Article

General design principles for combining functional modules into complete cells: From compatibility to emergence

WenJun Zhang
School of Life Sciences, Sun Yat-sen University, Guangzhou 510275, China; International Academy of Ecology and Environmental Sciences, Hong Kong
E-mail: wjzhang@iaees.org, zhwj@mail.sysu.edu.cn

Received 27 August 2026; *Accepted 9 September 2026*; *Published online 15 September 2026*; *Published 1 June 2027*

Abstract
Bottom-up synthetic biology has achieved substantial progress over the past decade, enabling the reconstitution of gene expression, metabolic cascades, and signal transduction as independent functional modules within giant unilamellar vesicles. However, combining these modules into integrated complete cells continues to face fundamental difficulties. Resource competition, signal crosstalk, chemical incompatibility, and loss of function after combination occur frequently. This study proposes a three-layer design principle framework organized around constraint, coupling, and emergence. Module independence is limited by the thermodynamic constraints of a shared resource pool, and the operation of any module inevitably alters the host free energy landscape and metabolic state. The feasible region of module coupling is defined jointly by the capacity of information transfer channels and noise tolerance. The function of a complete cell is not a simple superposition of module functions but emerges as new system-level dynamical behaviors when inter-module coupling strength crosses a critical threshold. This study formalizes these principles into an operational module compatibility matrix and a coupling strength criterion, and proposes experimental validation strategies. This framework offers a unified theoretical perspective for rational design from modules to complete cells, transforming scattered engineering experience into testable design principles.

Keywords synthetic cells; synthetic biology; modular design; non-equilibrium thermodynamics; self-organization; emergence; general design principles.

Network Biology
ISSN 2220-8879
URL: http://www.iaees.org/publications/journals/nb/online-version.asp
RSS: http://www.iaees.org/publications/journals/nb/rss.xml
E-mail: networkbiology@iaees.org
Editor-in-Chief: WenJun Zhang
Publisher: International Academy of Ecology and Environmental Sciences

1 Introduction

1.1 The Gap from Module Reconstitution to Complete Cells

One central goal of bottom-up synthetic biology is to construct systems with cell-like functions from non-living molecular components (Reverte López, 2025). Over the past decade, researchers have successfully

reconstituted various independent functional modules within lipid vesicles, including cell-free protein synthesis systems, metabolic cascade reactions, light-driven energy supply, and DNA-based gene circuits (Reverte López, 2025). These modules perform well in their respective isolated testing environments (Reverte López, 2025). However, when they are simultaneously encapsulated within the same vesicle, the system frequently undergoes functional collapse. Resource competition between modules, membrane protein interference with transport modules, and conflicting preferences for pH and ionic strength are common failure modes (Rijal et al., 2025).

This predicament is not merely an accumulation of technical details but points to a deeper problem. The composability of functional modules does not hold automatically (Inoue et al., 2014). Placing a module validated in an isolated environment into the context of a complete cell means it enters a coupled system that shares physicochemical resources, thermodynamic constraints, and information channels (Zhang, 2016b, 2018). The behavior of the module is no longer determined unilaterally by its own design parameters but is co-shaped by its interactions with other modules in the system (Zhang, 2016b). Understanding this transition from independent modules to coupled systems requires general design principles that go beyond traditional engineering modular thinking (Zhang, 2013).

1.2 Limitations of Existing Frameworks

The design-build-test-learn cycle and standardized BioBrick approach in synthetic biology inherit the metaphor of replaceable modules and standardized interfaces from electronic engineering (Inoue et al., 2014). This metaphor has achieved considerable success at the gene circuit level, but its limitations become apparent when the design object expands from a single circuit to a complete cell containing metabolism, signaling, transport, and energy supply (Reverte López, 2025). The assumption of standardized interfaces presupposes that no hidden coupling exists between modules, but in physical systems at the cellular scale, all modules are inevitably coupled through the shared solvent environment, membrane interfaces, free energy currencies (ATP, NADH), and ion gradients (Rijal et al., 2025).

Existing research has identified module compatibility as a core bottleneck in the current bottom-up synthetic cell field (Reverte López, 2025). However, most discussions remain at the phenomenological level of enumerating compatibility problems such as resource competition and insufficient orthogonality, lacking a unified theoretical framework that connects these problems (Rijal et al., 2025). Relatedly, research in theoretical biology on self-organization, dissipative structures, information theory, and network modularity provides rich conceptual tools, but these have not been systematically introduced into the module combination problem of synthetic cells (Shannon, 1948; Prigogine and Van Rysselberghe, 1963; Nicolis and Prigogine, 1977; Bak and Sneppen, 1993; Kauffman, 1993; Heinrich and Schuster, 1996; Elowitz and Leibler, 2000; Milo et al., 2002; Barabási and Oltvai, 2004; Cover and Thomas, 2005; Dill and Bromberg, 2010; Lorenz et al., 2011; Sauro, 2014; Briat et al., 2016; Zhang, 2012, 2013, 2016a-c, 2018, 2026a-d; Zhao et al., 2024; Chu et al., 2025). The goal of this study is to bridge this gap.

