# DNA Strand Displacement for Stochastic Decision Making based on Immune's Clonal Selection Algorithm

**Rizki Mardian, Kosuke Sekiyama, Toshio Fukuda**

***Abstract:*** *Recently, developments on DNA-based molecular system have been conducted, including information sensing, locomoting function, decision making, and so on. Stochastic decision making is an important mean to deal with the adaptability under the fluctuating environment and it has to be processed based on DNA computing in molecular level. Many applications are found in DNA brain, DNA computer, molecular robotics, etc. In this research, DNA Strand Displacement is chosen as a main framework to develop a nucleic acid based decision making. The idea of implementation is inspired by the principle of Clonal Selection Algorithm of the Immune System. The mechanism allows the system to self-organize an emergent behavior where DNA strands locally interact with each other according to Watson-Crick complementary process. We define some DNA-based operations based on DNA Strand Displacement reaction to represent the stochastic decision process with multiple choices of actions. Through this algorithm, DNA agents select the best action for a given problem. The software-based simulation results imply the successful implementation of the proposed DNA-based decision making scheme.*

***Keywords:*** *DNA Strand Displacement, DNA Computing, Stochastic Decision Making, Immune System, Clonal Selection Algorithm.*

## Introduction

In this research, we employ a DNA manipulation technique, referred to as DNA Strand Displacement [Zhand et al, 2007] to develop an artificial intelligent system. Various applications have been proposed on this mechanism, including: chemical reaction network [Yin et al, 2008; Soloveichik et al, 2010], logic circuit [Seelig et al, 2006; Qian & Winfree, 2011] as well as DNA brain [Qian et al, 2011], nano-motor and molecular robotics [Yin et al, 2004; Omabegho et al, 2009; Wand et al, 2010; Gu et al, 2010; Mardian et al, 2011]. Several distinguished results have been reported. For example, the DNA walker is designed to follow a predetermined path like the locomotion of autonomous robot [Yin et al, 2004; Omabegho et al, 2009]. The unique thing is that the behavior of the mobile DNA is not explicitly programmed, but it is driven by the peculiar structure interacting with the environment. This indicates the capability of information sensing as well. In other reports, the capability of a tweezer-shaped and a forklift-shaped DNA to interact with other biological materials opens the possibility for autonomous actuation in the molecular level [Wand et al, 2010; Gu et al, 2010]. However, most of these achievements still focus on primitive mechanical function.

Despite of its importance, there still not so many reference are found that address the issue of computational design of the DNA system. The compilation of multi-agents scheduling task based on DNA combinatorial is proposed in [Mardian et al, 2011]. However it is still lack with autonomy level, leaving a question mark about the potential of DNA-based system to carry a more complex task. In this work, we focus on the high-level information processing based on DNA Strand Displacement, to develop a stochastic decision making scheme. This is expected to introduce a new ability to choose the best response from the available option and it will offer a certain degree of autonomy to the DNA-based molecular system.

In this work, we implement a nature-inspired self-organizing method to achieve the capability of DNA agent in determining its best action to a given problem. We find this process is analogous to the scenario of immune system mechanism, particularly the clonal selection algorithm. When we are encountered by some pathogenic molecules from outside, our body will trigger a self-defense mechanism in order to fight against the harmful effect of those substrates. The interesting thing is that even there are tons of possibilities of foreign molecules intruding to our body, the immune system can respond accordingly [De Castro & Timmis, 2003]. This explains the adaptability of this system to the stimulus from the environment. Human's immune system consists of various components which work

together in order to provide the defense mechanism under many conditions. As there is no central rule to command the way the immune system provides the response, the system is solely driven by the distributed interaction between its components. It is self-organized, hence the principle can be adopted into any swarm intelligent system, like DNA molecular system.

To the best of our knowledge, this is the first work that integrates a DNA-based computation with the immune system computational method which has already been well-known in computer science study. For the long term, this work is also expected to promote a learning mechanism, planning and autonomous control of the DNA system.

The rest of this paper will be organized as follows. Section 2 describes brief theory of Natural Immune System. Section 3 explains the DNA Strand Displacement reaction and how it can be coded into stochastic computational DNA operations. Section 4 discusses the decision making scenario as well as the coding and implementation, followed by the simulation result. Finally, conclusions are summarized in Section 5.

## Natural Immune System

Immune system is a vast field in medical science study. In this paper we focus on the particular type referred to as Adaptive Immune System, which encompasses the recognition and adaptation of immune system to the specific type of foreign molecules attack. The main actor in this process is called the lymphocytes, a type of white blood cell specialized to neutralize pathogenic reaction. There are two types of lymphocytes namely B-Cell and T-Cell. Both of these cells perform different function even they were originated from the same place, called Bone Marrow. Some of lymphocytes then pass and get matured in another place called Thymus and become T-Cell, while the rest become B-Cell.

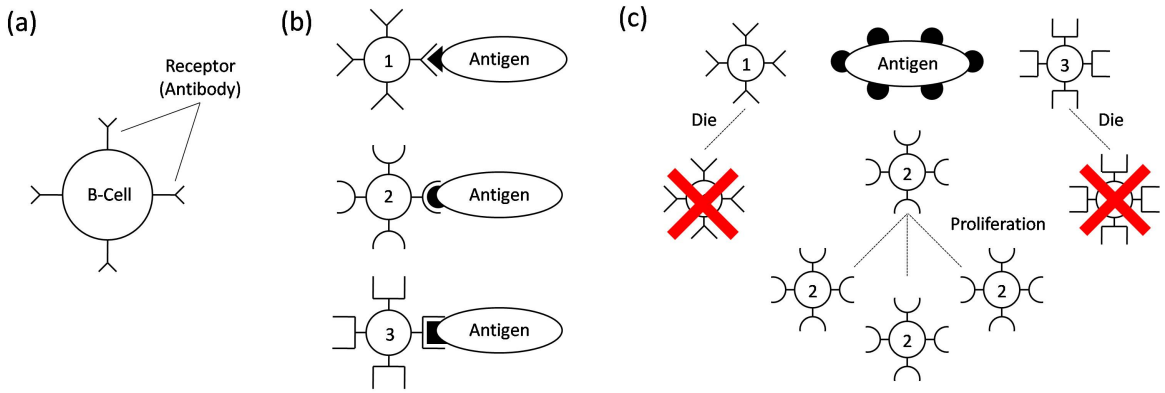


Figure 1: Natural Immune System: (a) Schematic of Antibody, (b) Matching Between Antigen - Antibody, (c) Clonal Selection Process

