International Journal of Innovative
Computing, Information and Control
Volume **8**, Number **3(A)**, March **2012**

ICIC International ©2012 ISSN 1349-4198
pp. **1907–1916**

AUTOMATION OF DNA COMPUTING READOUT METHOD BASED ON REAL-TIME PCR IMPLEMENTED ON DNA ENGINE OPTICON 2 SYSTEM

Muhammad Faiz Mohamed Saaid[1], Ismail Ibrahim[2], Shahdan Sudin[2]
Mohd Saberi Mohamad[2], Zulkifli Md. Yusof[2]
Jameel Abdulla Ahmed Mukred[2], Kamal Khalil[2]
Zuwairie Ibrahim[2] and Junzo Watada[3]

[1]Department of Biomedical Engineering
Faculty of Engineering
University of Malaya
Kuala Lumpur 50603, Malaysia
mfms@um.edu.my

[2]Universiti Teknologi Malaysia
Skudai, Johor Darul Takzim 81310, Malaysia
ismail_xyz@yahoo.com; saberi@utm.my
{ shahdan; zmdyusof; jameel; kamal; zuwairie }@fke.utm.my

[3]Graduate School of Information, Production and Systems
Waseda University
2-7 Hibikino, Wakamatsu, Kita-Kyushu 808-0135, Japan
junzow@osb.att.ne.jp

Received December 2010; revised July 2011

Abstract. *Previously, an automation of a DNA computing readout method for the Hamiltonian Path Problem (HPP) has been implemented based on LightCycler System. In this study, a similar readout approach is implemented based on DNA Engine Opticon 2 System. The readout approach consists of two steps: real-time amplification in vitro using TaqMan-based real-time PCR, followed by an in silico phase. The in silico phase consists of a data clustering algorithm and an information processing to extract the Hamiltonian path after the TaqMan "YES" and "NO" reactions have been identified. The result indicates that the automation of DNA computing readout method can be efficiently implemented on DNA Engine Opticon 2 System.*
Keywords: DNA computing, Hamiltonian path problem, Real-time PCR

1. **Introduction.** A new computing paradigm based on DNA molecules appeared in 1994 when L. M. Adleman [1] launched a novel *in vitro* approach to solve the so-called Hamiltonian Path Problem (HPP) with seven vertices by DNA molecules. The goal of the HPP is to determine whether any path exists which commences at the 'start city' and finishes at the 'end city', and passes through each of the remaining cities exactly once. In conventional silicon-based computers, information is stored as binary numbers in silicon-based memories; in this approach, he encoded the information of the vertices by random DNA sequences. The computation is performed in bio-molecular reaction fashion involving procedures such as hybridization, denaturation, ligation and Polymerase Chain Reaction (PCR). The output of the computation, also in the form of DNA molecules can be read and printed by a process called electrophoretical fluorescence.

Existing models of DNA computation are based on various combinations of bio-operations, which are *synthesizing*, *mixing*, *annealing* (*hybridization*), *melting* (*denaturation*),

amplifying (*copying*), *separating*, *extracting*, *cutting*, *ligating*, *substituting*, *detecting* and *reading* [2]. Based on this model, the DNA computation implementation can be classified into three important aspects: nucleic acid design, DNA algorithms and readout method. The first step for wet-lab experiment of DNA computation is to find a good set of DNA sequences. After that, the desired sequences are synthesized based on the specific problem. Then, the computational part of the DNA algorithms is performed, where *mixing*, *annealing* (*hybridization*), *melting* (*denaturation*), *amplifying* (*copying*), *separating*, *extracting*, *cutting*, *ligating*, *substituting* and *detecting* are fully applied to implement the algorithm for the computation. The final part of the implementation is visualization of the output result, where the *readout* operation can be implemented by utilizing the biotechnology, such as DNA sequencing. The readout method implementation issue is stated in [3] as an important drawback of current DNA computation, which requires the developments of high-throughput screening technologies to overcome the limitation imposed by existing readout methods. However, the readout problem receives less attention from researchers, instead of computational part of DNA computing.

In [4], a technique for reading out arbitrary graphs with up to n nodes using an $n \times n$ biochip incorporating standardized DNA sequences was proposed, which made the biochip universal for all graphs of the size. Such graph can be Directed Hamiltonian Path (DHP) at large, with all graphs being superimposed with each other. The superposition of graphs can be diluted by detecting n^2 different quantum dot barcode labels within the spots on the universal biochip. Then, the partial readout of special class of permutation graphs is subjected to computer-based heuristics for isolating individual graphs from a collection of graphs. However, this method is not experimentally verified in the laboratory.