1.3 Core Proposition and Structure of This Study

The core proposition of this study is that the feasible design space for combining functional modules into complete cells is delineated by three progressive layers of constraints: the thermodynamic layer, the information layer, and the emergence layer (Shannon, 1948; Zhang, 2013, 2016b, 2018, 2026a-c; Solé and De Domenico, 2025). These three layers are not parallel categories but a progressively tightening screening. Thermodynamic constraints exclude a large class of module combination schemes that are energetically unsustainable. Information constraints further exclude coupling topologies that are energetically feasible but cannot reliably transmit signals in noise. Only when the system simultaneously satisfies the first two layers of constraints and reaches sufficiently high coupling strength at the emergence layer can the function of a

complete cell possibly appear (Zhang, 2013, 2016b).

The proposal of this framework is built upon various regularities revealed by existing research. Quantitative analysis of the energy costs of biochemical networks by non-equilibrium thermodynamics provides the foundation for the first layer (Tu, 2026; Zhang, 2016b, 2018). The relationship between modularity and robustness in complex network theory informs the second layer (Hintze and Adami, 2008; Khammash, 2016; Zhang, 2016a). Mathematical results on collective emergence from autocatalytic network theory ground the third layer (Solé and De Domenico, 2025; Sousa et al., 2015). The task of this study is to integrate these scattered insights into an operational framework oriented toward synthetic cell design and to derive testable predictions from it.

2 Current Status of Functional Modules and Combination Barriers

2.1 Core Functional Modules That Have Been Achieved

In current bottom-up synthetic cell research, the functional modules that have been widely reconstituted can be classified into four categories: metabolic cascades, protein synthesis, signal transduction, and energy supply (Wang and Zhou, 2012; Reverte López, 2025). Metabolic cascade modules typically consist of several enzymes that catalyze the sequential conversion of small molecule substrates within vesicles (Reverte López, 2025). Protein synthesis modules depend on ribosomes, tRNAs, amino acids, and energy regeneration systems in cell-free systems, and their resource consumption intensity is much higher than that of other modules (Inoue et al., 2014). Signal transduction modules include membrane receptors, G proteins, and second messenger systems, and their reconstitution requires correct positioning and orientation of membrane proteins on the vesicle membrane (Wang and Zhou, 2012). Energy supply modules achieve energy regeneration through light-driven proton pumps, ATP synthase, or chemical fuels (Tu, 2026).

The common feature of these modules is that each of them is a dissipative system requiring continuous energy input to maintain function (Zhang, 2013, 2016b, 2026a-c). This fact is often masked by externally supplied energy during independent module testing, but in a complete cell, all modules must compete for the same internal energy pool (Rijal et al., 2025). The work of Reverte López (2025) provides a comprehensive demonstration of how modular assembly within giant unilamellar vesicles can recapitulate essential cellular processes as independent operating modules. Each synthetic module, composed of molecular building blocks, confers specific functions on the lipid-based vesicles (Reverte López, 2025). However, the efficient integration of various modules within vesicles to yield a functioning reconstituted system exhibiting desired properties remains one of the major challenges (Noireaux et al., 2003; Noireaux and Libchaber, 2004; Reverte López, 2025).

2.2 Typical Failure Modes After Combination

When the above modules are simultaneously encapsulated, the failure modes exhibit several recurring categories (Rijal et al., 2025; Inoue et al., 2014). Resource competition-type failure manifests as high consumption of ATP or amino acids by one module leading to inactivation of other modules due to insufficient energy (Rijal et al., 2025; Youssef et al., 2026). Research demonstrated experimentally that cellular resource limitations create unintended interactions among synthetic gene circuit modules, compromising circuit modularity. This challenge is particularly pronounced in circuits with positive feedback, where uneven resource allocation can lead to Winner-Takes-All behavior, favoring one module at the expense of others (Rijal et al., 2025). Synthetic constructs not only burden host machinery but also compete with one another for transcriptional and translational resources, reducing each other's expression by up to 60 percent (Elowitz and Leibler, 2000; Kim and Winfree, 2011; Rijal et al., 2025).