The whole process of this defense mechanism is depicted by fig.1. The lymphocytes contains some kind of receptors on their surface. In B-Cell, these receptors are called antibody (fig.1(a)), which can recognize and bind a particular part of the pathogenic molecules called antigen. This recognition process might be analogous to the relation between lock and key. There are a lot of types of B-Cell receptors, however they can perfectly bind to only a specific antigen that matches with them (fig.1(b)). After the binding between antibody and antigen, T-Cell will trigger the destructive reaction to nullify the foreign molecules attack. In addition to that, T-Cell also has important role to activate the B-Cell, particularly the best-matched one with the antigen. This exclusive process is called as Clonal Selection (fig.1(c)). Since the number of the best-matched B-Cell is stimulated through the differentiation process, the chance of the cells binding and destructing the pathogen will increase as their population increases. Meanwhile, the other type of B-Cell will die in the absence of interactions with antigen.

In the following section, we will discuss how to implement this simple self-organizing interaction methapor into a stochastic decision making scheme based on DNA reaction. For simplicity, we assume that B-Cell and antibody are regarded the same as there is no special need to distinguish them. We can simply ignore the important role of T-Cell

by assuming that the binding between antigen and antibody will always result in the destruction of the pathogenic. As for the foreign molecules, we can simply refer them as the antigen.

## DNA Strand Displacement Operation

In this section, the computational operation based on DNA Strand Displacement reaction is presented.

### Basic of DNA Strand Displacement

DNA Strand Displacement is a bio-chemical reaction, caused by random-walk, in which single-stranded DNA reacts with a multi-stranded DNA to produce the other single-stranded DNA [Seelig et al, 2006]. This process can be viewed as a computational mechanism working under bio-chemical environment, with DNA as medium to carry the structure as well as the information. Among all of the DNA manipulation techniques available so far, DNA Strand Displacement is said to be superior as it does not require any different molecules design except the strands itself, such as the restriction enzyme. Therefore, it is suitable even for a large application as the effort of preparing DNA on the tube will not become much more complicated as the scale increase[Qian & Winfree, 2011].

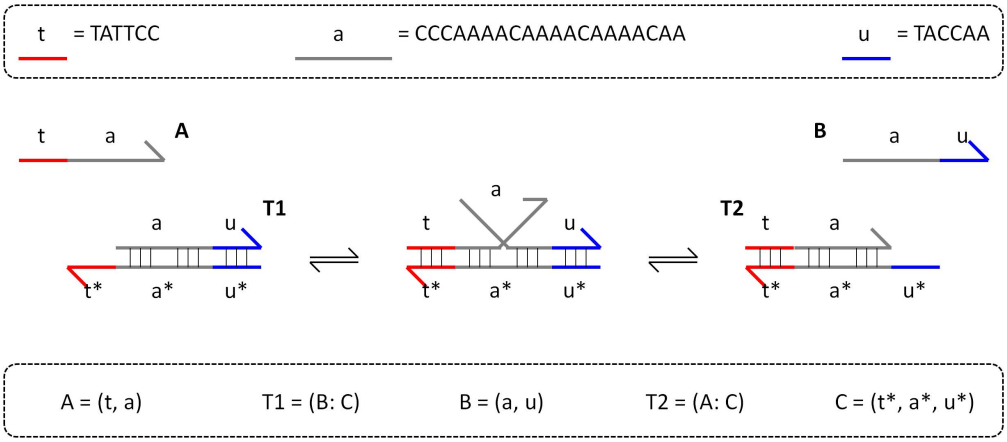


Figure 2: Basic of DNA Strand Displacement Reaction

The basic reaction of DNA Strand Displacement is described by fig.2. Abstractly, a DNA strand can be viewed as chunked domains with arbitrary alpha-numeric character to represent the encodings. By doing this, we can avoid the need to work directly with DNA nucleotide sequences, but instead we treat the domains as the simplest functional unit of the computation. One same character consistently represents the exact nucleotide sequence, with the * mark represent its complements (A with T, C with G and vice versa). We categorize the domains into two different types depending on the sequence length. The shorter (around 4-5 nucleotides) is referred to as toehold which is represented by the colored line, while the longer (around 20 nucleotides or more) is called as non-toehold which is represented by the grey line. This distinction is on purpose. Since the reaction happens by chance, the shorter sequence will reversibly bind faster than the longer one. As a result, any free toehold in the system can trigger the whole DNA Strand Displacement reaction to begin, while the non-toehold will provide the binding power as it does not easily unbind. Once a toehold from a single-stranded DNA is bound to its free complement in a multi-stranded DNA, it will alter its adjacent domain to also bind with the following sequences by entropy-difference, and to push out the old bindings from the multi-stranded. In this step, only if all the nucleotides match then the reaction can irreversibly proceed. As a result, a new single-stranded DNA will be released and replaced from the multi-stranded DNA through the process called as branch-migration [Zhand et al, 2007].

In fig.2, the reaction is equivalent to a computational procedure that transforms signal A (consisting of domain $t$ and $a$) into signal B (consisting of domain $a$ and $u$) through the binding to a multi stranded DNA T1 (which in turn is transformed into T2). In the end of this process, strand B can also perform a different reaction to T2 to produce strand A and T1 again. Thus, this reaction is considered as a reversible reaction. An irreversible DNA Strand

Displacement reaction can be achieved by making sure that there is no free toehold in the end of the process, so the multi stranded DNA will become a waste.

**Petri Net based DNA Operator**

Formally, we can compile DNA Strand Displacement reaction into an abstract representation which is more appealing for modeling high-level processes. In this work, we follow the outline as in [Mardian et al, 2011], which translates the DNA combinatorial into a process diagram, such as Petri Net. This modeling diagram has been used to explain concurrencies in many applications, including the description of chemical reactions [De Castro & Timmis, 2003], and a coordination of multi-agents systems [Angeli et al, 2007; Andersen et al, 2004], etc.

