A Molecular Evolutionary Algorithm for Learning Hypernetworks on Simulated DNA Computers

Ji-Hoon Lee
Interdisciplinary Program in Bioinformatics
Seoul National University
Seoul 151-742, Korea
jhlee@bi.snu.ac.kr

Bado Lee
School of Computer Sci. & Eng.
Seoul National University
Seoul 151-742, Korea
bdlee@bi.snu.ac.kr

Joon Shik Kim
Biointelligence Laboratory
Seoul National University
Seoul 151-742, Korea
jskim@bi.snu.ac.kr

Russell Deaton
Dept. of Computer Science and Engineering
University of Arkansas
Fayetteville, USA
rdeaton@uark.edu

Byoung-Tak Zhang
Biointelligence Laboratory
School of Computer Sci. & Eng.
Seoul National University
Seoul 151-742, Korea
btzhang@bi.snu.ac.kr

Abstract—**We describe a "molecular" evolutionary algorithm that can be implemented in DNA computing *in vitro* to learn the recently-proposed hypernetwork model of cognitive memory. The molecular learning process is designed to make it possible to perform wet-lab experiments using DNA molecules and bio-lab tools. We present the bio-experimental protocols for selection, amplification and mutation operators for evolving hypernetworks. We analyze the convergence properties of the molecular evolutionary algorithms on simulated DNA computers. The performance of the algorithms is demonstrated on the task of simulating the cognitive process of learning a language model from a drama corpus to identify the style of an unknown drama. We also discuss other applications of the molecular evolutionary algorithms. In addition to their feasibility in DNA computing, which opens a new horizon of *in vitro* evolutionary computing, the molecular evolutionary algorithm provides unique properties that are distinguished from conventional evolutionary algorithms and makes a new addition to the arsenal of tools in evolutionary computation.**

Keywords-molecular evolutionary algorithms, molecular evolutionary learning; DNA computing; hypernetwork; cognitive memory simualtion

I. Introduction

DNA molecules were verified as being useful materials in many fields. Many people showed the great interests in the ability of DNA molecules for information processing [1][2][3][4][5]. Adleman suggested that the DNA molecules can be used for computing devices by a solving complex combinatorial problem such as the Hamiltonian path problem [6]. DNA molecules are hybridized in massively parallel manner in test tubes. Even though each operation step is slower than the conventional computer, DNA computing can be faster if the coding of DNA sequences is well designed. After the report of Adleman, Baum recognized that the hybridization reaction of DNA molecules could be regarded as an associative and content addressable memory of immense capabilities [7]. Chen at al. showed that the associative memory and recall process can be implemented by DNA molecules [8]. In fact, content-addressable memories are widely used in a lot of computer contexts as in information retrieval, data compression, and database acceleration [9][10][11]. Content-addressable memory is also an important part of the human intelligence [12][13][14]. With these ideas, DNA molecules are attractive materials to realize a human-like intelligence through molecular machine learning.

The hypernetworks model is a molecular evolutionary architecture for human-like machine learning and cognitive memory [15]. A hypernetworks is a probabilistic graphical model based on hypergraph which consists of weighted hyperedges. Each hyperedge contains several vertices as features. The hypernetworks is trained by an evolutionary process using the DNA molecular operations of matching, selection, and amplification [15][16][17]. The hyperedges in hypernetworks are the molecular individuals. The weight of a hyperedge corresponds to the number of DNA molecule representing that hyperedge. Therefore the probability density of the hyperedges represents the population of DNA molecules. The hypernetworks model was inspired by molecular computing and theoretically realized in terms of molecular self-assembly [18]. However, previous works of molecular hypernetworks were implemented only *in silico* [18] [19]. We present improved and detailed *in vitro* experimental design for implementing molecular hypernetworks learning.

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MEST) (No. 2010-0017734), the IT R&D program of MKE/KEIT (KI002138, MARS), the Industrial Strategic Technology Development Program (10035348) funded by the Korean government (MKE), and the BK21-IT program. The ICT at Seoul National University provides research facilities for this study.

In this paper, we present a design of *in vitro* evolutionary DNA hypernetworks and *in silico* simulation for that design. The initial hypernetworks library consists of random DNA sequences, i.e., random hyperedges. Each variable or word in the training example would be replaced by nucleic acids alphabets. Every DNA hyperedge has special tag sequences to bind the two complementary DNA hyperedges during training. The random hyperedges in the initial library are matched to the hyperedges in the training examples and then separated by nanoparticles and electrophoresis. Separated hyperedges are amplified by polymerase chain reaction (PCR) machine. After the amplification step, the hyperedges are inserted into the initial library and the number of hyperedges in the library is normalized. After repeating the training process, the hypernetworks can be used as a molecular estimator. In the estimation step, DNA queries from an unknown corpus are matched to the hyperedges of the trained hypernetworks. Between the two classes of the hypernetworks, we can estimate that the class with more matched hyperedges is probably the class of the unknown corpus.