Signal crosstalk-type failure occurs when unexpected chemical dialogue exists between modules (Wang

and Zhou, 2012). Interface conflict-type failure originates from competitive occupation of the membrane interface by modules, where multiple membrane protein modules may compete for limited membrane area or impose incompatible requirements on membrane curvature and lipid composition (Reverte López, 2025). Inoue et al. (2014) provided a rigorous robustness analysis of genetic circuits constructed by bottom-up strategy, showing that an intrinsic property in a circuit may be broken by internally connecting external circuits. Their analysis using structured singular value methods can guarantee the validity of bottom-up construction when the system satisfies certain robustness conditions (Inoue et al., 2014).

The common structural feature of these failure modes is noteworthy. They all involve indirect coupling between modules through shared environmental variables such as energy level, intermediate metabolite concentration, and membrane physical state. This coupling through the environment differs from the direct signal connections intentionally established by designers. It is a systemic emergent behavior that is difficult to eliminate through local modifications (Zhang, 2012, 2016b, 2018).

2.3 Implicit Assumptions of Module Independence and Their Limitations

Current modular design practice implicitly assumes a key hypothesis: as long as modules are chemically orthogonal, they can be independently designed and optimized (Inoue et al., 2014). This assumption originates from the idealized model of module isolation in electronic engineering. However, in physical systems at the cellular scale, orthogonality can at best ensure that there are no direct molecular recognition cross-reactions between modules, but it cannot prevent them from coupling through thermodynamic variables (Tu, 2026).

A specific example is pH coupling. The net reaction of the glycolysis module produces protons, which lowers the local pH, and the membrane receptor activity of many signal transduction modules is highly sensitive to pH. Even if these two modules are completely orthogonal at the molecular level, they still undergo substantial coupling through the shared thermodynamic variable of proton concentration (Zhang, 2016b). More generally, any two dissipative modules operating in the same closed system must inevitably couple through at least one shared thermodynamic intensive quantity (Tu, 2026). This means that complete independence of modules is a thermodynamically unattainable limit rather than an engineering goal that can be approached (Zhang, 2013). The task of design principles is not to pursue independence but to manage coupling.

3 Three-Layer Design Principle Framework

The three-layer design principle framework is proposed as follows (Appendix).

3.1 First Layer: Thermodynamic Constraints—Shared Resource Pool and Free Energy Dissipation

3.1.1 Energy Budget as the First Threshold for Combination Screening

For a synthetic cell system containing N functional modules, the total free energy dissipation rate $\dot{D}_{\text{total}}$ required to maintain a non-equilibrium steady state is at least equal to the sum of the dissipation rates of each module plus the additional dissipation generated by inter-module coupling (Tu, 2026):

$$\dot{D}_{\text{total}} \geq \sum_{i=1}^{N} \dot{D}_i + \dot{D}_{\text{coupling}}$$

The $\dot{D}_{\text{coupling}}$ term is zero when modules are independent but may be significantly positive in coupled systems (Tu, 2026). The direct corollary of this inequality is that the energy feasibility of a module combination scheme cannot be predicted by separately testing the energy requirements of each module (Rijal et al., 2025). The coupling term may make the total demand far exceed the sum of individual module demands, leading to an energy crisis when the system is actually combined (Inoue et al., 2014).

Tu (2026) reviewed recent advances in nonequilibrium thermodynamics of biochemical networks, organized around two central questions: why free energy dissipation is essential for enabling or enhancing

biological function, and how energetic costs constrain functional performance. Using representative systems including kinetic proofreading, accurate sensory adaptation, ultrasensitive responses, and synchronization of biochemical oscillators, Tu (2026) showed that this framework provides new insights into molecular mechanisms and reveals general thermodynamic principles governing biological functions, including the emergence of fundamental energy-performance trade-offs. For information processing in biochemical networks, the Landauer principle links the minimum free energy cost of information erasure to the amount of information (Tu, 2026). This means that there is a lower bound on the energy cost of signal transduction modules determined by the information processing rate, and this lower bound cannot be eliminated by optimizing molecular mechanisms (Ferrell, 2002; Tu, 2026).

3.1.2 The Concept of Module Thermodynamic Footprint

To transform thermodynamic constraints into operational design criteria, this study introduces the concept of module thermodynamic footprint. The thermodynamic footprint of a functional module refers to the change in the host system thermodynamic state when the module operates, including changes in local chemical potential, changes in proton concentration, and perturbations of membrane potential (Tu, 2026; Zhang, 2016b). The larger the thermodynamic footprint of a module, the higher the probability of unexpected thermodynamic coupling with other modules (Zhang, 2013).