Previously, Ibrahim et al. [5] implemented a TaqMan based real-time Polymerase Chain Reaction (PCR) for reading out DNA solution that encodes the Hamiltonian path. The readout method, which has been implemented using a LightCycler System, consists of an *in vitro* computation and an *in silico* information processing. Several TaqMan reactions were performed to investigate the order of the Hamiltonian path in the *in vitro* computation part. The output of the real-time PCR can be distinguished as either "YES" or "NO" reaction. After that, the output from the *in vitro* computation was subjected into *in silico* algorithm to produce the Hamiltonian path.

Recently, Saaid et al. [6] proposed a data clustering technique to automatically identify the output of the real-time PCR. Fuzzy C-Means (FCM) clustering algorithm is applied to separate the "YES" and "NO" reactions. It was shown that the FCM is capable to cluster the two different reactions of real-time PCR.

In this paper, based on work in [6], an automation system for DNA computing readout method, which is implemented using DNA Engine Opticon 2 System and also consists of *in vitro-in silico* approach, is reported. During the *in silico* information processing phase, Alternative Fuzzy C-Means (AFCM) and FCM clustering algorithm are implemented for automatic classification of "YES" and "NO" reactions.

2. **Basic Notation and Principle.** First of all, $v_{1(a)}v_{2(b)}v_{3(c)}v_{4(d)}$ denotes a double-stranded DNA (dsDNA), which contains the base-pairs subsequences, v_1, v_2, v_3 and v_4, respectively. Here, the subscripts in parenthesis (a, b, c and d) indicate the length of each respective base-pair subsequence. For instance, $v_{1(20)}$ indicates that the length of the double-stranded subsequence, v_1 is 20 base-pairs (bp). When convenient, a dsDNA may also be represented without indicating segment lengths (e.g., $v_1v_2v_3v_4$).

A reaction denoted by TaqMan(v_0, v_k, v_l) indicates that real-time PCR is performed using forward primer v_0, reverse primer v_l and TaqMan probe v_k. Based on the proposed approach, there are two possible reaction conditions regarding the relative locations of

the TaqMan probe and reverse primer. In particular, the first condition occurs when the TaqMan probe specifically hybridizes to the template, between the forward and reverse primers, while the second occurs when the reverse primer hybridizes between the forward primer and the TaqMan probe. As shown in Figure 1, these two conditions would result in different amplification patterns during the real-time PCR, given that the same DNA template (i.e., assuming that they occurred separately, in two different PCR reactions). The higher fluorescent output of the first condition is a typical amplification plot for the real-time PCR. In contrast, the relatively lower fluorescent output of the second condition, which reflects the cleavage of a lower number of TaqMan probes via DNA polymerase due to the 'unfavourable' hybridization position of the reverse primer, is due to linear rather than exponential amplification of the template. Thus, TaqMan(v_0, v_k, v_l) = YES if an amplification plot similar to the first condition is observed, TaqMan(v_0, v_k, v_l) = NO if an amplification plot similar to the second condition is observed.

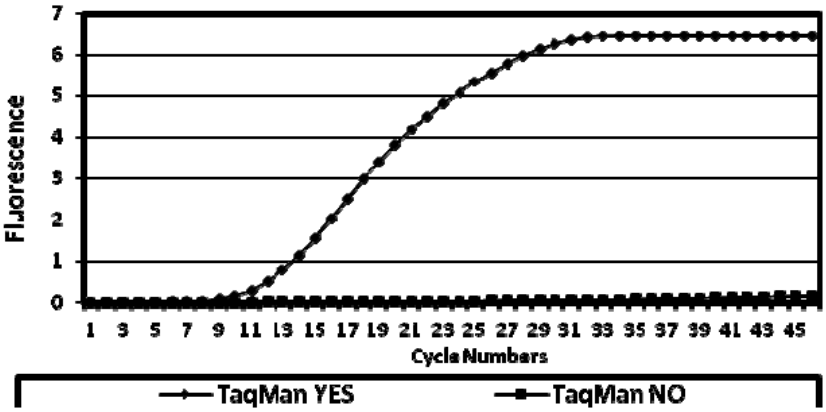

FIGURE 1. An example of amplification plots corresponding to TaqMan(v_0, v_k, v_l) = YES (first condition) and TaqMan(v_0, v_k, v_l) = NO (second condition) implemented on DNA engine opticon 2 system