To test the reliability of above procedures, we programmed an *in silico* simulator. The number of copies of each DNA hyperedge was determined by an evolutionary learning process. The target of the evolutionary process is the density estimation of the probability distribution of a corpus. Each class of hypernetworks can be learned in an unsupervised manner. A corpus style estimation problem can be solved with the trained hypernetworks. We have investigated the cognitive learning problem using TV series "Friends" and "Prison break" via simulation study. Each drama corpus was trained by *in silico* simulator. The weight distribution of a hypernetworks represents the probability distribution of the scripts from one of the two TV series, and they can be used as an estimator to determine the style of an unknown corpus. The simulation results show that the DNA hypernetworks performs learning and estimation efficiently and has the property of usual machine learning algorithms.

The paper is organized as follows. Section 2 introduces the basic concepts of hypernetworks and evolutionary learning of molecular hypernetworks. We present the DNA hypernetworks algorithm both in the *in vitro* and *in silico* aspects. In section 3, the results of the simulation and the discussion are presented. Conclusions and future works are described in Section 4.

II. Basic Architecture of Hypernetwork

A hypernetwork is a probabilistic graphical model based on hypergraph models inspired by massively parallel DNA self-assembly process [15][20]. A hypernetwork H is defined as triple $H = (X, E, W)$, where $X = \{x_1, x_2, \ldots, x_i\}$, $E = \{E_1, E_2, \ldots, E_i\}$, and $W = \{w_1, w_2, \ldots, w_i\}$ are the sets of vertices, hyperedges, and weights, respectively. A vertex is a value of a attribute and a hyperedge is the combination of vertices. A hyperedge can contain more than two vertices. The number of vertices in a hyperedges is called order or cardinality. A hyperedge of order k is referred to as k-hyperedge.

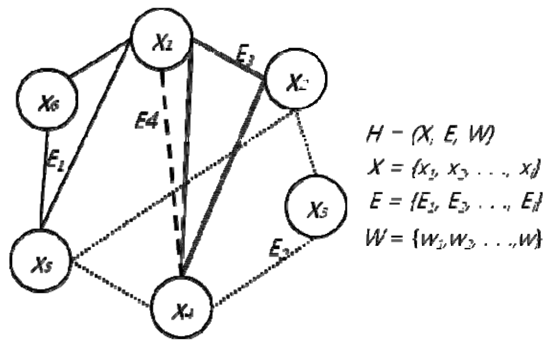


Figure 1. A hypergraph with four hyperedges. The orders of hyperedges E_1, E_2, E_3, and E_4 are 3, 4, 3 and, 2 respectively.

A hypernetwork can be considered as a probabilistic associative memory to store the part of a data set $D = \{\mathbf{x}^{(n)}\}_{n=1}^{N}$, where $\mathbf{x}^{(n)}$ denotes the n-th pattern to store. The energy of hypernetwork is defined as

$$\varepsilon(\mathbf{x}^{(n)}) = -\sum_{i=1}^{|E|} w_i f_i(\mathbf{x}^{(n)}) \tag{1}$$

The function f is defined as an indicator function which yields 1 in case that a hyperedge matches $\mathbf{x}^{(n)}$, otherwise 0. Given a data set $D = \{\mathbf{x}^{(n)}\}_{n=1}^{N}$, pattern retrieval probability $P(D|h)$ is defined as follows:

$$\begin{aligned} P(D \mid h) &= \prod_{n=1}^{N} P(\mathbf{x}^{(n)} \mid h) \\ &= \prod_{n=1}^{N} \left\{ \frac{1}{Z(h)} \exp\left\{-\beta\varepsilon(\mathbf{x}^{(n)})\right\} \right\} \\ &= \prod_{n=1}^{N} \left\{ \frac{1}{Z(h)} \exp\left\{ \beta \sum_{i=1}^{|E|} w_i f_i(\mathbf{x}^{(n)}) \right\} \right\} \end{aligned} \tag{2}$$

where h is a given hypernetwork and Z(h) is a partition function which is defined as follows:

$$Z(h) = \sum_{\mathbf{x}^{(n)} \in D} \exp\left\{ \beta \sum_{k=1}^{|E^*|} w_k f_k(\mathbf{x}^{(n)}) \right\} \tag{3}$$

where E^* is set of every possible combination of vertices from a data set.

For the classification problems, given a date set $D = \{\mathbf{x}^{(n)}, y^{(n)}\}_{n=1}^{N}$, $P(D|h)$ can be defined as follows:

$$\begin{aligned} P(D \mid h) &= \prod_{n=1}^{N} P(\mathbf{x}^{(n)}, y^{(n)} \mid h) \\ &= \prod_{n=1}^{N} P(y^{(n)} \mid \mathbf{x}^{(n)}, h) P(\mathbf{x}^{(n)} \mid h) \end{aligned} \tag{4}$$

Here, $y^{(n)}$ is class label for the n-th pattern. Because $P(y^{(n)} \mid \mathbf{x}^{(n)}, h)$ is the probability of correct label prediction for a given pattern and hypernetwork, the probability is defined for all data instances as follow:

$$P(y \mid \mathbf{x}, h) = \frac{1}{N}\sum_{n=1}^{N}\left\{1-\delta(\hat{y}^{(n)}, y^{(n)})\right\} \quad (5)$$

where $\hat{y}^{(n)}$ is the predicted label for the given pattern by the model h such that $\delta(\hat{y}^{(n)}, y^{(n)}) = (\hat{y}^{(n)} - y^{(n)})^2$.