The quantitative description of a module thermodynamic footprint can be defined by the changes in non-equilibrium steady-state parameters before and after module operation (Zhang, 2016a-b). For example, the thermodynamic footprint of an ATP-consuming module includes the magnitude of decrease in the ATP/ADP ratio, the magnitude of increase in inorganic phosphate concentration, and the resulting shift in phosphorylation potential (Li et al., 2025; Tu, 2026). When the footprints of multiple modules superimpose on the same thermodynamic variable, the system may be pushed far enough from its functional steady state that some modules cease to function (Rijal et al., 2025).

3.1.3 Corollary: Principle of Prioritizing Low-Footprint Modules

From the above analysis, a design principle can be derived. Under the premise of satisfying functional requirements, module variants with smaller thermodynamic footprints should be prioritized (Tu, 2026; Zhang, 2016a-b). For a metabolic cascade, this means prioritizing enzymatic pathways that consume less ATP per unit product conversion. For a signaling module, this means prioritizing signal transmission mechanisms that rely on stoichiometric amplification such as enzyme cascades rather than high-energy intermediates such as phosphorylated proteins. This principle echoes the experience of reducing host burden in synthetic biology but provides a thermodynamic basis for it. The essence of reducing burden is to decrease the module thermodynamic footprint, thereby lowering the probability of unexpected coupling (Rijal et al., 2025).

3.2 Second Layer: Information Constraints—Capacity and Noise of Coupling Channels

3.2.1 Inter-Module Coupling as an Information Transfer Problem

Among thermodynamically feasible module combination schemes, not all can achieve effective functional integration (Inoue et al., 2014). The coupling between modules, whether intentionally established by designers or unexpectedly emergent, is essentially an information transfer process (Shannon, 1948; Komorowski, 2025). The output of one module serves as the input of another module, and information is contaminated by noise and limited by channel capacity during transmission (Zhang, 2016c; Komorowski, 2025).

Komorowski (2025) derived closed-form information capacity formulas for canonical signaling models, quantifying how well different signaling models including binomial, multinomial, Poisson, Gaussian, and Gamma distributions (Zhang, 2026d) discriminate among input signals. These expressions clarify how key features such as signal or response range, noise scaling, pathway length, and receiver diversity shape the theoretical limits of sensing (Komorowski, 2025). The framework generalizes to any system where noisy

input-output relationships constrain transmission fidelity, including synthetic biology and engineered communication channels (Komorowski, 2025). The coupling channel capacity C between two modules defines the maximum amount of information that can be reliably transmitted per unit time under a given noise level (Komorowski, 2025). If the output change frequency of module A exceeds the channel capacity, module B will be unable to distinguish true signal changes from noise fluctuations, and coupling fails (Komorowski, 2025). Channel capacity is determined by the molecular mechanism of coupling, with diffusion coupling limited by molecular diffusion coefficients and distance, membrane contact coupling limited by contact area and receptor density, and metabolite-mediated coupling limited by enzyme catalytic rates and metabolite turnover times.

3.2.2 Crosstalk as Inter-Channel Interference

When multiple coupling channels exist in the system, interference between channels, or crosstalk, becomes a factor limiting the overall information transfer capability of the system (Komorowski, 2025; Zhang, 2016d). In synthetic cells, common sources of crosstalk include cross-reactions of signaling molecules with multiple receptor subtypes, competitive consumption of metabolic intermediates by multiple enzymes, and simultaneous influence of ion gradients on the activity of multiple membrane proteins (Levine and Hwa, 2007).

Formalizing crosstalk as inter-channel interference allows us to borrow results from multi-user information theory to derive design boundaries (Komorowski, 2025). A directly relevant conclusion is that when M coupling channels share the same physical transmission medium, there is an upper bound on the total information transfer capacity of the system, determined by the physical properties of the medium. As the number of channels increases, the effective capacity of each channel decreases (Komorowski, 2025). This means that increasing the number of modules or coupling complexity does not always enhance system function. Beyond a certain threshold, it may reduce overall information processing capability due to channel interference (Rijal et al., 2025).

3.2.3 Corollary: Orthogonality Design of Coupling Channels

The design principle that can be derived from information constraints is that in multi-module systems, coupling channels should achieve information orthogonality as much as possible (Komorowski, 2025). This means that the information transfer channels between different module pairs are physically distinguishable, so that the signal of one channel does not leak into another. This differs from chemical orthogonality at the molecular level. Chemical orthogonality focuses on non-crossing molecular recognition, while information orthogonality focuses on the separability of signal channels in a noisy environment (Komorowski, 2025). A chemically orthogonal system may not be informationally orthogonal. For example, two modules using different signaling molecules but sharing the same downstream effector are still coupled at the information level.