3. **The Real-Time PCR-based Readout Approach Implemented on DNA Engine Opticon 2 System.** In DNA computing for HPP, an output of an in vitro computation can be represented by a dsDNA $v_{a(20)}v_{b(20)}v_{c(20)}v_{d(20)}v_{e(20)}v_{f(20)}$, where the Hamiltonian path $V_a \to V_b \to V_c \to V_d \to V_e \to V_f$, begins at node V_a, ends at node V_f, and contains intermediate nodes V_b, V_c, V_d and V_e respectively. In the readout implementation, the starting and ending nodes are already known. Note that the presence of all intermediate nodes is also known in advance. The problem is how to determine the specific order of the intermediate nodes for a Hamiltonian path.

3.1. **The in vitro phase.** The *in vitro* part of the approach consists of $[(|V|-2)^2 - (|V|-2)]/2$ real-time PCR reactions, each denoted by TaqMan(v_0, v_k, v_l) for all k and l, such that $0 < k < |V|-2$, $1 < l < |V|-1$ and $k < l$, where $|V|$ is a number of nodes. For the seven nodes Hamiltonian path, $V_0 \to V_1 \to V_4 \to V_2 \to V_5 \to V_3 \to V_6$, ten different TaqMan reactions are performed with DNA template $v_0v_1v_4v_2v_5v_3v_6$. All the TaqMan reactions required for the readout along with the "YES" and "NO" classification are as follows:

(1) TaqMan(v_0, v_1, v_2) = YES
(2) TaqMan(v_0, v_1, v_3) = YES
(3) TaqMan(v_0, v_1, v_4) = YES
(4) TaqMan(v_0, v_1, v_5) = YES

(5) TaqMan(v_0, v_2, v_3) = YES
(6) TaqMan(v_0, v_2, v_4) = NO
(7) TaqMan(v_0, v_2, v_5) = YES
(8) TaqMan(v_0, v_3, v_4) = NO
(9) TaqMan(v_0, v_3, v_5) = NO
(10) TaqMan(v_0, v_4, v_5) = YES

At first, a pool of 140-bp input molecules $v_{0(20)}v_{1(20)}v_{4(20)}v_{2(20)}v_{5(20)}v_{3(20)}v_{6(20)}$ is prepared, via standard protocol of parallel overlap assembly (POA) of single-stranded DNA strands (ssDNAs). For this purpose, 13 ssDNAs are required, including additional ssDNAs, which act as link sequences for self-assembly. These strands are listed in Table 1. After completion, amplification via PCR was performed using the same protocol as POA. The forward primers and reverse primers used for the PCR reaction were 5'-CGTCAAGGCCGTCTCTATAT-3' and 5'-GTAGATTAAGAAGGTGCGCG-3', respectively. The PCR product was subjected to gel electrophoresis and the resultant gel image was captured, as shown in Figure 2. The 140-bp band in lane 2 shows that the input molecules have been successfully generated. Afterwards, the DNA of interest is extracted. The final solution for real-time PCR was prepared via dilution of the extracted solution, by adding ddH2O (Maxim Biotech, Japan) into $100\mu l$.

TABLE 1. The required single-stranded DNAs for the generation of input molecules

Name	DNA Sequences (5'-3')	Length (mer)
v_0	CGTCAAGGCCGTCTCTATAT	20
v_1	CCACTGGTTCTGCATGTAAC	20
v_4	TCCACGCTGCACTGTAATAC	20
v_2	TGGACAACCGCAGTTACTAC	20
v_5	ATGCGCCAGCTTCTAACTAC	20
v_3	AGGAAACCTCACGACAGTCT	20
v_6	CGCGCACCTTCTTAATCTAC	20
v_0v_1	GAACCAGTGGATATAGAGACGGCCTTGACG	30
v_1v_4	GCAGCGTGGAGTTACATGCA	20
v_4v_2	CGGTTGTCCAGTATTACAGT	20
v_2v_5	GCTGGCGCATGTAGTAACTG	20
v_5v_3	GAGGTTTCCTGTAGTTAGAA	20
v_3v_6	GTAGATTAAGAAGGTGCGCGAGACTGTCGT	30

Ten separate real-time PCR reactions were performed in parallel, in order to implement the first stage of the proposed HPP readout. After the initial activation step at 95°C for 15 minutes, the amplification consists of 45 cycles of denaturation and annealing/extension, performed at 94°C for 15s and 60°C for 60s, respectively. The resulting real-time PCR amplification plots are illustrated in Figure 3. After all real-time PCR reactions are completed, the *in vitro* output is subjected to an *in silico* phase to produce the satisfying Hamiltonian path of the HPP instance.