III. DNA Computer Protocol for Learning Hypernetworks

A DNA hypernetwork is represented as a collection of DNA hyperedges in a micro tube. Each vertex of the hyperedge is encoded by DNA oligonucleotide sequences. The DNA hyperedges are regarded as a chromosome or a solution in terms of an evolutionary algorithm. The fitness function is realized in the selection steps in the experimental learning algorithm. We can choose the hyperedge which is closest to the training solution in the selection step. The structure of the DNA hyperedge is shown in Figure 2. The direction of DNA strands from 5' to 3' is representing by the arrowhead. The DNA hyperedge is a single-stranded DNA consists of attributes (vertexes) and tag sequences. The tags are used to identify training hyperedges and test hyperedges in training process. A brief learning algorithm of in vitro DNA hypernetwork is as follows:

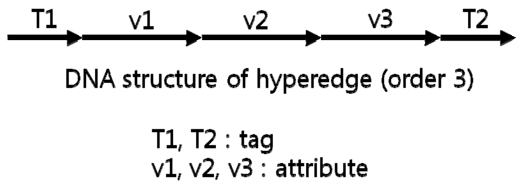


Figure 2. DNA structure of hyperedge. T1 and T2 are tag sequences that used to bind training or test hyperedges by complementary hybridization. Each attribute is represented by a unique DNA sequence.

* Learning algorithm of *In vitro* DNA hypernetwork

1) Initial step

Generate random DNA library (initial library) H of hyperedges h_i, i.e. $H = \{h_i | i = 1, \dots, M\}$. The attribute region of the hyperedges in the initial library will be coded by random DNA sequences (vertex regions). They have the same T1 and T2 sequences.

2) Training step

Given a training data set $D = \{(x_d | y_d) | d = 1, \dots, N\}$.

For d = 1 to N

{Get a training sample and generate a collection of hyperedges $H_d = \{h_j | j = 1, \dots, m\}$

2.1) Hybridization (library + training hyperedges)

Hybridize DNA hyperedges in hypernetwork library H (at first, the library of random hyperedges) with training hyperedges H_d.

2.2) Selection of the hybridized hyperedge duplexes

Arrange the hybridized DNA hyperedges by size using the electrophoresis. Select the duplex of perfect or partially mismatched DNA hyperedges (double-stranded DNA) by gel electrophoresis into H_s. The selection criterion is a fitness function in this algorithm.

2.3) PCR step

Amplify the selected hyperedges H_s into H_a using PCR machine. In this step, tag or complementary tag sequences (T1, T2^c) are used as primer

2.4) Separation of the amplified hyperedges

Separate only the amplified hyperedges H_b which have the same direction as the hyperedges in initial library by bead filtering.

2.5) Update the library

Update the library H by adding the separated hyperedges H_b to produce the new library H.

2.6) Normalization the library

The scale of the hypernetwork library H will be normalized by serial dilution technique.

}

Molecular learning starts with an initial library of random hyperedges. All the single-stranded DNA hyperedges have the same structure with tag sequences at the ends, and attribute sequences in the middle in the same order. In the training step, a certain amount of DNA hyperedges are moved to a new micro tube. Training hyperedges need to be extracted from training data for hybridization with initial library. If a DNA code of attribute 'v1' in the library is AATGC, its complementary DNA code is GCATT (both sequences are 5' to 3'), and it is expressed attribute 'v1^c' in a training hyperedge. The hybridization scheme is shown in figure 3.

A large number of hyperedges will be matched simultaneously in the tube. This operation is a process of information retrieval by parallel matching, but it is not accomplished by looking up an address. These hybridization reactions are fundamentally content-addressable. While figure 3 shows only a type of perfect match, there are three types of partial mismatches in hybridization step. When we select the matched hyperedge duplex in the separation step, we can divide hyperedges of perfect matched from partially mismatched ones because partially mismatched hyperedges move slower than perfect matched ones in gel electrophoresis. If the partially mismatched are not allowed in selection step, the hypernetwork is made up of only perfectly matched hyperedges.

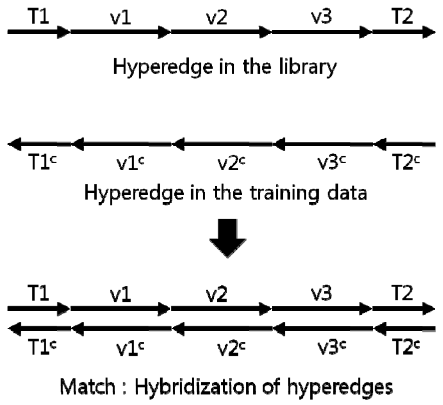

T1
v1
v2
v3
T2
Hyperedge in the library
T1c
v1c
v2c
v3c
T2c
Hyperedge in the training data
Match : Hybridization of hyperedges

Figure 3. Hybridization in training step. If a hyperedge in the library has the same attributes as a hyperedge in the training data, they will perfectly match each other. (v1^c has a complementary sequence to v1)

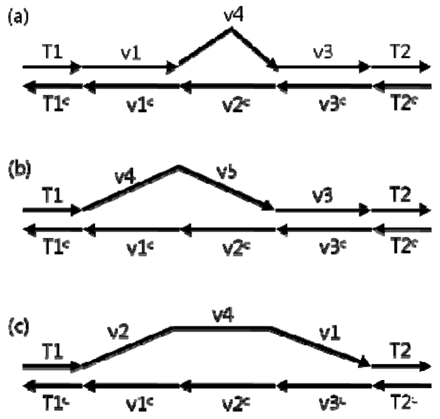


Figure 4. Different types of mismatches. A hybridized hyperedge with more mismatches moves more slowly in the gel separation step. (a) Just one mismatched attribute. (b) Two mismatched attributes. (c) Fully mismatched.

