdisl}} \frac{p_{l_i}}{p_{r_i} + p_{l_i}}}$ where l_{fdisl} and l_{bdisl} denote the number of bases displaced in the forward and backward direction respectively. Observe that A_i displaces P_i to the right at this rate and thus r_{fdisl} corresponds to the forward rate of displacement.

7.4 Likelihood and rate of a state transition

Suppose we assume that the polymerase binding at the $3'$ end of the WPCR strand hybridized with a transition rule R marks the beginning of a state transition while the $3'$ end of the WPCR strand binding to another transition rule R' whose current state is identical to the next state of R marks the end of a single state transition. Then based on the Markov Chain in Fig. 10, the probability of a state transition is given by $\left(\frac{r_{poly}}{r_{poly} + r_{wait}}\right) \cdot \left(\frac{r_{1f}}{r_{1f} + r_{exo}}\right) \cdot \left(\frac{r'_{poly}}{r_{wait} + r_{1b}}\right) \cdot \left(\frac{r_{2f}}{r_{2f} + r'_{exo}}\right) \cdot \left(\frac{r_{5f}}{r_{5f} + r_{2b}}\right)$ and the rate of state transition, given that $3'$ end of W is bound to a_i in R_i and the polymerase is bound at the end of it, is given by $r_{poly} \left(\frac{r_{1f}}{r_{1f} + r_{exo}}\right) \cdot \left(\frac{r'_{poly}}{r_{wait} + r_{1b}}\right) \cdot \left(\frac{r_{2f}}{r_{2f} + r'_{exo}}\right) \cdot \left(\frac{r_{5f}}{r_{5f} + r_{2b}}\right)$. Given that polymerization is a much faster than hybridization, the rate of state transition is really limited by the r_{wait} parameter which in turn, is dependent on the relative concentration of the strands and the enzymes. Hence it is possible to increase rate of state transition by decreasing the concentration of polymerase. As Sahu et al. (2008) observed that incubating the sample containing cargo on a DNA track with polymerase at 30°C for 30 min was more than sufficient for the enzyme to drive the cargo on the circular track several times.

A rule is “reset” only if the protected strand is set back to its old configuration after the next state is copied from the rule. The probability that a state will be reset or made reusable after participating in the computation is given by $\left(\frac{r_{poly}}{r_{poly} + r_{wait}}\right) \cdot \left(\frac{r_{1f}}{r_{1f} + r_{exo}}\right) \cdot \left(\frac{r'_{poly}}{r_{wait} + r_{1b}}\right) \cdot \left(\frac{r_{3f}}{r_{3f} + r'_{exo}}\right) \cdot \left(\frac{r_{fdisl}}{r_{fdisl} + r_{3b}}\right)$ while the corresponding rate is $r_{poly} \left(\frac{r_{1f}}{r_{1f} + r_{exo}}\right) \cdot \left(\frac{r'_{poly}}{r_{wait} + r_{1b}}\right) \cdot \left(\frac{r_{3f}}{r_{3f} + r'_{exo}}\right) \cdot \left(\frac{r_{fdisl}}{r_{fdisl} + r_{3b}}\right)$. The rate limiting step in resetting a rule is the strand displacement operation and its rate can be biased by manipulating the toehold length (meaning length of w_i in each rule R_i).

7.5 Likelihood of preventing back-hybridization

Recall that a back-hybridization is an event where the old current state, say a at the $3'$ end of the WPCR strand paired with the next state b copied from the rule R (with a as the current state and b as the next state) binds to R in preference to rule R' which has b as the new current state since the first hybridization is energetically more favorable than the second hybridization. The probability of preventing back-hybridization in a reusable rule IR-WPCR machine that uses folding WPCR to prevent back-hybridization, is a function of two independent events: (1) the old current state (a) forming a stem loop at the $3'$ end of the WPCR strand at the conclusion of a state transition and (2) the next state (b) in the rule that was just executed (R) hiding in another hairpin loop at the same time. Now the probability of the old current state forming a stem loop at the $3'$ end of the WPCR strand at the end of a state transition is $p_1 = \left(\frac{r_{poly}}{r_{poly} + r_{wait}}\right) \cdot \left(\frac{r_{1f}}{r_{1f} + r_{exo}}\right) \cdot \left(\frac{r'_{poly}}{r_{wait} + r_{1b}}\right) \cdot \left(\frac{r_{2f}}{r_{2f} + r'_{exo}}\right)$ while the probability of