3.2. **In silico phase.**

3.2.1. *FCM and AFCM clustering algorithms.* Once the in vitro phase is completed, the data from DNA Engine Opticon 2 System was exported into a computer. These data were written in text file format, which show the fluorescence intensity from the first to the 46th thermal cycle for all 10 different TaqMan reactions. These data are then subjected to a clustering algorithm based on FCM and AFCM.

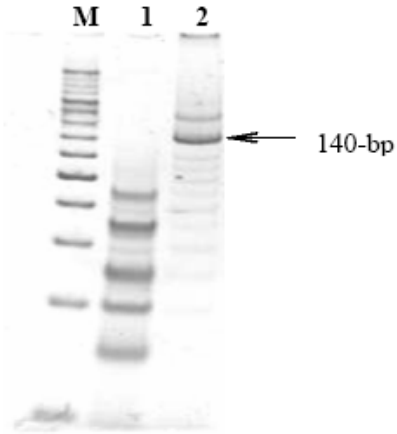

FIGURE 2. Gel image for the preparation of input molecules. Lane M denotes a 20-bp molecular marker, lane 1 is the product of initial pool generation based on parallel overlap assembly, and lane 2 is the amplified PCR product.

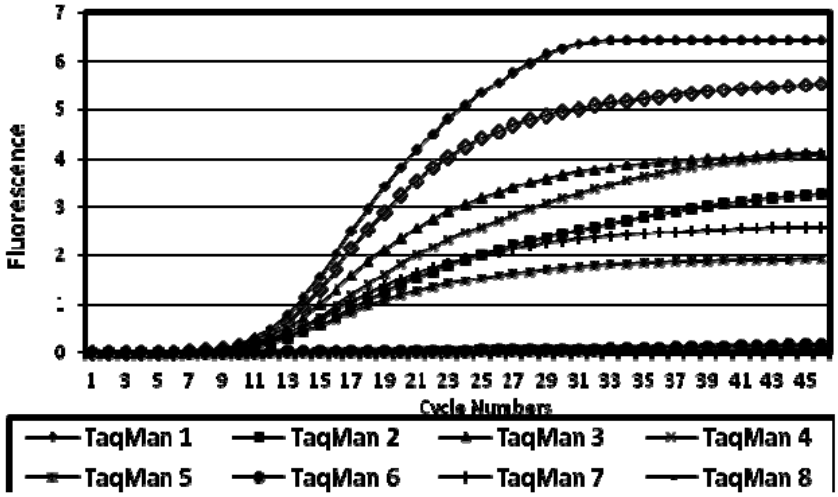

FIGURE 3. Output of real-time PCR for readout of $V_0 \to V_1 \to V_4 \to V_2 \to V_5 \to V_3 \to V_6$ implemented on DNA engine opticon 2 system. Reaction 1 to 10 indicate the TaqMan(v_0, v_k, v_l) reactions.

Fuzzy Clustering has become a well known and powerful method in cluster analysis, and has been applied in many fields such as image segmentation [12] and speaker recognition [13]. FCM is a data clustering technique based on the optimization of objective function [7]:

$$J(U,Y) = \sum_{i=1}^{C} \sum_{j=1}^{N} (\mu_{ij})^m \left\| x_j - y_i \right\|^2 \tag{1}$$

where x_j is the data, y_i is the cluster center, N is the number of data, C is the number of cluster, and m indicates the fuzziness value index. $||x_j - y_i||$ is Euclidean distance between x_j and y_i. Each data point in the data set is required to belong to exactly one cluster. Let $X = \{x_1, x_2, \cdots, x_N\}$ be a collection of data. By minimizing the objective function (1), X is classified into C homogeneous clusters, where the y_i values in $Y = \{y_1, y_2, \cdots, y_C\}$ are the cluster centers. In FCM, $U = (\mu_{ij})_{N*C}$ is a fuzzy partition matrix, in which μ_{ij} indicates the membership degree of each data point in the data set to cluster i. The value