However, if we allow partially mismatched hyperedges, the hypernetwork library can be updated with new hyperedges which may not exist in the training data. The selection measure is a fitness function of this algorithm. We expect that a mixture of hyperedges with perfect matches and partial mismatches with just one attribute would be a more optimal solution in the hypernetwork population. We can also think of the selection operation as a kind of mutation process. They can change the structure of hypernetworks and can affect the evolution of hypernetwork library. Selected hyperedges are amplified by PCR machine in the amplification step. Specially designed primers, conjugated with beads for filtering, are inserted at the PCR step. After amplification, the copied hyperedges are filtered by bead separation technique. We want to select in only one direction of single-stranded DNA strands, which have the same type of the hyperedges in the library, by this purification step. The update process carried out using the purified hyperedges shown in figure 5.

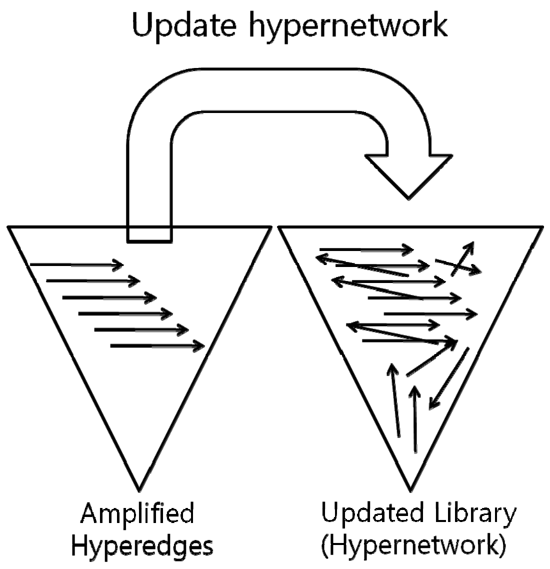


Figure 5. Hypernetwork update process. The amplified hyperedges are inserted to the hypernetwork library.

After the update process, we can perform the training repeatedly. The experimental techniques of matching, selection, and amplification which are used in the entire process of the algorithm are kinds of evolutionary operators. In the course of the evolutionary learning process, the hypernetwork gradually approaches the target distribution.

IV. SIMULATION ALGORITHM OF LEARNING DNA HYPERNETWORKS

We made the DNA hypernetwork learning algorithm by applying the Metropolis algorithm [21] for our simulation study:

1) Set initial library = random sequences

2) Randomly select K inputs from the training data with an accept probability P_{acc}

3) Alternate input with probability P_{al}

$$P_{al} = \min\left\{1, \exp\left(-\beta \times \Delta h \cdot [\Delta A]\right)\right\}$$
$$\beta = \frac{1}{kT}$$

k: Boltzmann constant
T: Absolute temperature
A = extent of alternation
h = distance measure

4) **If** input $\in$ library

Increase weight of corresponding element in the library

else

{library} = {input} $\cup$ {library}

5) Sort element of the library according to their weights.

6) **While** population of the library exceeds $POP_{\max}$

Dispose elements with least weight

7) Iterate step 2 through 6

The step 1) is same process as the initial step in molecular learning protocol. Similarly, we transferred the molecular training steps as step 2) to 7). In step 3), alternation is important step in simulating partial mismatches of DNA hyperedges. For example of cognitive learning, consider the input string (hyperedge) = 'I love you', and the alternate strings = {'we love you', 'I love him', 'you love me'…}. For in vitro experiments, there is every possible sequence of hyperedges and every possible alternate should be generated. But in our simulation, the generation is restricted because of the lack of computer memory. If we assume there are 40000 words in human language, we can alternate a three words string such as 'I love you' allowing an alternation of only one word. Then the result would be 64×10^{12} kinds of hyperedges which are almost impossible to simulate on digital computer. This is why we use "don't care" symbol *. If "don't care" symbol appears, there can be any possible words. Strings with don't care symbol can be regarded as follows,

$$\text{'I } * \text{ you'} = \{\text{'I } * \text{ you'} \mid * \in \text{ Every words in language}\}.$$

P_{al} can be regarded as a probability of a partial mismatch or a perfect match. And also can be viewed as a fitness function in a genetic algorithm. We cannot simulate the real procedure of chemical reactions *in silico*, which is the different case in the molecular algorithm. On the contrary to the initial step of learning algorithms of *in vitro* DNA hypernetwork, we do not create a random initial library. This is because the memory in the computer is relatively small compared to that of DNA computer. For this reason we merely assume there is random library at the initial step of computer simulation. We used Hamming distance h to calculate the distances from matched hyperedges in training process. The distance between perfectly matched hyperedges is zero and between partially mismatched hyperedges is just one attribute in order 3 hyperedges. The temperature T was regarded as a constant to reduce simulation time. We also restrict library size to $POP_{\max}$ for the same reason. Limiting $POP_{\max}$ and accepting input with probability P_{acc} corresponds to the dilution process in the DNA computing, since dilution eliminates insignificant elements in the library.