Petri Net successfully models the interaction among the DNA strands within a DNA Strand Displacement system, which is feasible for our current purpose. Generally, the binding of m-number of single-stranded DNA inputs and the release of n-number of single-stranded DNA outputs can be viewed as an extension from m-number of processes into n-number of processes (referred to as m-to-n operator, with m and n are integer numbers). This is depicted by graphical notation in fig.3(a), in which the circular shape represents the process and the rectangular shape represents the transition between processes. The dynamics of the system is depicted by the marking token given into processes which denotes those are currently running or the current state of the system. By applying a simple rule, if every process connected to any transition contains the token, the process can then be extended; we can build a discrete event-driven network which captures the interaction between the processes. This notation can be implemented by DNA Strand Displacement reaction as shown in fig.3(b).

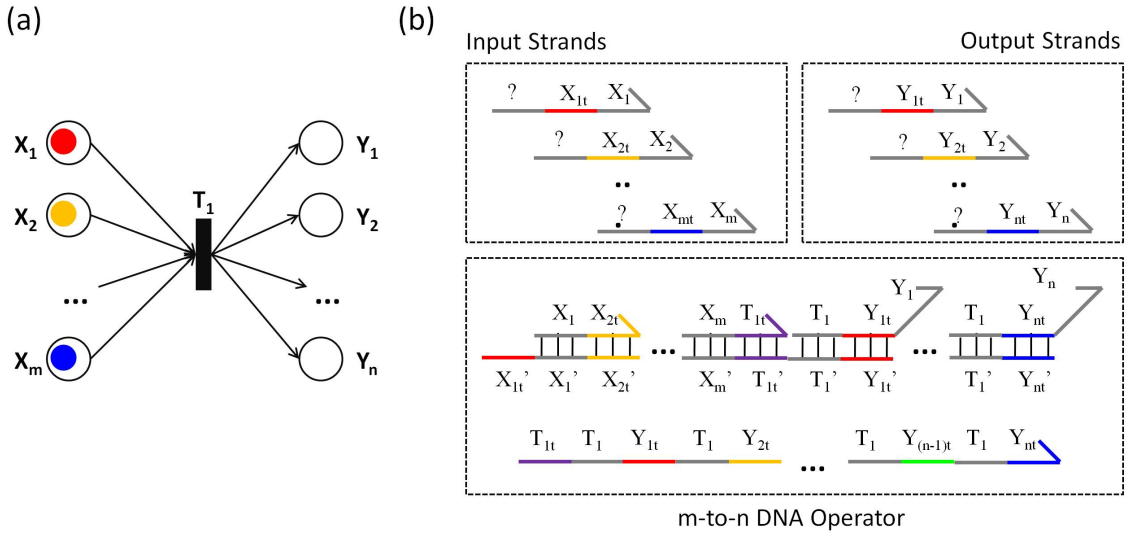


Figure 3: (a) Abstraction of DNA Displacement Reaction into Process Diagram, (b) Implementation of m-to-n Operator [Mardian et al, 2011]

The processes are encoded as a set of single-stranded DNA (input and output strands in the upper box) and the transitions as a set of multi-stranded DNA (also consists of another longer single-stranded DNA). This is based on the idea that in both cases multi-stranded DNA and transition act as "fuel" that facilitate the change of the systems state. On the other hand, single-stranded DNA can be coded uniformly, thus can be used as the information storage. The tokens are all free single-stranded DNA at particular time (we refer to as active strands; single-stranded DNA is inactive when it is bound to the multi-stranded).

The design of the multi-stranded DNA depends on the single-stranded DNA which is supposed to bind and to release. In fig.3(b), the left part of the structure consecutively follows all the input strands from $X_1$ to $X_m$. When a single stranded DNA binds, it will free the next toehold so that the rests can also react (it displaces another single stranded that can be considered as a waste). For example, strand $X_1 = (?, X_{1t}, X_1)$ will detach a waste $I_1 = (X_1, X_{2t})$ and free the toehold $X_{2t}$. In the next step, strand $X_2$ binds and does the same thing for the next

input strand, and so on. When the inputs are complete, a particular toehold that is assigned uniquely for each multi-stranded DNA will be freed. It is supposed to react with the long single-stranded DNA which in turn will release all the output strands from $Y_1$ to $Y_n$. Thus, the design of this strand, as well as the right part of the multi-stranded DNA structure, depends on all the output strands. This means that the complete reaction can only be achieved as long as the all the input strands are given, otherwise the reaction will be reverted. This is similar to the idea of the extension of the marking tokens in the Petri Net semantics. By giving all the active strands and multi-stranded DNA as the initial condition, a reaction network can be designed to achieve some intended behavior [Soloveichik et al, 2010].

The compilation of DNA-based Petri Net can be written in a formal notation. For example, Petri Net as in fig.3 is equivalent to the algebra given by eq.(1).

$$\begin{aligned} X_l|X_2|...|X_m|T_1 &\mapsto ? \\ T_1 := [X_l, X_2, ..., X_m].[Y_l, Y_2, ..., Y_n] & \\ X_l|X_2|...|X_m|[X_l, X_2, ..., X_m].[Y_l, Y_2, ..., Y_n] &\mapsto Y_l|Y_2|...|Y_n \end{aligned} \tag{1}$$

Symbol '|' means the mutual existence of DNA strands, while symbol '$\mapsto$' means the reaction happens giving the initial condition. According to fig.3, single-stranded DNA $X_1, X_2, ..., X_m$ act as the inputs together with multi-stranded DNA $T_1$ (first line). The design of $T_1$ can be explained by the strands it binds and the strands it releases, represented by symbol '.' (second line). The set of input and output strands are separated by symbol '[ ]'. Thus, the result from this reaction can be inferred as $Y_1, Y_2, ..., Y_n$ (third line). In case of one strand is missing, there will be no result is produced as this specification is not met.

**Stochastic DNA Operator**

In this paper, we extend the works from [Mardian et al, 2011] by introducing a new model of stochastic DNA operation. As shown in fig.4(a), one process can be extended into two processes connected by two different transitions giving the possibility either to go from process A to process B1, or from process A to process B2. This is in constrast with 1-to-2 operator [Mardian et al, 2011], which is depicted by fig.4(b). In that operator, one process is forked into two different processes but with no conditional choice is occured. Our implementation, however, resembles of the two 1-to-1 operators which take inputs from the same processes that only one of them will be executed at the same time. As this mechanism works by chance, we refer this implementation as Stochastic DNA Operator. In normal condition, both $T_1$ and $T_2$ have the uniform 50-50 probability to proceed.