U should satisfy the following conditions:

$$\mu_{ij} \in [0,1], \quad \forall i = 1, \ldots, C, \quad \forall j = 1, \ldots, N \tag{2}$$

$$\sum_{i=1}^{C} \mu_{ij} = 1, \quad \forall j = 1, \ldots, N \tag{3}$$

The cluster center can then be calculated as:

$$y_i = \frac{\sum_{j=1}^{N} (\mu_{ij})^m x_j}{\sum_{j=1}^{N} (\mu_{ij})^m}, \quad \forall i = 1, \ldots, C \tag{4}$$

and the fuzzy partition matrix, U, is updated using the following equation:

$$\mu_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\|x_j - y_i\|}{\|x_j - y_k\|} \right)^{\frac{2}{m-1}}} \tag{5}$$

AFCM, proposed by Wu and Yang [7], is based on the minimization of an objective function:

$$J(U, Y) = \sum_{i=1}^{C} \sum_{j=1}^{N} (\mu_{ij})^m \left(1 - \exp\left(-\beta \|x_j - y_i\|^2\right)\right) \tag{6}$$

This objective function involves an exponential distance between x_j and y_i, which is given by:

$$d(x_j, y_i) = \left(1 - \exp\left(-\beta \|x_j - y_i\|^2\right)\right)^{1/2} \tag{7}$$

where β is a positive constant, defined by:

$$\beta = \left(\frac{\sum_{j=1}^{N} \|x_j - \bar{x}\|^2}{N} \right)^{-1} \tag{8}$$

and $\bar{x}$ is defined as:

$$\bar{x} = \left(\frac{\sum_{j=1}^{N} x_j}{N} \right) \tag{9}$$

The necessary condition for minimizing (6) is given in (10) and (11):

$$y_i = \frac{\sum_{j=1}^{N} (\mu_{ij})^m \exp\left(-\beta \|x_j - y_i\|^2\right) x_j}{\sum_{j=1}^{N} (\mu_{ij})^m \exp\left(-\beta \|x_j - y_i\|^2\right)}, \quad \forall i = 1, \ldots, C \tag{10}$$

$$\mu_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\left(1 - \exp\left(-\beta \|x_j - y_i\|^2\right)\right)}{\left(1 - \exp\left(-\beta \|x_j - y_k\|^2\right)\right)} \right)^{\frac{1}{m-1}}} \tag{11}$$

Step 1:	Initialize the membership matrix U with random values.
Step 2:	Calculate the cluster center, Y.
Step 3:	Update the fuzzy partition matrix.
Step 4:	Calculate cost function J.
Step 5:	If $\|\|U(t+1) - U(t)\|\| < \varepsilon$ then stop; otherwise, go to Step 2.

FIGURE 4. The FCM and AFCM algorithms

Step 1:	Initialize the membership matrix U with random values.
Step 2:	Calculate the cluster center, Y.
Step 3:	Update the fuzzy partition matrix.
Step 4:	Calculate cost function J.
Step 5:	If $\|\|U(t+1) - U(t)\|\| < \varepsilon$ then stop; otherwise, go to Step 2.
Step 6:	Classify each TaqMan reaction using the predefined rule.

FIGURE 5. Classification of TaqMan reaction using FCM and AFCM algorithms

The FCM and AFCM algorithm are described in Figure 4. The clustering algorithm begin by initializing the fuzzy partition matrix. At iteration step t, cluster center is calculated, followed by updating the partition matrix. Next the cost function J is calculated. The process can be stopped if $||U(t+1)-U(t)|| < \varepsilon$, where ε is the error value for stopping criterion.

3.2.2. *Implementation of clustering algorithm.* In order to cluster the TaqMan reaction results into, "YES" and "NO" reactions, each reaction plot is represented as a vector, $\boldsymbol{x_j} = \{x_{j(1)}, x_{j(2)}, \cdots, x_{j(46)}\}$, where $x_{j(i)}$ denotes the fluorescence intensity measured after amplification cycle ith in TaqMan reaction j. The reactions are then clustered into two groups, with centers at $\boldsymbol{y_1} = \{y_{1(1)}, y_{1(2)}, \cdots, y_{1(46)}\}$ and $\boldsymbol{y_2} = \{y_{2(1)}, y_{2(2)}, \cdots, y_{2(46)}\}$. These two centers can be viewed as a plot similar to the TaqMan reaction "YES" and "NO". Then, the TaqMan reactions are classified into "YES" and "NO" groups, by comparing the partition matrix U. Let us say that $\boldsymbol{y_2}$ represents the "YES" center, and $\boldsymbol{y_1}$ represent the "NO" center (note that $\boldsymbol{y_2}$ does not always represent the "YES" center, when the clustering algorithm is run). We can say that $y_{2(46)} > y_{1(46)}$. Consider example values, μ_{11} and μ_{21}, which are equal to 0.6 and 0.4, respectively. The "YES" and "NO" reactions can be determined by the following rule:

if $(y_{1(46)} > y_{2(46)}$ and $\mu_{1j} > \mu_{2j})$ or $(y_{2(46)} > y_{1(46)}$ and $\mu_{2j} > \mu_{1j})$
$\boldsymbol{x_j}$ = "YES"
else $\boldsymbol{x_j}$ = "NO"