V. Simulation Results ans Discussion

For the simulation study, we used two corpora American TV drama "Friends" and "Prison break" [22]. Each corpus is composed of whole season of the drama scripts. We preprocessed the sentences of the scripts for training and test data in the simulation study. Each hyperedge (phrase) is extracted from consecutive positions in the sentence.

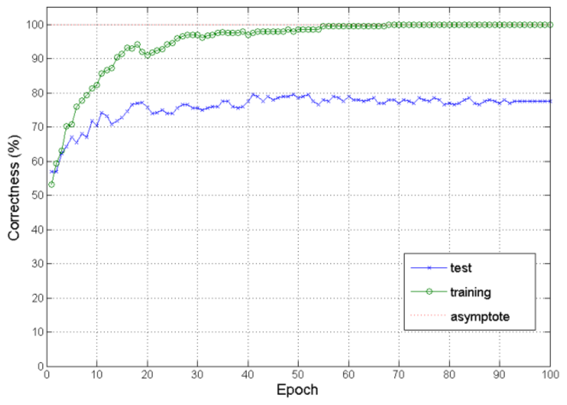


Figure 6. The learning curve of the simulation study of DNA hypernetworks model. The result shows the expected performance of molecular learning. The blue line indicates the corpus style estimation result of test set. The graph shows a steady upward learning curve. On the contrary the training graph shows a sharp increase.

The process of extracting hyperedges was performed by moving a 3-order window to the right in a word-by-word manner. For example, given the sentence 'A new cabinet has been formed', four hyperedges of order-three can be extracted, generating 'A new cabinet', 'new cabinet has', 'cabinet has been', and 'has been formed'. In the simulation study, we use only 50,000 hyperedges in each corpus to adjust the balance. In fact, the cognitive learning process is a sort of an unsupervised learning (we don't use the class information in training step). Therefore, we need criteria for judging the score of performance. The DNA hypernetworks trained from different class data can be used as corpus style estimators. The evolving DNA hypernetworks takes after the style of the training drama corpus. Using the hypernetworks, we can carry out a series of test to estimate the drama style of some unknown drama sentences. The data was divided into training and test sets. The 80% of the total was included in the training set. Two hypernetwork was trained with the molecular learning algorithm, one with the Friends and one with Prison Break. Then, the learned hyperedges were compared with those in the test set. Accuracy refers to the number of test hyperedges that were correctly classified as either Friends or Prison Break. The simulation was repeated 5 times and the accuracies averaged. The simulations were performed to show their molecular learning properties and estimation performances.

Figure 6 shows the change of accuracies over generations in the corpus style estimation tasks. The training accuracy increased and saturated early and the testing accuracy was comparatively slower. In the training, we showed the test hyperedges (for the estimation) obtained from training data. But in the testing case, we showed really test hyperedges which were not used in training data. We did not compare our algorithm with other machine learning algorithms. The reason is because the others have a very little chance to realize the molecular learning properties. The highly ranked hyperedges after the simulation are shown in figure 9. At a glance, it is rather hard to notice the difference between the hyperedges of "Friends" and "Prison Break". This is due to two reasons. First, the most frequent patterns are likely to be uniformly distributed in human language. And second, this model maintains about 20000 or more hyperedges in a population that is much smaller than in

DNA computing. Every one of them participates to model a language corpus. We did observe interesting patterns. Hyperedges such as ‘oh my god’ appeared to be dominant in “Friends” while was not in “Prison Break”. Seventh hyperedge from the top in “Prison Break” was ‘out of here’ and it seems to represent the TV show well. Figure 7 shows that slight variations of partially mismatched strings were capable of modeling delicate features of human language corpora.

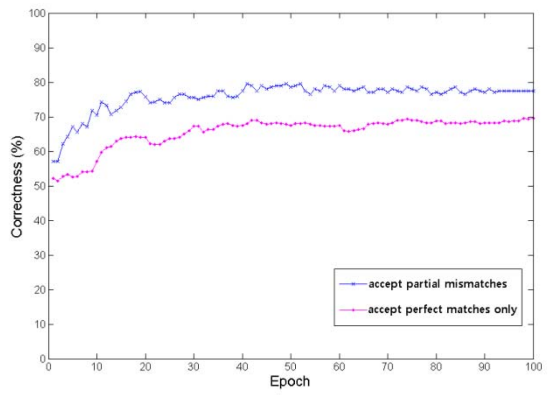


Figure 7. The learning curve for different selection processes. When we allow the partial mismatches, the estimation performance is slightly changed.

Also it shows a comparison between accepting partial mismatches or allowing perfect matches in the selection step. Allowing partial mismatch shows better results because accepting partially mismatched hyperedges will result in a much more diverse population through mutation. It shows that allowing partial mismatches in the selection step is an essential process for molecular machine learning.