3.3 Third Layer: Emergence—Coupling Strength and New System Behaviors

3.3.1 From Composability to Emergence

The first two layers of constraints answer the question of which module combinations are feasible, but they do not answer under what conditions feasible combinations produce integrated functions (Zhang, 2016b, 2018). The functions of a complete cell, such as maintenance of metabolic homeostasis, adaptive responses to environmental changes, and self-replication capability, are not properties of any single module but are new behaviors that emerge at the system level when inter-module interactions reach sufficient strength (Zhang, 2013, 2016b).

Autocatalytic network theory provides a mathematical foundation for understanding this emergence (Solé and De Domenico, 2025). The concept of autocatalytic sets is a foundational idea in understanding the origin of life and self-organization in complex systems. The idea revolves around the notion that life could emerge

from a network of molecules that catalyze the formation of each other in a mutually reinforcing manner, with these autocatalytic sets being collections of molecules where each member is produced through reactions catalyzed by other members of the set (Solé and De Domenico, 2025). Solé and De Domenico (2025) proposed that as the number of molecules and reactions increases, there is a critical point, akin to a phase transition (Haken, 1977, 1983, 2004; Zhang, 2016b, 2026a, 2026c), at which large connected networks of reactions are inevitable, including autocatalytic sets (Solé and De Domenico, 2025). In a random reaction network, the likelihood that a reaction exists between two molecules is determined by a probability parameter. When the initial set exceeds a critical diversity, autocatalytic reactions generate large molecular species in abundance (Solé and De Domenico, 2025). The deep implication of this result is that emergence is not designed but happens by itself when system parameters cross critical conditions (Zhang, 2016b, 2026a, 2026c). The designer's role is to create conditions that make emergence possible, rather than to directly specify the specific form of emergence (Zhang, 2013).

3.3.2 Coupling Strength Critical Condition

Migrating this insight to the module combination problem, a hypothesis of coupling strength critical condition can be proposed. There exists a coupling strength threshold κ_c. When the effective coupling strength between modules $\kappa < \kappa_c$, the system behaves as simple coexistence of modules. When κ approaches κ_c, the system enters a critical region, where long-range correlations and amplified responses may appear. When $\kappa > \kappa_c$, the system enters an integrated state, where collective dynamical behaviors that do not exist when modules are independent appear, such as coupled oscillations, traveling wave propagation, or global synchronization (Zhang, 2013, 2016b).

This hypothesis has parallels in the study of critical phenomena in biological networks. Research on hierarchical modular neuronal networks has shown that network topology significantly influences critical behavior, with sparse modular architectures sustaining criticality more robustly than fully connected ones (Wang et al., 2025). Homeostatic mechanisms can stabilize activity near criticality, even as modular interactions introduce structural inhomogeneities (Rusch et al., 2025). These inhomogeneities can produce quasicritical dynamics and Griffiths-like phases, broadening the range of near-critical behavior (Rusch et al., 2025). The concept of self-organized criticality (SOC) describes dynamical systems with extended spatial and temporal degrees of freedom that naturally evolve to a critical state that is weakly stable and displays activity avalanches with power-law distributions of size and duration (Rusch et al., 2025; Zhang, 2016b, 2018).

The testable corollaries of the coupling strength critical condition include the following. In coupling strength scanning experiments, a nonlinear jump in system response should be observed rather than a smooth transition (Strogatz, 2015; Zhang, 2013, 2016b). The system near the critical point should exhibit abnormally high sensitivity to small perturbations, known as critical slowing down (Zhang, 2016b, 2026a, 2026c). The system function in the integrated state cannot be decomposed into a linear combination of module functions (Solé and De Domenico, 2025).

3.3.3 Design Implications of Emergent Functions

If emergent functions are indeed controlled by the coupling strength critical condition, then the design strategy needs to shift from precisely specifying the parameters of each module to tuning the system to near the critical region (Zhang, 2013). The fundamental difficulty of this strategy is that the critical region is usually accompanied by increased sensitivity to noise and decreased stability (Zhang, 2016b, 2026a, 2026c). A system at the critical point has maximum dynamic range and response sensitivity but is also most easily perturbed away from the functional state (Zhang, 2013, 2016b). Therefore, the design of emergent functions is essentially a trade-off problem between sensitivity and stability.

The concept of self-organized criticality may provide clues to resolving this trade-off (Zhang, 2012, 2016b,

2018). Research on hierarchical modular networks has demonstrated that networks with modular structure can extend the parameter region of coupling strength over which critical states are reached compared to non-modular networks, enhancing the robustness of criticality as well as function (Rusch et al., 2025). If the system has an intrinsic mechanism to maintain itself near the critical state, for example through slow negative feedback regulating coupling strength, then it can maintain high sensitivity while avoiding excessive instability (Zhang, 2013, 2016b). Whether such self-tuning mechanisms exist in synthetic cells is a question awaiting experimental exploration.