Based on the proposed rule, we can classify $\boldsymbol{x_j}$ as a "NO" reaction since $\mu_{1j} > \mu_{2j}$ and $y_{1(46)} < y_{2(46)}$. Applying this rule, we can classify the "YES" and "NO" reactions for each set of TaqMan reactions. The whole classification process can be described in Figure 5. In this implementation, $\varepsilon = 0.00001$, $N = 10$, $C = 2$ and $m = 2$. The clustering process has been done using Matlab 7.0, using a computer with 2.8GHz processor and 2GB RAM.

Figures 6 and 7 shows the result of implementation of FCM and AFCM algorithm. Fuzzy partition values with the classification of "YES" and "NO" reactions are listed in Tables 2 and 3.

3.2.3. *In silico information processing.* Based on Tables 2 and 3, result from AFCM implementation was selected and subjected to an *in silico* information processing as follows:

Input: N[0...|V| − 1] = 2 // N[0, ?, ?, ?, ?, ?, 6]
A[1...|V| − 2] = |V| // A[1, 1, 1, 1, 1]
for $k = 1$ to $|V| - 3$

for $l = k + 1$ to $|V| - 2$
if TaqMan(v_0, v_k, v_l) = YES
$\mathrm{A}[l] = \mathrm{A}[l] + 1$
else $\mathrm{A}[k] = \mathrm{A}[k] + 1$
endif
endfor
$\mathrm{N}[\mathrm{A}[k]] = k$
endfor
$\mathrm{N}[\mathrm{A}[|V| - 2]] = |V| - 2$

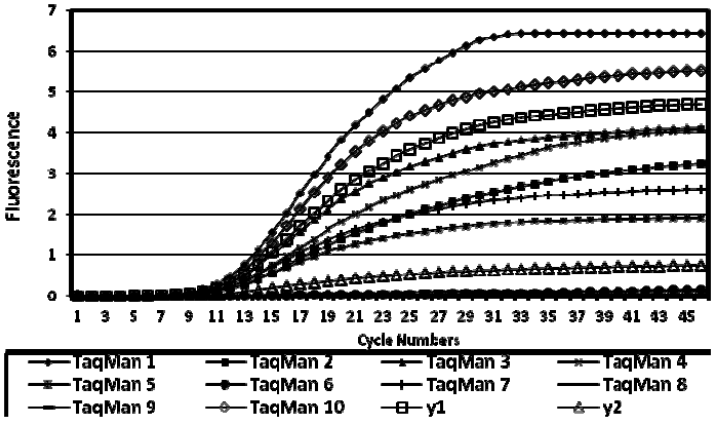

FIGURE 6. Output of real-time PCR with "YES" and "NO" centers implemented by FCM clustering algorithm with $y_{1(46)} > y_{2(46)}$

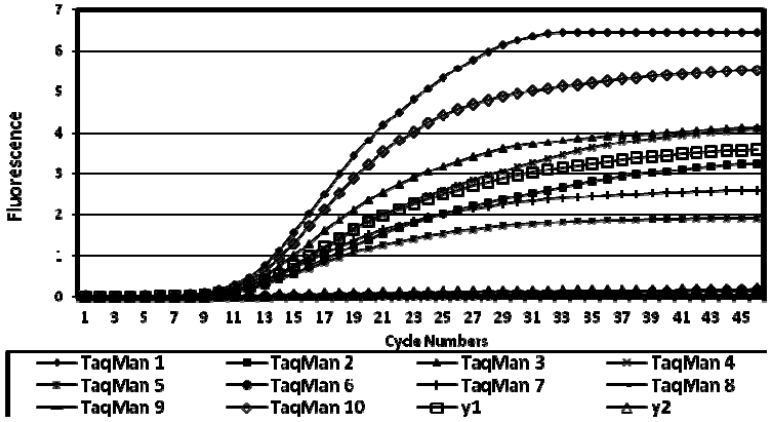

FIGURE 7. Output of real-time PCR with "YES" and "NO" centers implemented by AFCM clustering algorithm with $y_{1(46)} > y_{2(46)}$