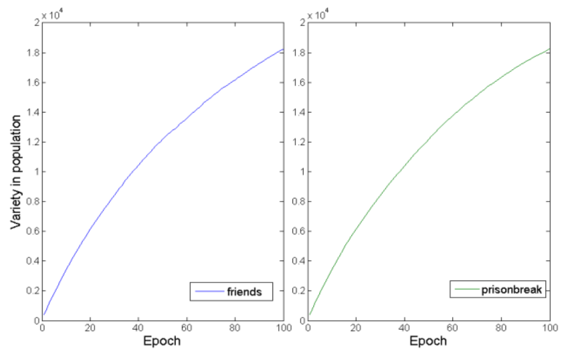


Figure 8.The diversity of the population of each hypernetwork.

Figure 8 shows population diversity of the hypernetworks versus epoch without the limits on maximum population or accept probability. The tendency in this graph is because we count the hyperedges as one with “don’t care” in them. This is in fact not the same in the case of molecular learning. Because hyperedges with “don’t care” represent all the partially mismatched hyperedges, the number of which is the size of word dictionary. So, the size of diversity will grow exponentially in case of DNA computing.

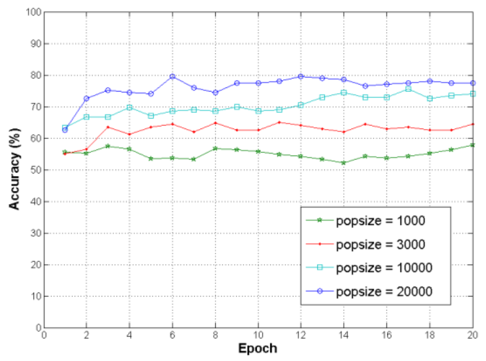
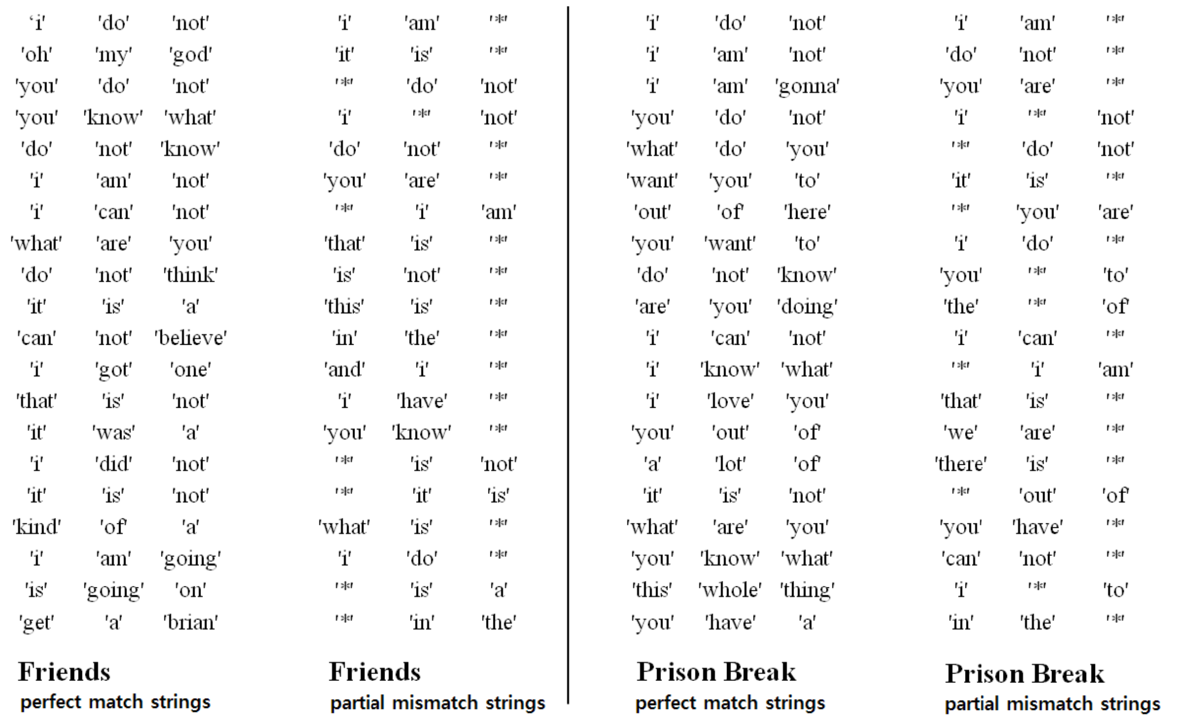


Figure 10. The accuracy curve for different size of the population.