4 Formalization and Predictions of the Theoretical Framework

4.1 Module Compatibility Matrix

The above three-layer principles can be integrated into an operational design tool, the module compatibility matrix (Zhang, 2018). For a candidate module set $M=\{M_1,M_2,\ldots,M_N\}$, define an $N\times N$ matrix **K**, whose element K_{ij} represents the compatibility index between module *i* and module *j* (Zhang, 2018):

$$K_{ij}=f(\Delta\mu_{ij}^{\text{overlap}} / \Delta\mu_{i}^{\text{footprint}} ,C_{ij}/R_{ij},\kappa_{ij})$$

where $\Delta\mu_{ij}^{\text{overlap}}$ is the degree of overlap of the two modules' footprints in thermodynamic variable space, C_{ij} is the coupling channel capacity, R_{ij} is the required information transfer rate, and κ_{ij} is the effective coupling strength (Tu, 2026; Komorowski, 2025; Zhang, 2013, 2016b).

A natural choice for the functional form is a product of normalized factors. Define the thermodynamic compatibility factor as

$$\eta_{ij}=1-|\Delta\boldsymbol{\phi}_i\cdot\Delta\boldsymbol{\phi}_j| / (|\Delta\boldsymbol{\phi}_i|\,|\Delta\boldsymbol{\phi}_j|)$$

the information compatibility factor as

$$\xi_{ij}=\min(1,C_{ij}/R_{ij})$$

and the emergence factor as

$$\zeta_{ij}=\tanh(\kappa_{ij}/\kappa_c)$$

Then the compatibility index is given by

$$K_{ij}=\eta_{ij}\cdot\xi_{ij}\cdot\zeta_{ij}$$

This functional form ensures that $K_{ij}\in[0,1]$, with values near 1 indicating high compatibility and values near 0 indicating incompatibility. The diagonal elements $K_{ii}=1$ by convention.

The structural properties of the matrix have design implications (Zhang, 2012, 2018). If the matrix is approximately diagonal with off-diagonal elements much smaller than diagonal elements, modules are approximately independent, and the system behaves as assembled rather than integrated. If the matrix has dense strong off-diagonal elements, extensive strong coupling exists between modules, and the system may

enter an integrated state but also faces stability risks (Rijal et al., 2025). The design goal is to identify those element configurations that bring the system near the critical region (Zhang, 2013, 2016b).

A connection to cooperative game theory can sharpen the formalization. The compatibility matrix **K** can be interpreted as defining a characteristic function $v(S)$ for each subset $S \subseteq M$ of modules:

$$v(S)=\sum_{i\in S} K_{ii} + \sum_{i,j\in S, i<j} K_{ij} \min(\kappa_{ij}, \kappa_{ji})$$

The Shapley value of module i in this cooperative game is

$$\phi_i(v)=\sum_{S\subseteq M\setminus\{i\}} |S|!\,(N-|S|-1)!\,[v(S\cup\{i\})-v(S)]/N!$$

This Shapley value quantifies the marginal contribution of module i to the total compatibility of the system, accounting for all possible orderings of module addition. Modules with high Shapley values contribute disproportionately to system integration, while those with low or negative values may be candidates for removal or redesign. This provides a principled approach to module selection and prioritization in synthetic cell design.

4.2 Key Assumptions and Testable Predictions

The framework in this study is built on three core assumptions, each corresponding to experimentally testable predictions (Zhang, 2016b, 2026; Appendix).

Assumption 1 (Thermodynamic screening assumption): The thermodynamic feasibility of a module combination scheme can be approximately judged by the vector sum of the thermodynamic footprints of individual modules (Tu, 2026). Prediction: Module combinations with highly overlapping footprints should exhibit more frequent functional loss in experiments than combinations with non-overlapping footprints, and the time scale of functional loss correlates with the degree of footprint overlap (Rijal et al., 2025).

Assumption 2 (Information channel constraint assumption): The effective capacity of coupling channels limits the complexity of module coupling that the system can support (Komorowski, 2025). Prediction: Under the condition of fixed total information transfer capacity, increasing the number of coupled modules should lead to a decrease in the information fidelity of each coupling, and when the number of modules exceeds the upper limit allowed by channel capacity, system function collapses (Gillespie, 1977; Komorowski, 2025).