In this algorithm, an array ($\mathrm{N}[0 \ldots |V| - 1$) that stores all the nodes of the Hamiltonian path is defined. In addition, an array of aggregation values ($\mathrm{A}[1..|V| - 2]$) that is also defined to locate the Hamiltonian path in each array of nodes. The input array N is first initialized to N = {0, ?, ?, ?, ?, ?, 5} since the start and end of the path are known, in advance. Next, the aggregation array A is initialized to A = {1, 1, 1, 1, 1}. During the loop operations of the algorithm, the value of the array A is increased in each iteration step. The aggregation array $\mathrm{A}[i]$ is used to index the nodes array for each value of k. After the loop operation $|V| - 2$ is assigned to the $\mathrm{N}[\mathrm{A}[|V| - 2]$. The output of the *in silico* algorithm can be viewed by calling back all the nodes array, N[0] to N[|V| − 1]. The outcome of this *in silico* algorithms is N = {0, 1, 4, 2, 5, 3, 6}.

TABLE 2. Partition matrix value for each TaqMan reaction based on FCM clustering algorithm

TaqMan	μ_{1j}	μ_{2j}	Manual observation	Reaction $y_{1(46)} > y_{2(46)}$
1 TaqMan(v_0, v_1, v_2)	0.894540	0.105460	"YES"	"NO"
2 TaqMan(v_0, v_1, v_3)	0.606420	0.393580	"YES"	"YES"
3 TaqMan(v_0, v_1, v_4)	0.972090	0.027907	"YES"	"YES"
4 TaqMan(v_0, v_1, v_5)	0.906060	0.093941	"YES"	"YES"
5 TaqMan(v_0, v_2, v_3)	0.172220	0.827780	"YES"	"NO"
6 TaqMan(v_0, v_2, v_4)	0.018769	0.981230	"NO"	"NO"
7 TaqMan(v_0, v_2, v_5)	0.450780	0.549220	"YES"	"NO"
8 TaqMan(v_0, v_3, v_4)	0.021623	0.978380	"NO"	"NO"
9 TaqMan(v_0, v_3, v_5)	0.017352	0.982650	"NO"	"NO"
10 TaqMan(v_0, v_4, v_5)	0.967440	0.032556	"YES"	"YES"

TABLE 3. Partition matrix value for each TaqMan reaction based on AFCM clustering algorithm

TaqMan reaction	μ_{1j}	μ_{2j}	Manual observation	Reaction ($y_{1(46)} > y_{2(46)}$)
1 TaqMan(v_0, v_1, v_2)	0.526750	0.473250	"YES"	"YES"
2 TaqMan(v_0, v_1, v_3)	0.939490	0.060506	"YES"	"YES"
3 TaqMan(v_0, v_1, v_4)	0.906020	0.093975	"YES"	"YES"
4 TaqMan(v_0, v_1, v_5)	0.970780	0.029225	"YES"	"YES"
5 TaqMan(v_0, v_2, v_3)	0.584040	0.415960	"YES"	"YES"
6 TaqMan(v_0, v_2, v_4)	0.000716	0.999280	"NO"	"NO"
7 TaqMan(v_0, v_2, v_5)	0.848600	0.151400	"YES"	"YES"
8 TaqMan(v_0, v_3, v_4)	0.002650	0.997350	"NO"	"NO"
9 TaqMan(v_0, v_3, v_5)	0.000141	0.999860	"NO"	"NO"
10 TaqMan(v_0, v_4, v_5)	0.618130	0.381870	"YES"	"YES"

4. **Discussion.** In the *in-vitro* implementation, the amplification responses observed in this study differ from the amplification responses, which are typically obtained in the life sciences and medicine. In particular, while in the life science and medical applications, the initial copy number of the DNA template is normally very low but in the current study the input molecule is actually a DNA species extracted from a polyacrylamide gel, which exists at a high concentration. That is the main reason why the amplification signals in the current study appeared more rapidly than normal.