the next state in the rule that was just executed to also hide in another hairpin loop at the end of the state transition is $p_2 = \left(\frac{r_{poly}}{r_{poly}+r_{wait}}\right) \cdot \left(\frac{r_{1f}}{r_{1f}+r_{exo}}\right) \cdot \left(\frac{r'_{poly}}{r_{wait}+r_{1b}}\right) \cdot \left(\frac{r_{3f}}{r_{3f}+r'_{exo}}\right) \cdot \left(\frac{r_{fdispl}}{r_{fdispl}+r_{3b}}\right) \cdot \left(\frac{r_{4f}}{r_{bdispl}+r_{4f}}\right)$. Hence the probability of preventing back-hybridization after a state transition is $p = p_1 \cdot p_2$. Consequently, the probability of a back-hybridization event is $1-p$. While the likelihood of preventing back-hybridization is dependent on the rate of a polymerase binding with the WPCR machine, the rate limiting step in this phenomenon is really the strand displacement operation that releases the next state in a rewrite rule and allows it to be hidden in a stem-loop. Furthermore, observe that the first four terms of the probability of a state transition appear in the expression for p_1 while the first three terms of the same expression appear in the expression for p_2 . Thus the probability of preventing back-hybridization is approximately proportional to the square of the probability of a state transition. In other words, when the likelihood of a state transition increases, the likelihood of a back-hybridization event decreases and vice versa.

8 DNA design of IR-WPCR computing on a 3 state machine

In this section we provide a concrete example of the execution of the IR-WPCR protocol on a 3 state machine to illustrate the concept of a IR-WPCR machine further. We also describe how one can verify computation on this machine using gel electrophoresis and Fluorescence Resonance Energy Transfer (FRET) method. For clarity and simplicity, we will describe the protocol for only reusable rule IR-WPCR but has no capability to prevent back-hybridization. We have also not included the supporting nanostructure for the machine in the following description.

8.1 Encoding of the WPCR strand

Suppose we have a 3 state machine: $s_1 \rightarrow s_2 \rightarrow s_3$. Thus we have to encode only two transition rules viz. $s_1 \rightarrow s_2$ and $s_2 \rightarrow s_3$ in the WPCR strand $W(x_1 - y_1 - z_1 - a_1 - b_1 - w_1 - y_1 - S - x_2 - y_2 - z_2 - a_2 - b_2 - w_2 - y_2 - S - S' - a_1^*)$. Here S (6 bases) is the stopper sequence and S' is the spacer sequence that allows W to form a hairpin structure easily. Observe that a_1 is the encoding for state s_1 , while $b_1 - w_1 - y_1$ (or a_2) is the encoding for s_2 and finally $b_2 - w_2 - y_2$ is the encoding for s_3 . Two (s_2 and s_3) out of three states are used for priming and, hence, each state comprises of at least 15 bases.

We use polymerase $\phi 29$ to drive the computing, because of its excellent strand displacement capability (Sahu et al. 2008). However, it has to be used in very low concentration because of its high fidelity in higher concentrations. Further, if any state is used as a molecular beacon (Tyagi and Kramer 1996) for FRET experiments (e.g., $b_2w_2y_2$) the length has to be at least 24 bases. Besides W , we need to consider P_1 's DNA design. P_1 binds to the x_1 and w_1y_1 of rule $R_1 (s_1 \rightarrow s_2)$ before extension and, hence, at least three bases are necessary for each section to ensure stable hybridization. It should be remembered that, since $R_2 (s_2 \rightarrow s_3)$ is the last transition rule visited by the hairpin, it is not necessary to include P_2 or A_2 for R_2 . The resultant compact hairpin structure of W is more suitable for gel analysis.

We can perform two distinct experiments E_1 and E_2 to verify whether both state transitions are executed. This is essential because, in IR-WPCR protocols, once the polymerase is added to the solution, in theory, the reaction goes to completion. Thus, it is