Assumption 3 (Emergence criticality assumption): There exists a critical interval of coupling strength in which the system exhibits maximum functional integration and response sensitivity (Zhang, 2016e). Prediction: In coupling strength parameter scanning experiments, a non-monotonic change in functional integration should be observed, with a peak appearing near the critical coupling strength (Rusch et al., 2025).

4.3 Experimental Validation Strategies

The above predictions can be tested through the following strategies (Rijal et al., 2025; Komorowski, 2025). In testing the thermodynamic screening assumption, a set of protein synthesis module variants with different ATP consumption rates can be designed and combined with the same metabolic cascade module, measuring the relationship between the functional maintenance time of the combined system and the ATP consumption rate of the protein synthesis module (Rijal et al., 2025). In testing the information channel constraint assumption, light-controlled gene circuits can be used to construct a tunable number of coupled modules controlled by orthogonal light wavelengths, measuring the number of independent signal channels that the system can simultaneously maintain when the total signaling molecule pool is fixed (Bialek and Setayeshgar, 2005; Komorowski, 2025). In testing the emergence criticality assumption, the coupling strength parameter can be

scanned by changing receptor density on the membrane or the diffusion rate of signaling molecules, measuring the change in the form of the system response function (Rusch et al., 2025).

The work of Reverte López (2025) provides a concrete experimental platform for such studies. The in vitro co-reconstitution of the actomyosin contraction module with the MinDE system under optimized encapsulation conditions demonstrated that integrating both modules yields effective MinDE-driven positioning of actomyosin rings at the equator of giant unilamellar vesicles (Reverte López, 2025). The MinDE system self-organizes on membranes through ATP-driven attachment-detachment cycles, forming dynamic and quasi-stationary patterns (Reverte López, 2025). This system provides a well-characterized module whose coupling strength with the membrane can be systematically varied.

The common challenge of these experiments is how to independently control the three variables of thermodynamic footprint, channel capacity, and coupling strength (Bialek and Setayeshgar, 2005; Zhang, 2016b). Microfluidic technology and optogenetic tools provide possible means for this. Microfluidics can control the composition and external environment of vesicles, and optogenetic tools can precisely control module activity in time and space (Rijal et al., 2025).

5 Discussion

5.1 Relationship with Existing Design Frameworks

The three-layer framework proposed in this study has several points of dialogue with existing research. The chassis-circuit paradigm in synthetic biology implicitly treats the chassis as a passive resource provider, while the thermodynamic layer of this study redefines the chassis-module relationship as bidirectional. Modules not only consume chassis resources but also alter the chassis thermodynamic state, thereby feeding back on their own and other modules functions (Tu, 2026). This perspective is consistent with discussions of cellular economics but extends them from quantitative analysis of metabolic load to a general framework of thermodynamic coupling (Rijal et al., 2025).

The connection with autocatalytic network theory is more direct (Solé and De Domenico, 2025). Their results on the emergence of collectively autocatalytic sets provide a mathematical precedent for the coupling strength critical condition, but autocatalytic network theory typically deals with the relationship between molecular species and reaction numbers rather than coupling between functional modules (Solé and De Domenico, 2025). Elevating the concept of critical emergence from the molecular network level to the functional module level is a theoretical innovation of the framework in this study (Zhang, 2013, 2016b).

The formalization using the Shapley value connects the framework to cooperative game theory, providing a rigorous mathematical basis for quantifying module contributions. This connection allows the compatibility matrix to be interpreted not merely as a heuristic scoring system but as a well-defined mathematical object with properties that can be analyzed using established results from game theory.

5.2 Theoretical Limitations and Open Questions

The major limitation of the framework in this study lies in the fact that its definition of module depends on functional division, and functional division itself has a certain degree of subjectivity (Zhang, 2012, 2018). An entity regarded as a single module by a designer, such as a multi-enzyme complex, may be regarded as an assembly of multiple modules at another analysis scale (Reverte López, 2025). This scale dependence means that the construction of the compatibility matrix requires explicit specification of the analysis granularity (Rijal et al., 2025).

The second limitation is that the mathematical foundation of the emergence criticality assumption is not yet fully solid (Zhang, 2013, 2016b). The critical phenomena in autocatalytic networks have rigorous mathematical proofs, but the critical condition for coupling strength between functional modules is currently

more of an inspiring analogy than a proven theorem (Solé and De Domenico, 2025). Establishing a rigorous mathematical mapping from inter-module coupling to system emergence behavior is an important direction for future theoretical work.

Another unresolved question is the controllability of the critical region (Zhang, 2012, 2016b, 2018). If emergent functions indeed require the system to be near criticality, then how to achieve and maintain the critical state in synthetic cells is a core technical challenge (Zhang, 2016b). Research on neuronal networks has shown that homeostatic mechanisms can stabilize activity near criticality, but whether analogous mechanisms can be engineered into synthetic cells remains unknown (Rusch et al., 2025).