The decision making in this operation can be implemented by population control method to increase the likehood to proceed one process than the other. Basically, DNA Strand Displacement is a probabilistic reaction happening by chance. The more DNA strands population are given, the better chance the reaction will actually happen. This means, if the number of $T_1$ is given by bigger amount, i.e 3 * $T_2$ population, the transition probability will change into 75-25. Therefore, our objective is to dynamically maximize the population number of the intended transition.

In many implementation of DNA Strand Displacement so far, it is a lot easier to manipulate single-stranded DNA instead of multi-stranded DNA directly [Qian & Winfree, 2011; Mardian et al, 2011; Cardelli, 2009]. One suggestion is by buffering technique of multi-stranded carrier which intends to make particular multi-stranded DNA "partially inactive" until the designed single-stranded DNA releases it through reaction (normally act as additional output strand from previous reaction). By doing this, an operator which produces multi-stranded DNA can be designed [Cardelli, 2009]. However, it is still difficult to cascade this output in multi-layered reaction.

Our implementation suggests the usage of the 2-to-1 operators instead. As shown in fig.4(c), some modification are done by adding additional single-stranded DNA for each of transition which is represented by smaller and colored-token containing processes, also referred to as implicit states of the operator. This is logically identical with the operator as in fig.4(a). By doing so, instead of directly control the number of multi-stranded DNA, we can control the number of single-stranded DNA those represent the implicit states to yield the same result. Note that now multi-stranded DNA can be separately viewed as the mediator-only for the reaction. Its population number has no direct influence for the system dynamics as long as it is maintained uniform. Thus, to ensure the correctness of the

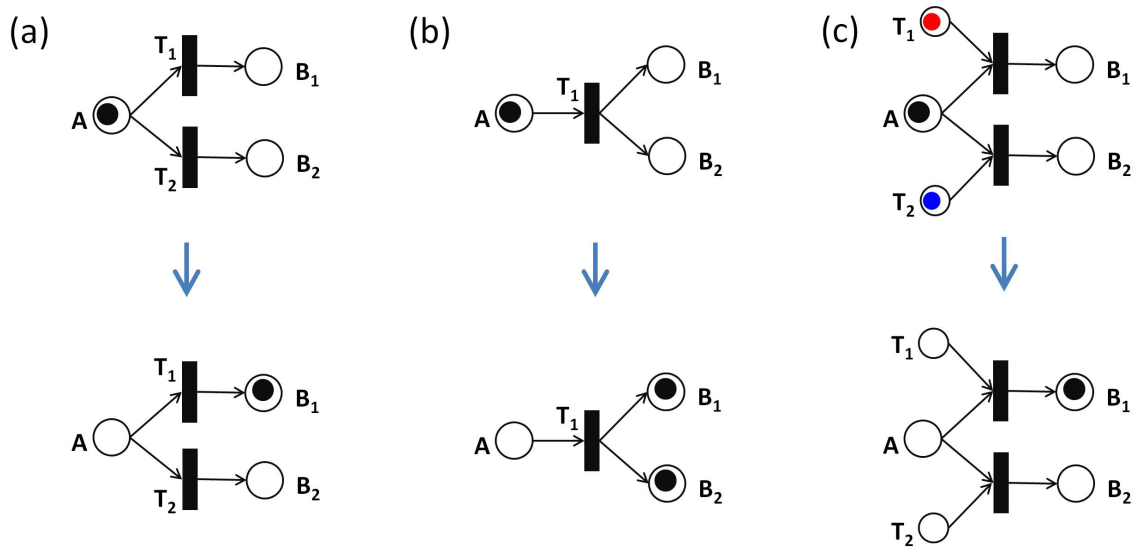


Figure 4: (a) Basic Idea of Stochastic DNA Operator, (b) Comparison with Normal 1-to-2 Operator [Mardian et al, 2011], (c) Stochastic DNA Operator by Implicit State

reaction, the population of them assumed unbounded or reasonably very large that it will not get exhausted before the single-stranded itself.

## Implementation and Results

In this section, the implementation of the stochastic decision making scheme by DNA reaction is discussed. To evaluate the scenario, a software-based simulation is conducted by Microsoft Visual DSD Simulator [Phillips & Cardelli, 2009].

### Scenario of Stochastic Decision Making

By using the Stochastic DNA Operator, we can model the scenario of decision making of DNA molecular agent. Schematically, this can be seen as a mapping function between a set of action strategies to a set of problems, with some weight value represents the strength of the relation (see fig.5(a)).

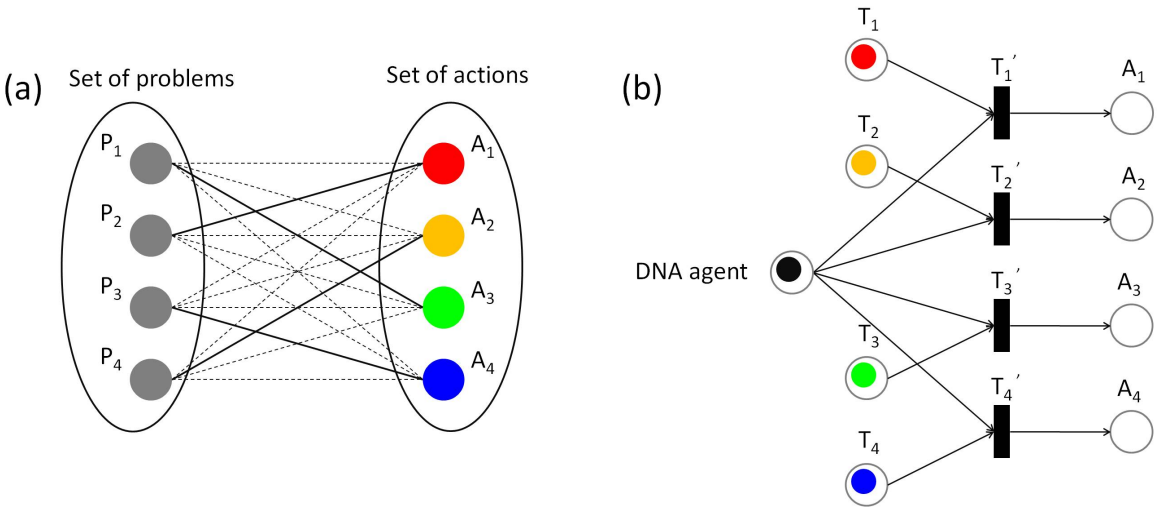


Figure 5: (a) Mapping function between the set of problems and actions, (b) Schematic of Decision Making Module of the DNA Agent as Stochastic DNA Operator