From the implementation FCM and AFCM, it was shown that AFCM produced all corrects classification of "YES" and "NO" reactions, compare to FCM. In [6], the result from real-time PCR is correctly classified into "YES" and "NO" groups based on FCM clustering algorithm. Meanwhile, implementation of FCM in this research produced two errors, where in Table 2, TaqMan1 and TaqMan 5 were wrongly classified. This is because, FCM is sensitive to noise and outliers [11], which result errors in the automatic classification algorithms. To avoid noise and outliers conditions, several techniques have been introduced to increase the robustness of the algorithms for clustering of object data, such as Possibilistic C-Means (PCM) [8] and fuzzy noise-clustering approach [9]. However, the performance of PCM depends ona good initialization as well asaccurate estimation of resolution or scale parameter, η [8]. In noise-clustering approach, the noise distance parameter, δ, is a user specified parameter and the clustering results could be sensitive

tovariations in the noise distance [10]. AFCM has been chosen for the implementation of data clustering during the *in silico* phase since the exponential distance is robust, in terms of handling of outliers and noises. Furthermore, AFCM does not require any parameter estimation compared to PCM and noise-clustering approach.

5. **Conclusions.** This research offers an automation of real-time PCR-based readout approach for DNA computing, which is implemented on DNA Engine Opticon 2 System. In the *in vitro* phase of the readout approach, each real-time PCR reaction is mapped to a binary output ("YES" or "NO"), based on the occurrence or absence of an exponential amplification. In the *in silico* phase, by applying the AFCM clustering algorithm to the output of real-time PCR, two different TaqMan reactions, "YES" and "NO", can be clearly distinguished. Then, the subsequent *in silico* information processing is capable of determining the Hamiltonian path of the input instance.

Acknowledgment. This research is supported financially by the University of Malaya Post Graduate Research Grant (PPP PS078/2010B), University Malaya Research Grant (UMRG-025/09AET), University Malaya/Ministry of Higher Education (MOHE) High Impact Research (HIR) Grant(D000016-16001) and Ministry of Science, Technology, and Innovation (SciendFund MOSTI-13-02-03-3075). Muhammad Faiz Mohamed Saaid is indebted to University of Malaya for granting him a financial support under Bright Spark Scheme University of Malaya (SBSUM) and opportunity to do this research. The authors also would like to thank Universiti Teknologi Malaysia for supporting this research by UTM GUP research grant (Vote Q.J130000.7123.00H67).

REFERENCES

[1] L. M. Adleman, Molecular computation of solutions to combinatorial problems, *Science*, vol.266, pp.1021-1024, 1994.

[2] L. Kari, DNA computing in vitro and in vivo, *Future Generation Computer System*, vol.17, pp.823-834, 2001.

[3] C. V. Henkel, *Experimental DNA Computing*, Ph.D. Thesis, Leiden University, 2005.

[4] D. H. Wood, C. L. T. Clelland and C. Bancroft, Universal biochip readout of directed hamiltonian path problems, *Lecture Notes in Computer Science*, vol.2568, pp.168-181, 1999.

[5] Z. Ibrahim, J. A. Rose, A. Suyama and M. Khalid, Experimental implementation and analysis of a DNA computing readout method based on real-time PCR with TaqMan probes, *Natural Computing Journal*, vol.7, no.2, pp.277-286, 2008.

[6] M. F. M. Saaid, Z. Ibrahim, Z. M. Yusof and J. Watada, Automation of a DNA computing readout method based on real-time PCR implemented on a LightCycler system, *International Journal of Innovative Computing, Information and Control*, vol.6, no.10, pp.4263-4272, 2010.

[7] J. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.

[8] R. Krishnapuram and J. M. Keller, A possibilistic approach to clustering, *IEEE Trans. Fuzzy Syst.*, vol.1, no.2, pp.98-110, 1993.

[9] R. N. Dave, Characterization and detection of noise in clustering, *Patt. Rec. Letters*, vol.12, pp.657-664, 1991.

[10] A. Banerjee and R. N. Dave, The fuzzy mega-cluster: Robustifying FCM by scaling down memberships, *Lecture Notes in Computer Science*, vol.3613, pp.444-453, 2005.

[11] K. L. Wu and M. S. Yang, Alternative C-means clustering algorithm, *Pattern Recognition*, vol.35, pp.2267-2278, 2000.

[12] Z. Chien, T. Qiu and S. Ruan, A segmentation algorithm for brain MR images using fuzzy model and level sets, *International Journal of Innovative Computing, Information and Control*, vol.6, no.12, pp.5565-5574, 2010.

[13] M. Zhang and K. Zou, The application of fuzzy clustering after improvement on speaker recognition, *ICIC Express Letters*, vol.2, no.3, pp.263-267, 2008.