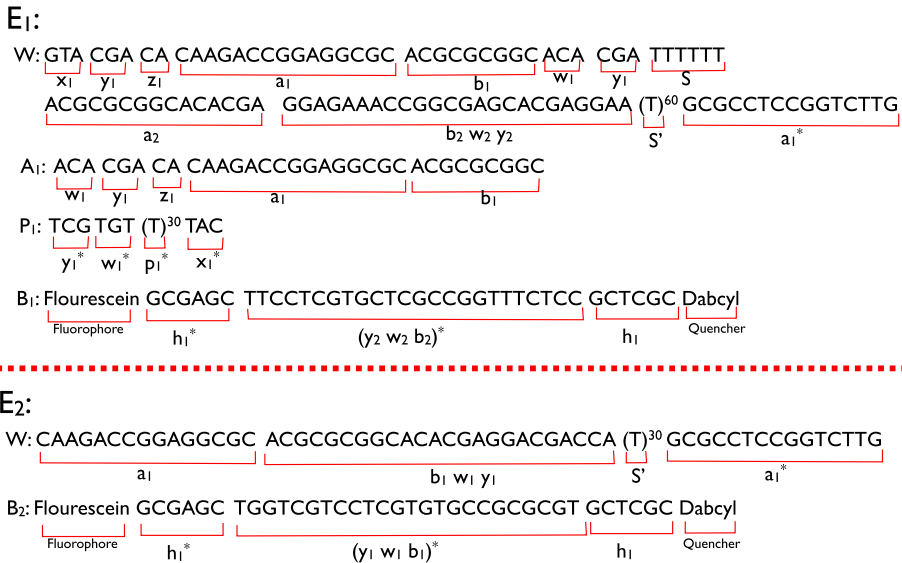


Fig. 11 DNA Sequences for experiments E_1 and E_2

not easy to probe the state of W after every state transition. Hence, E_1 corresponds to the experiment when W encodes $s_1 \rightarrow s_2$ as well as $s_2 \rightarrow s_3$ while, in E_2 , W encodes only $s_1 \rightarrow s_2$. Since we have only a three state machine these experiments together allow us to study the secondary structure of W in solution. It should be noted that for E_2 we do not need either P_1 or A_1 since s_2 is the final state for this experiment. Sequences designed using sequence symmetry minimization (Seeman 1990) for E_1 and E_2 are shown in Fig. 11.

8.2 Preparing the WPCR strand for computation

For experiment E_1 in order to ensure that W attains the secondary structure appropriate for IR-WPCR computing, we adopt the simple preparation protocol. We add P_1 to W and anneal the mixture to 30°C so that W can attain the hairpin structure and P_1 can bind to rule R_1 . Auxiliary strand A_1 is added in the next step. W is now ready to compute as soon as ϕ_{29} is added to the solution. For experiment E_2 we only need to anneal W to 30°C and add the polymerase to initiate computation.

8.3 Verification of computation with gel electrophoresis

Once very low concentration (about $0.05 \mu\text{M}$) of ϕ_{29} is added to the solution at 30°C , the resultant mixture is ready to be analyzed in about 30 min. A 10% native PAGE can be used to determine the size of the final product (W). For experiment E_1 , the auxiliary strand A_1 forms a separate band. We can compare it with the sizes of the WPCR strand prior to both state transitions. Ideally we should see a heavier band because at the end of computation, we have an extended P_1 hybridized to A_1 while still being hybridized to R_1 in W . For experiment E_2 , the distinction may be less prominent since this experiment neither involves A_1 or P_1 . Hence, gel analysis may not be quite conclusive. Information obtained from FRET analysis for both experiments may be more useful.

8.4 Verification of computation with FRET analysis

To perform FRET analysis, we use the molecular beacon (Tyagi and Kramer 1996) technique. For E_1 we use an extended $b_2w_2y_2$ region (about 24 bases in length) that is bound to a single strand B_1 encoded as $h(b_2w_2y_2)^*h^*$. There is a fluorophore and a quencher at the two ends of B_1 . h and h^* are complementary to each other and each is about 6 bases long. The idea of this molecular beacon is that, when W copies the next state in R_2 , it will displace B_1 . Consequently, B_1 is released and the h portion of B_1 binds to its h^* portion, resulting in a hairpin. As a result, there is a detectable drop in fluorescence signal. Similarly for E_2 , the molecular beacon B_2 ($h(b_1w_1y_1)^*h^*$) is hybridized to $b_1w_1y_1$ region of R_1 in W . Here too, as soon as the next state in R_1 is copied by ϕ_{29} , B_2 is released, forms a hairpin and the existing fluorescence is quenched.

9 Conclusion

In summary, WPCR is an useful model of computation for running distinct programs in parallel. In other words, the significance of WPCR computation lies in the fact that while other forms of autonomous molecular computing such as tiling assembly (Winfree et al. 1998) or Benenson automata (Benenson et al. 2001) operate based on global rules, it is possible to execute multiple WPCR machines, each holding its own distinct program, in parallel. However, existing protocols are not isothermal and autocatalytic at the same time, thus limiting the number of steps a WPCR machine can execute as well as its flexibility. In this paper, we presented three distinct isothermal and autocatalytic protocols for computing with WPCR. The key idea is to use strand displacement and polymerization of a secondary priming sequence to dehybridize the 3' of the WPCR strand once the next state is copied eliminating the need for a thermal cycle. Another important aspect of one of our protocols (reusable rule IR-WPCR that can prevent back-hybridization) is that it is capable of minimizing back-annealing events. This is a major problem with the original WPCR machine since it limits the program execution to only a few steps. However, folding WPCR protocol that is part of this reusable rule IR-WPCR protocol, in theory, can eliminate back-annealing completely, thus enabling uninterrupted program execution on a IR-WPCR machine. Furthermore, the rules in this machine are reusable after next states are copied from them, thus making it computationally equivalent to the original WPCR machine. Additionally, the protocol is isothermal and autocatalytic, making it readily usable in a wide variety of applications in diverse environments. One immediate future direction is to verify with laboratory experiments and computer simulation the proposed design of the 3 state machine (Sect. 8) and compare the yields with that of the original WPCR implementation (Sakamoto et al. 1999; Hagiya et al. 1999).