The accuracy curves for different sizes of the populations are shown in figure 10. The training data were randomly drawn from 10000 hyperedges and 1000 test data from another corpus



‘i'	'do'	'not'
'oh'	'my'	'god'
'you'	'do'	'not'
'you'	'know'	'what'
'do'	'not'	'know'
'i'	'am'	'not'
'i'	'can'	'not'
'what'	'are'	'you'
'do'	'not'	'think'
'it'	'is'	'a'
'can'	'not'	'believe'
'i'	'got'	'one'
'that'	'is'	'not'
'it'	'was'	'a'
'i'	'did'	'not'
'it'	'is'	'not'
'kind'	'of'	'a'
'i'	'am'	'going'
'is'	'going'	'on'
'get'	'a'	'brian'

Friends
perfect match strings

'i'	'am'	'*'
'it'	'is'	'*'
'*'	'do'	'not'
'i'	'*'	'not'
'do'	'not'	'*'
'you'	'are'	'*'
'*'	'i'	'am'
'that'	'is'	'*'
'is'	'not'	'*'
'this'	'is'	'*'
'in'	'the'	'*'
'and'	'i'	'*'
'i'	'have'	'*'
'you'	'know'	'*'
'*'	'is'	'not'
'*'	'it'	'is'
'what'	'is'	'*'
'i'	'do'	'*'
'*'	'is'	'a'
'*'	'in'	'the'

Friends
partial mismatch strings

'i'	'do'	'not'
'i'	'am'	'not'
'i'	'am'	'gonna'
'you'	'do'	'not'
'what'	'do'	'you'
'want'	'you'	'to'
'out'	'of'	'here'
'you'	'want'	'to'
'do'	'not'	'know'
'are'	'you'	'doing'
'i'	'can'	'not'
'i'	'know'	'what'
'i'	'love'	'you'
'you'	'out'	'of'
'a'	'lot'	'of'
'it'	'is'	'not'
'what'	'are'	'you'
'you'	'know'	'what'
'this'	'whole'	'thing'
'you'	'have'	'a'

Prison Break
perfect match strings

'i'	'am'	'*'
'do'	'not'	'*'
'you'	'are'	'*'
'i'	'*'	'not'
'*'	'do'	'not'
'it'	'is'	'*'
'*'	'you'	'are'
'i'	'do'	'*'
'you'	'*'	'to'
'the'	'*'	'of'
'i'	'can'	'*'
'*'	'i'	'am'
'that'	'is'	'*'
'we'	'are'	'*'
'there'	'is'	'*'
'*'	'out'	'of'
'you'	'have'	'*'
'can'	'not'	'*'
'i'	'*'	'to'
'in'	'the'	'*'

Prison Break
partial mismatch strings

of 10000 hyperedges. All the other parameters except the population size are fixed. We can observe that as population grows, the accuracy increases. This is because diversity comes with population size. However it is impossible to simulate on a computer the size of the population in molecular learning systems. We can estimate that it will show better results on molecular DNA hypernetworks even with tremendous size of data. Early convergence on this graph is due to the high accept probability. With high accept probability the model learns faster with the risk of trapping in local optima. It will cause a fluctuation effect on the learning curve.

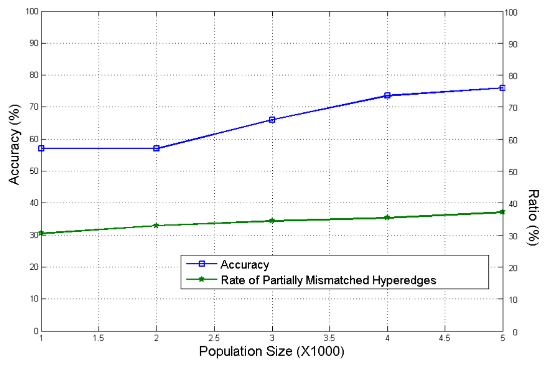


Figure 11. The accuracy and rate of partially mismatches hyperedges curves for different population size.

Figure 11 shows the association between accuracy and the rate of partially mismatched hyperedges for different sizes of the population. It shows the rate of partially mismatched hyperedges is increasing when growing the population size. We can say that the accuracy affected by the population size and the rate of partially mismatched hyperedges in the hypernetwork library.

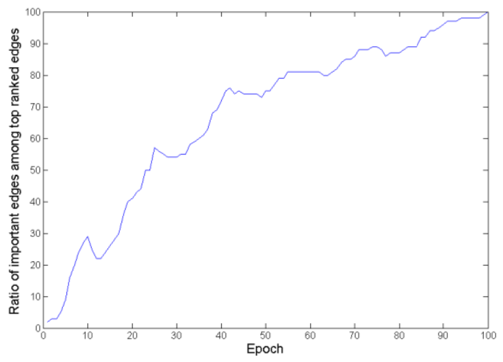


Figure 12. The change of the number of important hyperedges.

Figure 12 shows the emergence of important group of the hyperedges. Highly ranked hyperedges in the final epoch were regarded as important because the accuracy grows within the iteration. However the graph does not seem to be stabilized. This is because the language data are so diverse that similar patterns are hard to grasp and thus makes the task of estimation harder. If we perform it using DNA molecules, the diversity is represented well in the hypernetwork library (as cognitive memory) and we can estimate the target naturally.

VI. CONCLUSIONS

We designed an *in vitro* evolutionary molecular learning method and simulated the algorithm by corpus style estimation tasks. The algorithm was designed to make it possible learning in molecular level environment. We could see the molecular learning properties in the simulation results. The DNA hypernetworks showed several properties of general machine learning algorithms. They can be used to estimate the style of drama corpus from unknown test sentences. The hypernetworks are learned by the evolutionary operators such as selection, mutation and amplification. These operations drove the initial DNA population into the distribution of a training data. The population size was one of the important factors in the DNA hypernetworks learning. A sufficiently large size of the population was needed to derive good results in the estimation process. Also the mutation process was a crucial factor for the hypernetworks to make their discernments as the estimators. We also found that some patterns of partially mismatched hyperedges made the estimating performance better. The hypernetworks is a kind of probabilistic graphical model. The probabilistic graphical model is the machine learning algorithm designed for simulating a cognitive function of human. This function includes inference, classification, and clustering. We hope our novel approach gives an evolutionary contribution to the simulation of cognitive function of human in the formulation of machine learning.