Supposing that there are set of problems $P = \{P_1, P_2, P_3, P_4\}$ and set of actions $A = \{A_1, A_2, A_3, A_4\}$, such there exists the weight function $W = P \times A$ which is represented by the connection between each of action and problem. The best relation is captured by the bold line, while the dashed line represent the rest; the configuration in this example is arbitrarily determined. Thus, set of solution $S = \{(P_2, A_1), (P_4, A_2), (P_1, A_3), (P_3, A_4)\}$. Decision making problem in this scenario is like giving a query for DNA agent to find the pair of the best action from the given problem. For example, by introducing $P_2$, the agent is supposed to make a decision of $A_1$. By stimulating $P_1$, action $A_3$ will be taken, and so forth. Currently, we assume that only one problem is given at the same time.

The implementation of stochastic DNA operator can be found in the decision making module itself (see fig.5(b)). Supposed that we have a DNA molecular agent which represent the current state of the system. This is based on the selected action in the previous step or also can be an initial condition. By giving an arbitrary problem as the stimulus, it is clear that our objective is to maximize the number of corresponding transition's population (or single-stranded DNA which represents a implicit state connected to it) so that the DNA agent can be directed into the best action. In the next iteration, this action will be a new state of the agent itself, and the process can be repeated until the main goal is accomplished. It is important to note that this scenario is expected to be adaptive. Whenever there is a change in the stimulus, the system should respond accordingly and to change the distribution of DNA strands population as well, so that the new best action can be promoted. This problem can be solved by taking methapor of Natural Immune System as the detail will be discussed in the following section.

**From Immune System to DNA Coding**

The first thing to do in implementing stochastic decision making scheme based on Natural Immune System is to decide the encoding of every component into DNA structures. As briefly explained before, we only consider antigen and antibody as the main components in our system. We neglect the importance of B-cell's and T-cell's roles as well as to simplify the pathogen molecules as the antigen. Thus, we employ the metaphor of the interaction between antigen and antibodies instead of the pathogenic and lymphocytes.

Antigen acts as a stimulus for the antibodies. Whenever the body, in this case DNA agent, is introduced with particular type of antigen, the corresponding type of antibody will be stimulated to increase their number in order to fight back. From this analogy, there are two scenarios that can be modeled. First, antigen is the representation of the agent's objective or problem to be solved, while antibodies are the action strategies to be taken. Second, the relationship between antigen and antibody, especially among the match-type, is like a predator and prey. The existence of antigen will be a 'food source' for particular antibody, and the reaction between both will nullify the number of antigen.

In addition to that, we have to be able to measure the distance between antigen and antibody as the weight function in fig.5(a) (also referred to as "affinity" in following the term from Natural Immune System). This represents how effective one antibody fights the antigen. By default, the pairing of nucleotides in DNA Strand Displacement is a discrete matching problem; in the sense that only when all nucleotides match then the reaction will proceed, or else it will be reverted. Thus, instead of coding the antigen and antibody by one single-stranded DNA, we will use the term of "collection-of-strands" which consists of many single-stranded DNA. For the notation, the lowercase alphanumeric character represents antigen and the uppercase alphanumeric character as antibody. For example, $p = \{u_1, u_2, u_3, ..., u_i\}$ means antigen $p$ is represented by $i$-number of single-stranded DNA from $u_1$ until $u_i$; and $A = \{V_1, V_2, V_3, ..., V_j\}$ means antibody $A$ is represented by $j$-number of single-stranded DNA from $V_1$ until $V_j$; with $i$ and $j$ are arbitrary integer numbers. The longer collection will make more combinations. In our model, we set $i = j = 4$ as the collection-of-strands for each antigen and antibody.

The affinity between antigen and antibody can be observed on how many from both collection-of-strands match each other. We can utilize 2-to-n operator to define the reaction between match pair, with $n$ represents the number of output are expected. Antibody possessing the collection-of-strands with highest affinity will be selected as the best choice. For example, in the case where antigen $p$ matches with antibody $A$, we should have $i$-number of 2-to-n operator, with $i \leq j$ and $n$ represents any number of output strands are expected. We can encode other antibodies as different collection-of-strands' combination, i.e: antibody $B = \{V_{1*}, V_{2*}, V_3, ..., V_j\}$ and $C = \{V_{1*}, V_{2*}, V_{3*}, ..., V_{j*}\}$.In this case, antibody $B$ posesses a high affinity to antigen $p$ (even not exact match since

there are two differences) compared than antibody $C$. Antibody $A$ is the best response to antigen $a$. However, in the absence of that, antibody $B$ may serve as the best choice.

It is important to make sure that the collection-of-strands from antibody does not mix each other, in order to preserve the clear coding. The easiest way to do so, these strands are made partially inactive until they are needed for reaction, initially in the form of multi-stranded DNA. It can be activated by additional single-stranded DNA called "trigger strand". Thus, the definitions of antigen and antibody index $k$-th are given by eq.(2), which implements 1-to-n operator, with n is the length of the collection-of-strands (in our case, n=4).

$$\begin{aligned} trigger^{antigen}|[trigger^{antigen}].[u_1,u_2,u_3,u_4] &\mapsto u_1|u_2|u_3|u_4 \\ trigger^{k}|[trigger^{k}].[V_1^k,V_2^k,V_3^k,V_4^k] &\mapsto V_1^k|V_2^k|V_3^k|V_4^k \end{aligned} \tag{2}$$

The rest of the coding is to determine the output strands. Our purpose is to increase the population of single-stranded DNA represents the implicit state of the selected transition. Thus, this strand will be produced by the reaction between antibody and antigen the transition leads. The operators are defined as eq.(3). It can be read as the binding of single-stranded DNA index $l$-th from both antigen and antibody index $k$-th to release another single-stranded DNA represents the implicit state of transition index $k$-th. Intrinsically, the more collection-of-strands from an antibody binds to the antigen, the more the corresponding transition will be produced. As a result, it will increase the likehood of that transition to be executed.