Acknowledgments This work is supported by NSF EMT NANO grant CCF-0829798 and CCF-0523555.

References

- Benenson Y, Paz-Elizur T, Adar R, Keinan E, Livneh Z, Shapiro E (2001) Programmable and autonomous computing machine made of biomolecules. *Nature* 414:430–434
- Goodman R, Berry R, Turberfield A (2004) The single-step synthesis of a DNA tetrahedron. *Chem Commun* 12:1372–1373

- Hagiya M, Arita M, Kiga D, Sakamoto K, Yokoyama S (1999) Towards parallel evaluation of boolean μ formulas with molecules. In: Rubin H, Woods DH (eds) DNA based computers, vol III, pp 55–72. American Mathematical Society, Providence, RI
- Hames BD, Higgins SJ (1995) Gene probes, vol 2. Oxford University Press, Oxford
- Komiya K, Yamamura M, Rose J (2008) Experimental validation of signal dependent operation in Whiplash PCR. DNA 14, LNCS
- Majumder U, LaBean T, Reif J (2008) Activatable tiles for compact, robust programmable assembly and other applications. DNA 13. LNCS 287:195–214
- Mathieu F, Liao S, Kopatsch J, Wang T, Mao C, Seeman N (2005) Six-helix bundles designed from DNA. Nano Lett 5:661–665
- Matsuda D, Yamamura M (2002) Cascading Whiplash PCR with a nicking enzyme. Lect Notes Comput Sci 2568:38–46
- Nishikawa A, Hagiya M (1999) Towards a system for simulating DNA computing with Whiplash PCR. In: Angeline PJ, Michalewicz Z, Schoenauer M, Yao X, Zalzal A (eds) Proceedings of the congress on evolutionary computation, vol 2, pp 960–966. Mayflower Hotel, Washington DC, USA. IEEE Press, New York
- Rose J, Komiya K, Yaegashi S, Hagiya M (2006) Displacement Whiplash PCR: optimized architecture and experimental validation, DNA 12. Lect Notes Comput Sci 4287:393–403
- Rose JA, Deaton RJ, Hagiya M, Suyama A (2001) Pna-mediated Whiplash PCR. Lect Notes Comput Sci 2340:104 – 116
- Sahu S, LaBean T, Reif JH (2008) A DNA nanotransport device powered by polymerase phi 29. Nano Lett 8(11):3870–3878
- Sakamoto K, Kiga D, Komiya K, Gouzu H, Yokoyama S, Ikeda S, Sugiyama H, Hagiya M (1999) State transitions by molecules. Biosystems 52:81–91
- Saturno J, Blanco L, Salas M, Esteban J (1995) A novel kinetic analysis to calculate nucleotide affinity of proofreading DNA polymerases. J Bio Chem 270(52):31235–31243
- Seeman N (1990) *De novo* design of sequences for nucleic acid structural engineering. J Biomol Struct Dyn 8:573–581
- Shih W, Quispe J, Joyce G (2004) A 1.7-kilobase single-stranded DNA that folds into a nanoscale octahedron. Nature 427:618–621
- Soloveichik D, Winfree E (2005) The computational power of Benenson automata. Theor Comput Sci 244:279–297
- Thompson BJ, Camien MN, Warner RC (1976) Kinetics of branch migration in double-stranded DNA. Proc Natl Acad Sci USA 73(7):2299–2303
- Tyagi S, Kramer F (1996) Molecular beacons: probes that fluoresce upon hybridization. Nat Biotechnol 14(3):303–308
- Winfree E (1998a) Simulation of computing by self-assembly. Technical report 1998.22, Caltech
- Winfree E (1998b) Whiplash PCR for $\sigma(1)$ computing. Technical report 1998.23, Caltech
- Winfree E, Liu F, Wenzler L, Seeman N (1998) Design and self-assembly of two-dimensional DNA crystals. Nature 394(6693):539–544