Molecular associative memory

An associative memory framework with exponential storage capacity for DNA computing

algolma
laboratoire d'algorithmique
laboratoire de mathématiques algorithmiques

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

1. Introduction

Associative memory problem : Find the closest stored vector (in Hamming distance) to a given query vector.

Neural implementation

- Using neural networks, connection weights are adjusted in order to perform association.
- Recall procedure is iterative and relies on simple neural operations.
- Design criteria: maximizing the number of stored patterns C while having some noise tolerance.

Molecular implementation

- Synthesize C DNA strands as stored vectors.
- Recall procedure is usually done in one shot via chemical reactions and relies on highly parallelism of DNA computing.
- Design criteria: finding proper DNA sequences to minimize probability of error during the recall phase.

2. The problem

Current molecular associative memories are either

- **low in storage capacity**, if implemented using molecular realizations of neural networks [3].

Or

- **very complex to implement**, if all the stored sequences have to be synthesized [7], [1].

3. The proposed solution

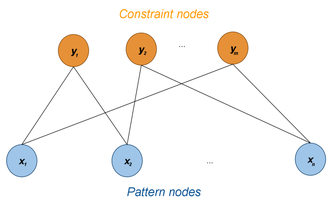
We introduce an associative memory framework with exponential storage capacity based on transcriptional networks of *DNA switches* proposed by [3].

Advantages over current methods

- Exponential storage capacities with current neural network-based approaches can not be achieved.
- For other methods, although having exponential storage capacities is possible, it is very complex as it requires synthesizing an extraordinarily large number of DNA strands.

4. Model and method

- We utilize a bipartite network of *DNA switches* with n pattern nodes and m constraint nodes.
- The connectivity of the network is determined by the adjacency matrix H.



- The state of each pattern node j, denoted by x_j, can either be 1 (activation) or -1 (suppression).
- The state of each constraint node i (denoted by y_i) can be 1 (activation), -1 (suppression) or 0 (non-transcribed).
- Each constraint node y_i has a decision threshold b_i.
- Given the vector of decision thresholds b and pattern nodes states x, we fix H such that $Hx = b$.

Hence, instead of *memorizing* all possible random sequences of length n, we store only those that satisfy m constraints.

5. The association process

The proposed framework finds the closest stored pattern to the probe $\hat{x}$ via forward and backward iterations.

Forward iteration

- Constraint nodes decide their state based on simple *neural* operations:

$$y_i = \begin{cases} 1, & h_i < b_i \\ 0, & h_i = b_i \\ -1, & \text{otherwise} \end{cases}.$$

where $h_i = \sum_{j=1}^{n} H_{ij} x_j$,

Backward iteration

- Each pattern node j computes the quantity

$$g_j = \frac{\sum_{i=1}^{m} H_{ij} y_i}{d_p}.$$

The sign of g_j is an indication of the sign of the noise that affects x_j, and $|g_j|$ indicates the confidence level in the decision.

- The state of pattern DNA node j is updated using either of the following two strategies:
  1. *Winner-take-all* strategy: only the node with the maximum $|g_j|$ is updated.
  2. *Bit-flipping* strategy: all pattern nodes are updated based on the sign of g_j.

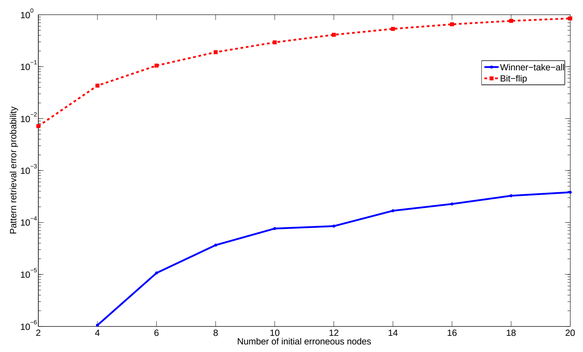
6. Results

Theoretical results

- The proposed framework is guaranteed to correct two erroneous nodes [4].
- For proper choice of row degrees in the constraint matrix, it also admits an exponential storage capacity in terms of n.

Numerical results

- The following graph illustrates the pattern retrieval error probability against the number of initial erroneous nodes.



7. Some remarks

Full details about the approach can be found in [4].

The proposed method have other possible applications as well:

- Designing artificial transcriptional networks to govern the activity of cells, for instance in combating certain diseases.
- Iterative error correction in DNA computing instead of pre-designed *error-avoiding* DNA sequences.

8. Previous works

Neural Associative Memory

- Extensive studies in past decades [2], [5].
- Storage capacity has been shown to be at best equal to n, the number of neurons, *when required to memorize purely random patterns*.
- Recently, some works have been done to improve the storage capacity by memorizing *structured patterns* (see [4] and references therein).

Molecular Associative Memory

- In contrast to neural associative memory, most approaches are already concerned with memorizing *structured patterns* to minimize recall probability of error.
- These approaches synthesize all the stored patterns and store them in a vessel [7], [1].
- Coding theory can help in designing DNA strands that admit low probability of error in the recall process [6].
- Some approaches that implement neural networks using DNA strands can be used as a means of implementing associative memory as well [3].
- However, the *storage capacity* of molecular associative memory is not well-studied yet.

References

[1] J. Chen, R. Deaton, Y. Z. Wang, *A DNA-based memory with in vitro learning and associative recall*, Lect. Notes in Comp. Sci., Volume 2943, 2004, pp. 145-156.

[2] J. J. Hopfield, *Neural networks and physical systems with emergent collective computational abilities*, Proc. Natl. Acad. Sci., Vol. 79, 1982, pp. 2554-2558.

[3] J. Kim, J. J. Hopfield, E. Winfree, *Neural network computation by in vitro transcriptional circuits*, Adv. Neur. Inf. Proc. Sys. (NIPS), Vol. 17, 2004, pp. 681-688.

[4] K. R. Kumar, A. H. Salavati, A. Shokrollahi, *Exponential pattern retrieval capacity with non-binary associative memory*, submitted to Information Theory Workshop 2011.

[5] R. McEliece, E. Posner, E. Rodemich, S. Venkatesh, *The capacity of the Hopfield associative memory*, IEEE Trans. Inf. Theory, Jul. 1987.

[6] O. Milenkovic, N. Kashyap, "On the Design of Codes for DNA Computing" Lect. Notes in Comp. Sci., Vol. 3969, 2006, pp. 100-119.

[7] J. H. Reif, T. H. LaBean, *Computationally inspired biotechnologies: improved DNA synthesis and associative search using error-correcting codes and vector-quantization*, Lect. Notes in Comp. Sci., Vol. 2054, 2001, pp. 145-172.

Acknowledgment

The authors would like to thank Prof. Wulfram Gerstner for helpful comments and discussions. This work was supported by Grant 228021-ECCSciEng of the European Research Council.