$$V_l^k|u_l^k|[V_l^k,u_l^k].[T_k] \mapsto T_k \tag{3}$$

As the structure for antigen-antibody reaction has been defined, the last part is to implement the module for decision making itself. This is quite trivial as we can map the stochastic DNA operator directly which can be written as in eq.(4). In this case, the implementation are based on 2-to-1 operators. $T_k$ is an implicit state index $k$-th, $Internal$ is the DNA agent's internal state, and $trigger^k$ is the trigger strand with the same definition as in eq.(3). From here, we can see that the number of trigger strand's population depends on the implicit state's strand. Meanwhile, the implicit state's population are controlled by the interaction between antigen and antibody. Therefore, DNA agent's internal state will gradually converge into an antibody that yields the best interaction. This implementation is equal to the schematic of the scenario as we expect in fig.5(b) before.

$$T_k|Internal|[T_k,Internal].[trigger^k] \mapsto trigger^k \tag{4}$$

Intuitively, all of those definition above already represents a stochastic decision model scenario based on Natural Immune System through DNA Strand Displacement reaction. DNA agent is expected to be able to respond reactively according to the given stimulus. However, when there is similar structure between antibodies, the final production result of each antibody will not be much different, which might leave a question about the effectiveness of making a decision in the actual problem. To encounter this problem, we suggest an additional rule. If the interaction between antigen and antibody is considered as a positive stimulation to converge to the best action, then we will add another component acts as negative suppression to depress the rest antibodies. This principle can be introduced through some 2-to-0 operators that naturally "kill" the implicit state $T_k$ population through the time. Unless they are produced rapidly through the interaction with antigen, the number populations are forced to decrease. These operator are defined as eq.(5), where $Killer$ is the new defined single-stranded DNA to control the rate of negative suppression.

$$Killer|T_k|[Killer,T_k].[] \mapsto \emptyset \tag{5}$$

**Simulation Result**

A software-based simulation is conducted by using Microsoft Visual DSD Simulator [Phillips & Cardelli, 2009] to evaluate the designed scenario. Visual DSD is a programming language for composable DNA circuits that includes

basic elements of sequence domains, toeholds and branch migration. It compiles a collection of DNA strands into a reaction network based on DNA Strand Displacement, also includes a stochastic simulator to graph the species' population over time. It takes the assumption that the strands do not possess any secondary structure. This tool had been employed to run and verified various actual implementations of DNA Strand Displacement systems [Mardian et al, 2011; Phillips & Cardelli, 2009; Lakin & Phillips, 2011], which motivate the usage of this simulator at this stage of our work.

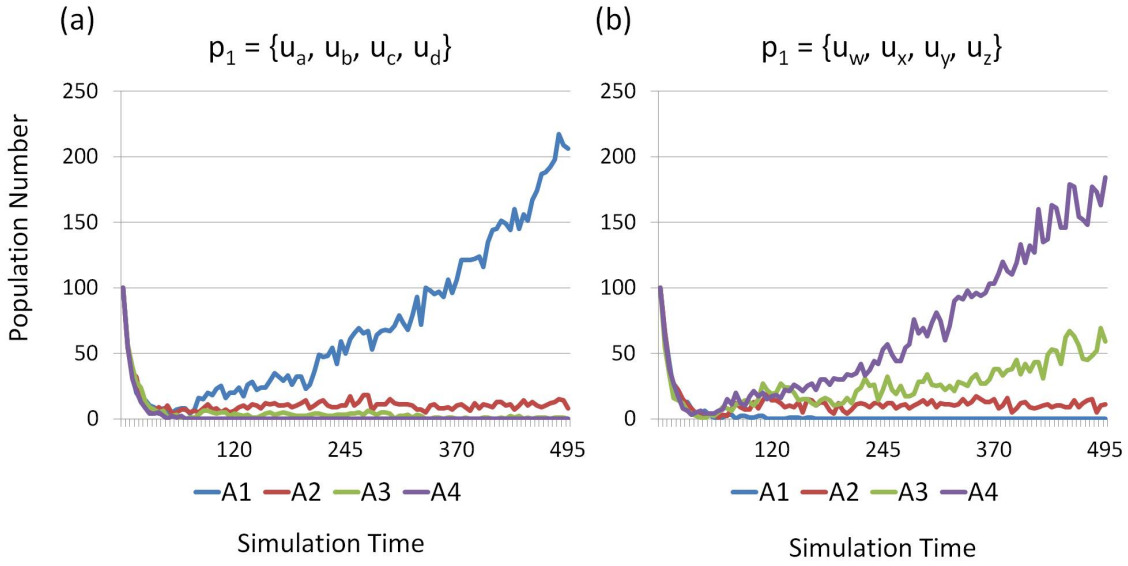


Figure 6: Simulation Result of Stochastic Decision Making Scheme

Consider there are 4 antibodies listed with random collection-of-strands structure, $A_1 = \{V_a, V_b, V_c, V_d\}$, $A_2 = \{V_w, V_x, V_c, V_d\}$, $A_3 = \{V_w, V_x, V_y, V_d\}$, and $A_4 = \{V_w, V_x, V_y, V_z\}$. Single-stranded DNA representing DNA agent, trigger strands and killer strand, together with all multi-stranded DNA are assumed to be in a very large number of initial population, that they will not get deprecated until the end of simulation (or asummed to be constantly refurbished). fig.6 shows the results from this implementation. In the left-side result, antigen $p_1 = \{u_a, u_b, u_c, u_d\}$ is introduced as the stimulus for DNA agent. The simulation yields the result that converges to antibody $A_1$ as the preferable action. This meets the expectation, as the structure of antibody $A_1$ is the best-match for antigen $p_1$. In the right-side, as the input is substituted to antigen $p_2 = \{u_w, u_x, u_y, u_z\}$, DNA agent makes a new choice to antibody $A_4$ as its best strategy. This decision successfully concludes the adaptability of this stochastic decision making scheme in accordance to the environmental changes.

