Towards Programmable Self-Assembly in Biological and Artificial Systems

by A. M. Bouchard and G. C. Osbourn

Motivation—Dynamic self-assembly is a ubiquitous process in non-equilibrium physical and biological systems. Our research program pursues the theme that in such systems, dynamic self-assembly and computation are intimately related: that dynamic self-assembly can be used to perform computation, and that if the computational language can be understood, it can be used to program self-assembly. The first goal is to show that computation emerges in biological systems via self-assembly. This leads to two technological goals: (1) a novel self-assembling software technology modeled after biological computation, and (2) creating artificial systems (e.g., materials or circuits) that are programmed to dynamically self-assemble into desired hierarchical structures with the robustness of biological systems.

Accomplishment—We have built a common agent-based simulation infrastructure for use both in the self-assembling software and the stochastic simulations of self-assembling physical systems. Each agent embodies the properties of proteins that are required for self-assembly and computation-selective binding, which triggers actuation and the exposing or hiding of other binding sites.

We have implemented a variety of software self-assembly processes, including generating data structures, multiple kinds of executable code structures, dynamic execution pathways, hierarchies of software modules, movement of modules within the hierarchy and triggers that execute or inhibit certain code structures. We have also implemented a number of novel software constructs that are enabled by the self-assembly approach, such as the ability to override or monitor existing code externally, without modifying the original source code, recompiling, or terminating its execution.

As a first step toward developing programmed self-assembly in physical systems, we have demonstrated that selected idealized proteins can be designed to carry out computation. We find that stochastic versions of any deterministic Turing machine can in principle be obtained using dynamic self-assembly of proteins that exhibit commonly available properties. We also find that partial equilibration of the far-from-equilibrium stochastic protein networks intrinsically leads to increasing computational errors with length of computation, so that an ensemble of such computing networks diverges in configuration with time to different internal states and different computational results. This is a direct consequence of the stochastic nature of the protein networks and the second law of thermodynamics. (See figures.) This implies that if natural systems perform computations with low error rates, they must employ error-correction mechanisms as part of the algorithm.

Significance—These are our first results in the new Collective Hierarchical Systems technical thrust for Center 1100. This initial success suggests that programming complex self-assembly processes is, indeed, feasible. The next phase of this work will include: demonstrating error-correction schemes for the computational protein networks; developing an English-like language to autogenerate the self-assembling software; and developing simulations to guide experimentalists in harnessing cellular machinery (motor proteins and microtubules) to assemble novel materials.

Sponsors for various phases of this work include: DOE Office of Basic Energy Sciences and Laboratory Directed Research & Development
Contact: Ann M. Bouchard; Lasers, Optics, & Remote Sensing; Dept. 1118
Phone: (505) 845-0552, Fax: (505) 844-5459, E-mail: bouchar@sandia.gov

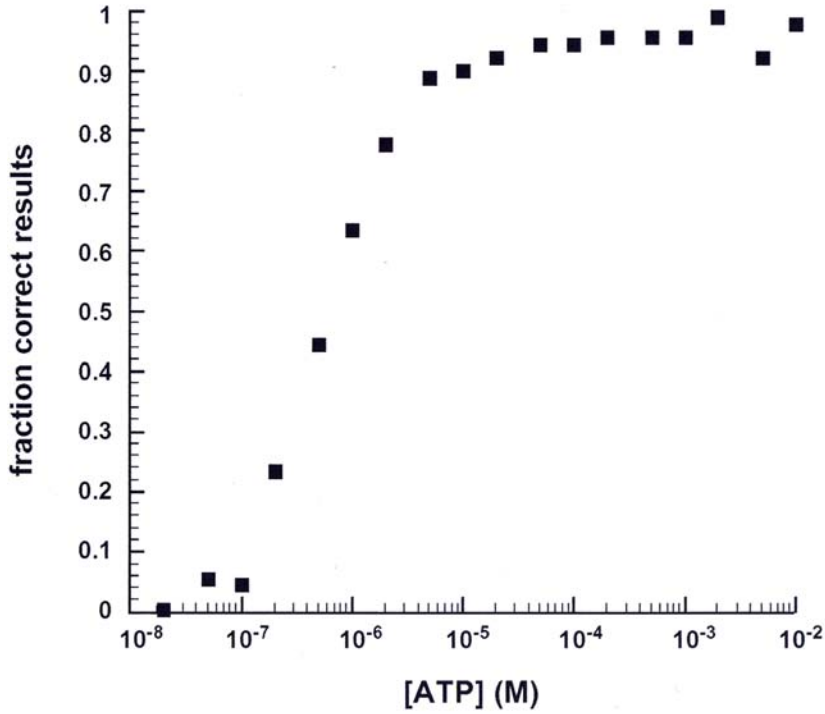

Figure 1. We implemented simulations of computing (a*b)+(c*d)+(e*f) using an ATP-driven dynamic stochastic protein network. The fraction of correct final results as a function of ATP concentration is shown. These results make clear that the dynamic, non-equilibrium behavior of these protein networks is driven by the free energy of the ATP population.

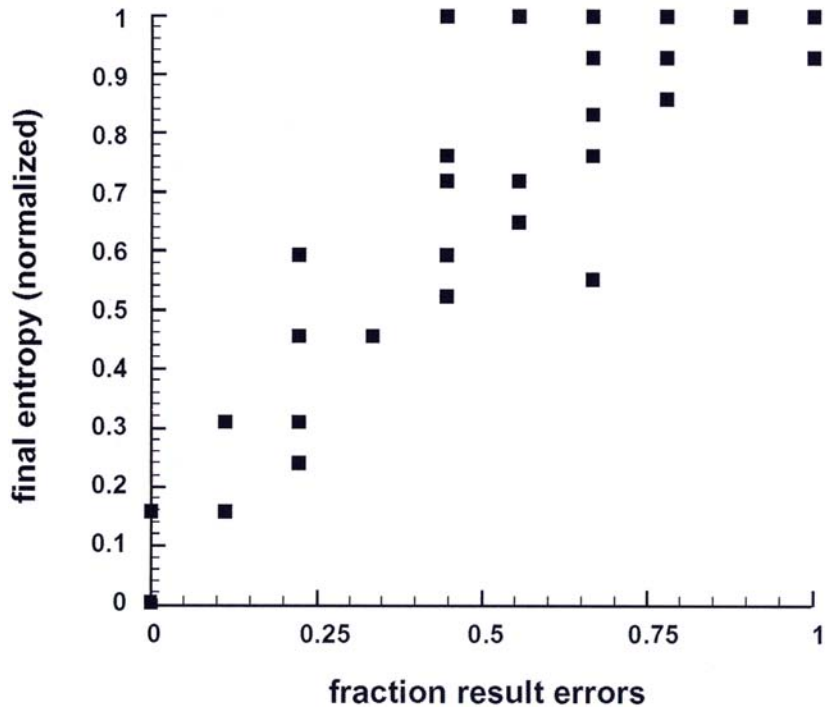

Figure 2. A scatter plot of final entropy as a function of errors for the computation described in Fig. 1. These results show that ending in a more highly ordered state is clearly correlated with high yields of correct computational results, so that maintaining far-from-equilibrium configurations is the desired outcome for these protein networks. The entropy captures all configurational differences, including those that do not disrupt the final register values, and this produces the scatter in the plot.