To realize a molecular machine learner, we have to perform a wet-lab experiment which implements the proposed algorithm. However, the state of the art techniques in molecular computing have not yet arrived at a mature stage. There are many problems for us to solve. These obstacles include generating organized random DNA sequences, synthesizing a designed DNA sequence in large amounts, and cross hybridization problem. Even though the situation is indeed tough, we hope we can improve the status of DNA computing with the experimental protocols and simulation results in this paper.

REFERENCES

[1] T. Sienko, A. Adamatzky, N. G. Rambidi, and M. Conrad, Molecular computing, *Genetic Programming and Evolvable Machines*, vol. 6, pp. 349–351, 2005.

[2] J. Elbaz, O. Lioubashevski, F. Wang, F. Remacle, R. D. Levine, and I. Willner, DNA computing circuits using libraries of DNAzyme subunits, *Nature Nanotechnology*, vol. 5, pp. 417–422, 2010.

[3] M. Stojanovic, Two Molecular Information Processing Systems Based on Catalytic Nucleic Acids, *Natural Computing*, pp. 55–63, 2010.

[4] T. Ran, S. Kaplan, and E. Shapiro, Molecular implementation of simple logic programs, *Nature Nanotechnology*, vol. 4, pp. 642–648, 2010.

[5] R. D. Barish, R. Schulman, P. W. Rothemund, and E. Winfree, An information-bearing seed for nucleating algorithmic self-assembly, *Proceedings of the National Academy of Sciences*, vol. 106, p. 6054, 2009.

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

[7] E. B. Baum, Building an associative memory vastly larger than the brain, *Science*, vol. 268, pp. 583, 1995.

[8] J. Chen, R. Deaton, and Y. Z. Wang, A DNA-based memory with in vitro learning and associative recall, *Natural Computing*, vol. 4, pp. 83–101, 1995.

[9] C. Y. Lee and M. C. Paull, A content addressable distributed logic memory with applications to information retrieval, *Proceedings of the IEEE*, vol. 51, pp. 924–932, 1963.

[10] C. Y. Lee and R. Y. Yang, High-throughput data compressor designs using content addressable memory, *IEE Proceedings Circuits, Devices and Systems*, pp. 69–73, 1995.

[11] K. Pagiamtzis and A. Sheikholeslami, Content-addressable memory (CAM) circuits and architectures: A tutorial and survey, *IEEE Journal of Solid-State Circuits*, vol. 41, pp. 712–727, 2006.

[12] D.E. Rumelhart, The architecture of mind: A connectionist approach, *M*, vol. 1, pp. 133–159, 1989.

[13] G.E. Hinton, J.L. McClelland, and D.E. Rumelhart, Distributed representations, *Carnegie Mellon University Computer Science Dept.*, 1984.

[14] T. Kohonen, Content-Addressable Memories, Springer-Verlag, 1987.

[15] B.T. Zhang, Hypernetworks: A molecular evolutionary architecture for cognitive learning and memory, *Computational Intelligence Magazine, IEEE*, vol. 3, pp. 49–63, 2008.

[16] J.K. Kim and B.T. Zhang, Evolving hypernetworks for pattern classification, *IEEE Congress on Evolutionary Computation (CEC 2007)*, pp. 313-319, 2007.

[17] J. Bootkrajang, S. Kim, and B.T. Zhang, Evolutionary hypernetwork classifiers for protein-proteininteraction sentence filtering, *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pp. 185–192, 2009.

[18] B. T. Zhang and H. Y. Jang, Molecular programming: evolving genetic programs in a test tube, *The Genetic and Evolutionary Computation Conference (GECCO 2005)*, pp. 1761–1768, 2005.

[19] S. Kim, M.O. Heo, and B.T. Zhang, Text classifiers evolved on a simulated DNA computer, *IEEE Congress on Evolutionary Computation (CEC 2006)*, pp. 2646–2652, 2006.

[20] J.W. Ha, B.H. Kim, B. Lee, and B.T. Zhang, “Layered hypernetwork models for cross-modal associative text and image keyword generation in multimodal information retrieval,” *PRICAI 2010: Trends in Artificial Intelligence*, 2010, pp. 76–87.

[21] J. S. Kim, J. W. Lee, Y. K. Noh, J. Y. Park, D. Y. Lee, K. Yang, Y. G. Chai, J.C. Kim, and B.T. Zhang, An evolutionary Monte Carlo algorithm for predicting DNA hybridization, *Biosystems*, vol. 91, pp. 69–75, 2008.

[22] B. T. Zhang and C. H. Park, Self-Assembling Hypernetworks for Cognitive Learning of Linguistic Memory, *Proceedings of International Conference on Computer, Electrical, and Systems Science, and Engineering (CESSE2008)(WASET)*, Vol. 27, pp. 134-138, 2008.