Based on this result (fig.6(a)), we can observe the behavior of DNA agent as depicted by fig.7. It shows the cumulative of actions taken based on each of strategies population. As seen, the agent takes the highest population strategy more frequently than the others, depicted by strategy $A_1$. The partial reaction network of DNA Strand Displacement system that represents this stochastic decision making scheme in case of strategy (or antibody) $A_1$ is shown in fig.8. Each of block denotes every step in implementation from eq.(2) to eq.(4). Eq.(5) is omitted in this diagram as it happens in separate reaction. Note that since other antibodies also equally go through the same process, there will happen competition among them to produce their own trigger strands (in this example, $trigger_{a1} = (tvah, tvat, tva)$). The more the collection-of-strand of any particular antibody match with the antigen's, the more this strand will be produced. Finally, it will result in the faster increment of the corresponding antibody population.

## Conclusions

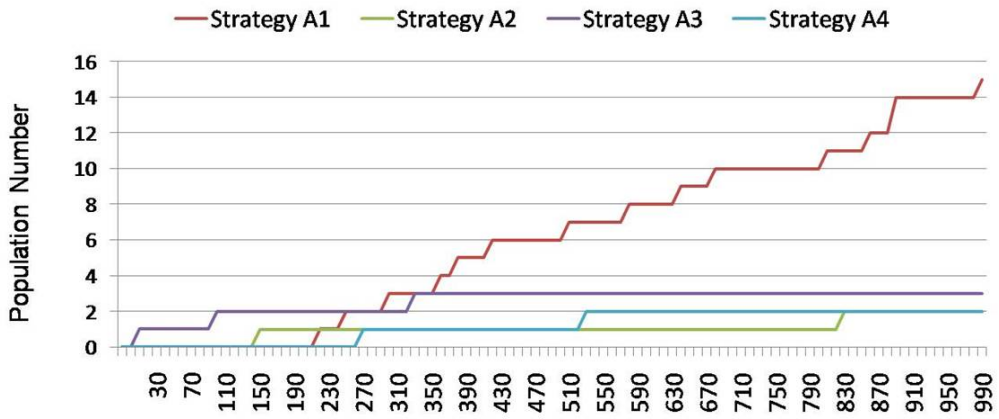


Figure 7: Cumulative of Action Strategies Taken by DNA Agent

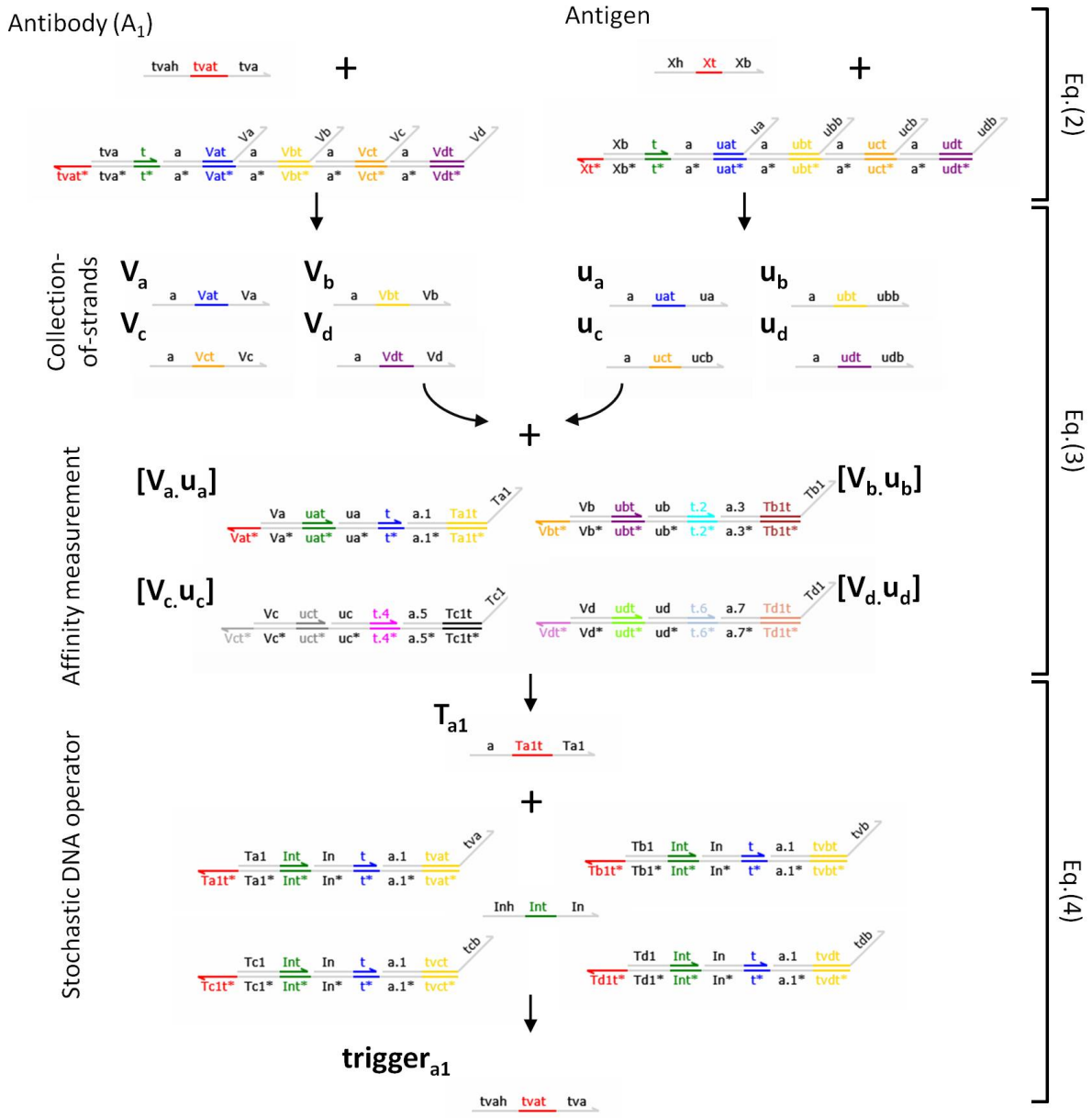


Figure 8: DNA Reaction Network of Stochastic Decision making (Antibody = $A_1$)

In this paper, a stochastic decision making scheme for DNA Strand Displacement system is implemented. When given a certain problem from the environment and several possible actions to take, DNA agent is supposed to select the best solution to deal with the problem. As the input of the problem can be dynamically substituted, the decision making module is expected to be adaptive to the change of stimulus. To solve this problem, we proposed the computational method based on Clonal Selection Algorithm of the Natural Immune System, which is an example of the self-organizing systems from nature. We designed the computational scenario above the modelling notation of Petri Net and evaluated the model through software-based simulation. From the observation, the behavior of DNA agent was illustrated.

