

Molecular Evolutionary Learning of DNA Hypernetworks for Hand-written Digit Classification

Christina Baek¹, Je-Hwan Ryu¹, Ji-Hoon Lee², Byoung-Tak Zhang³

¹Interdisciplinary Program in Neuroscience, Seoul National University, Republic of Korea,

²Interdisciplinary Program in Bioinformatics, Seoul National University, Republic of Korea,

³Department of Computer Science & Engineering, Seoul National University, Republic of Korea

{dsbaek, jhryu, jhlee, btzhang}@bi.snu.ac.kr

An intersection between biology and computation is explored by classifying handwritten digits using DNA computing. To do this, we propose employing the Molecular Evolutionary Hypernetwork [1] *in vitro* using specifically encoded DNA as pixels of images. This machine learning algorithm consists of various stages; matching of training or test data with the Hypernetwork (starting with a random single-stranded library of 3-order hyperedges), selection and amplification of similar sequences, isolation, classification, feedback and update of the Hypernetwork used in the next round of training. A key limiting factor of this design is the issue of scalability, as a very large dataset would need to be provided for demonstrating the learning process [2]. To address this, and for the verification of each experimental step, we design, create and test a random single-stranded DNA library using various techniques, which provides a base of larger scale experiments demonstrating molecular learning.

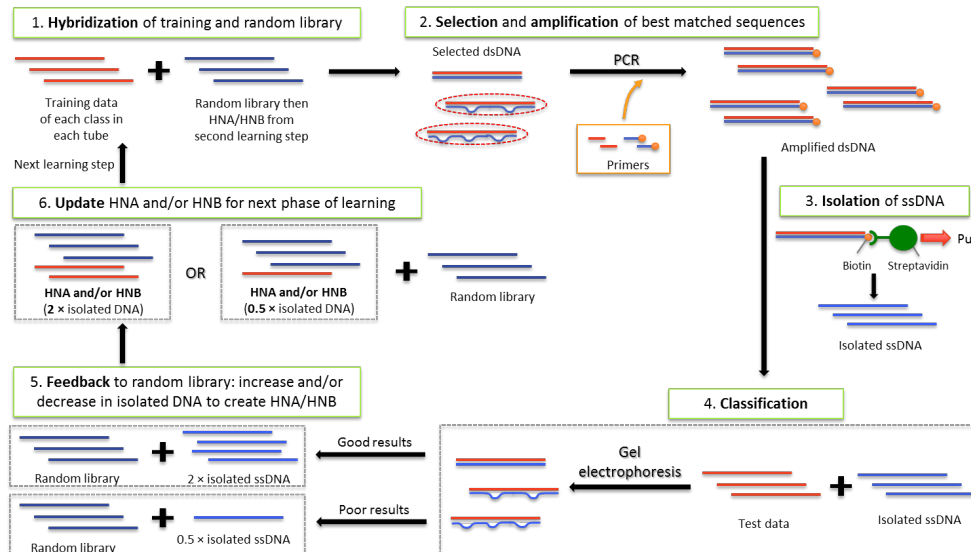


Figure 4. Overall experimental process of the evolutionary molecular Hypernetwork.

References

1. Zhang, B.-T. Hypernetworks: A molecular evolutionary architecture for cognitive learning and memory. *IEEE Comput. Intell. Mag.* 3, 49-63 (2008)
2. Lee, J.-H., Lee, B., Kim, J., Deaton, R. & Zhang, B.-T. A molecular evolutionary algorithm for learning hypernetworks on simulated DNA computers. *IEEE Cong. Evol. Comp. (CEC)*, 2735–2742 (2011)

Acknowledgement

This work was supported by Samsung Foundation for Future Research Promotion (SRFC-1401-12)