This result provides some important findings to the development of DNA-based molecular system. Even this is still in early stage, the capability to change the decision accordingly can promote certain level of autonomy. Moreover, this achievement can contribute to the new DNA-based information processing method by using population control mechanism; as so far, most of DNA computational techniques rely on a stochastic reaction. Apart from that, it is also too early to conclude that our implementation had met what it is required for DNA device to work autonomously in a real task. One challenge to consider in the future is that currently there is still no mechanism to preserve information permanently within our DNA agent. The decision making process in our scenario requires a gradual effort that of course time and resource consuming. This might not be a problem in the infrequent occurence. However, in the real life, we often encounter the same problem after several times. If DNA agent has to come through the same process repeatedly, its effectivity will be reduced. The memory concept and learning mechanism for future DNA based intelligent system is inevitably important to develop.

## Bibliography

[Zhand et al, 2007] Zhang, D. Y., Turberfield, A. J., Yurke, B., Winfree, E.: Engineering Entropy-driven Reactions and Networks Catalyzed by DNA. Science 318. 1121-1125. 2007

[Yin et al, 2008] Yin, P., Choi, H., Calvert, C., Pierce, N.: Programming Biomolecular Self-assembly Pathways. Nature 451. 318-322. 2008

[Soloveichik et al, 2010] Soloveichik, D., Seelig, G., Winfree, E.: DNA as Universal Substrate for Chemical Kinetics. PNAS 107. 5393-5398. 2010

[Seelig et al, 2006] Seelig, G., Soloveichik, D., Zhang, D. Y., Winfree, E.: Enzyme-free Nucleic Acid Logic Circuits. Science 314. 1585-1588. 2006

[Qian & Winfree, 2011] Qian, L., Winfree, E.: Scaling Up Digital Circuit Computation with DNA Strand Displacement Cascades. Science 332. 1196-1201. 2011

[Qian et al, 2011] Qian, L., Winfree, E., Bruck, J.: Neural Network Computation with DNA Strand Displacement Cascades. Nature 475. 368-372. 2011

[Yin et al, 2004] Yin, P., Yan, H., Daniell, X. G., Turberfield, A. J., Reif, J. H.: A Unidirectional DNA Walker That Moves Autonomously Along a Track. Angewandte Chemie Int. Ed. 43. 4906 - 4911. 2004

[Omabegho et al, 2009] Omabegho, T., Sha, R., Seeman, N. C.: A Bipedal DNA Brownian Motor with Coordinated Legs. Science 324. 67-71. 2009

[Wand et al, 2010] Wang, Z. G., Elbaz, J., Remacle, F., Levine, R. D., Willner, I.: All-DNA Finite-state Automata with Finite Memory. PNAS 107. 21996-22001. 2010

[Gu et al, 2010] Gu, H., Chao J., Xiao, S. J., Seeman, N. C.: A Proximity-based Programmable DNA Nanoscale Assembly Line. Nature 465. 202-205. 2010

[Mardian et al, 2011] Mardian, R., Sekiyama, K., Fukuda, T.: Multiple Entities Interaction Model through DNA based Petri Net. 5th International Conference on Automation. Robotics and Applications. 2011

[De Castro & Timmis, 2003] De Castro, L. N., Timmis, J. I.: Artificial Immune Systems as a Novel Soft Computing Paradigm. Soft Computing 7. 526-544. Springer: Heidelberg. 2003

[Angeli et al, 2007] Angeli, D., Lenheer, P. D., Sontag, E.: A Petri Net Approach to Persistence Analysis in Chemical Reaction Networks. Bio. Ctrl. Theory: Current Challenges LNCIS 357. 181-216. Springer: Heidelberg. 2007

[Andersen et al, 2004] Andersen, M., Jensen, R., Bak, T., Quottrup, M.: Motion Planning in Multi-robot Systems Using Timed Automata. 5th IFAC/EURON Symposium On Intelligent Autonomous Vehicles. 2004

[Damas & Lima, 2004] Damas, B., Lima, P.: Stochastic Discrete Event Model of a Multi-robot Team Playing an Adversarial Game. 5th IFAC/EURON Symposium On Intelligent Autonomous Vehicles. 2004

[Cardelli, 2009] Cardelli, L.: Strand Algebras for DNA Computing. DNA-15 LNCS 5877. 12-24. Springer: Heidelberg. 2009

[Phillips & Cardelli, 2009] Phillips, A., Cardelli, L.: A Programming Language for Composable DNA Circuits. J. R. Soc. Interface. 2009

[Lakin & Phillips, 2011] Lakin, M. R., Phillips, A.: Modelling, Simulating and Verifying Turing-Powerful Strand Displacement Systems. DNA-17 LNCS 6937. 130-144. Springer: Heidelberg. 2011

## Authors' Information

***Rizki Mardian*** *- Master Student Candidate, Department of Micro-Nano Systems Engineering, Graduate School of Engineering, Nagoya University, Furo-cho, Chikusa-ku, Nagoya, Japan. 464-8601; e-mail: rizki@robo.mein.nagoya-u.ac.jp*

*Major Fields of Scientific Research: DNA Computing, Molecular Robotics, Self-organizing Systems*

***Kosuke Sekiyama*** *- Associate Professor, Department of Micro-Nano Systems Engineering, Graduate School of Engineering, Nagoya University, Furo-cho, Chikusa-ku, Nagoya, Japan. 464-8601; e-mail: sekiyama@mein.nagoya-u.ac.jp*

*Major Fields of Scientific Research: Multi-robot Systems, Self-organizing Systems, Intelligent Systems*

***Toshio Fukuda*** *- Professor, Department of Micro-Nano Systems Engineering, Graduate School of Engineering, Nagoya University, Furo-cho, Chikusa-ku, Nagoya, Japan. 464-8601; e-mail: fukuda@mein.nagoya-u.ac.jp*

*Major Fields of Scientific Research: Bio Inspired Robotics, Micro-Nano Manipulation, Multi-scale Intelligence*