5.3 Implications for Synthetic Cell Engineering

Despite the above limitations, the framework in this study has several direct implications for synthetic cell engineering practice (Rijal et al., 2025). The evaluation of module combinations should not only focus on molecular-level orthogonality but also on the overlap of thermodynamic footprints and the information capacity of coupling channels (Tu, 2026). The goal of system design should not be to maximize module independence but to tune the system to a window where coupling strength allows integrated functions to emerge (Zhang, 2013, 2016b). Failed module combinations may provide more design information than successful combinations. A system that collapses due to thermodynamic coupling precisely reveals which thermodynamic variables are key coupling pathways.

From a longer-term perspective, if the functions of complete cells indeed follow some emergence logic, then the ultimate success of bottom-up synthetic biology may not depend on the fine optimization of individual modules but on the rational design of inter-module coupling topology (Zhang, 2016b). This perspective redirects the core problem of synthetic biology from how to build better modules to how to build better inter-module relationships (Zhang, 2012, 2018).

Appendix
Detailed Computational Methods

A.1 Thermodynamic Footprint Calculation

The thermodynamic footprint of a functional module is defined as the vector of changes in thermodynamic state variables of the host system upon module operation (Tu, 2026; Zhang, 2016b):

$$\Delta\phi_i=(\Delta\mu_{\mathrm{ATP}},\Delta\mathrm{pH},\Delta\Psi,\Delta[\mathrm{P}_i],\ldots)_i$$

For an ATP-consuming module, the phosphorylation potential shift is calculated as:

$$\Delta\mu_{\mathrm{ATP}}=RT\ln([\mathrm{ATP}][\mathrm{P}_i]/[\mathrm{ADP}])_{\mathrm{after}}-RT\ln([\mathrm{ATP}][\mathrm{P}_i]/[\mathrm{ADP}])_{\mathrm{before}}$$

where R is the gas constant (8.314 J moł ¹ K⁻¹), T is the absolute temperature (310 K for biological systems), and concentrations are in mol⁻L ¹. The phosphorylation potential $\Delta G_p = \Delta G^{0'} + RT\ln([\mathrm{ATP}][\mathrm{P}_i]/[\mathrm{ADP}])$ provides a direct measure of the free energy available from ATP hydrolysis under cellular conditions (Tu, 2026).

A.2 Information Capacity Calculation

The information capacity of a coupling channel is calculated using the Fisher information framework (Komorowski, 2025). For a signaling system described by a conditional distribution $P(Y|X = x)$, the mutual information is:

$$\mathrm{MI}(X,Y)=\int_X\int_Y P(x,y)\log_2 (P(y|x)/P(y))dxdy$$

For a Gaussian channel with signal-dependent noise variance σ_n^2 and signal variance σ_s^2:

$$C=\log_2(1+ \sigma_s^2/\sigma_n^2)/2$$

Following Komorowski (2025), for a Poisson signaling model with mean response $\lambda(x)$:

$$C=\log_2(2\pi e)/2+\int p(x)\ln(\lambda'(x)^2/\lambda(x))dx/2$$

For a binomial model with n independent receptors and activation probability $p(x)$:

$$C=\log_2(2\pi e)/2+\int p(x)\ln(np'(x)^2/(p(x)(1- p(x))))dx/2$$

These closed-form expressions allow direct computation of channel capacity from biophysical parameters without computationally expensive mutual information estimation (Komorowski, 2025).

A.3 Coupling Strength Critical Condition Simulation

The coupling strength critical condition is simulated using a system of coupled differential equations (Zhang, 2016b, 2018):

$$d\mathbf{x}_i/dt=\mathbf{F}_i(\mathbf{x}_i)+\kappa \sum_{j\neq i} \mathbf{G}_{ij} (\mathbf{x}_i, \mathbf{x}_j)$$

where $\mathbf{F}_i$ describes the internal dynamics of module i, $\mathbf{G}_{ij}$ describes the coupling function between modules i and j, and κ is the coupling strength parameter. The critical value κ_c is determined by analyzing the eigenvalues of the Jacobian matrix $\mathbf{J}$ at the steady state $\mathbf{x}^*$:

$$J_{ij}=\partial[F_i(\mathbf{x}_i)+\kappa\sum_{k\neq i} G_{ik}(\mathbf{x}_i,\mathbf{x}_k)]/\partial x_j |_{\mathbf{x}=\mathbf{x}^*}$$

The system undergoes a Hopf bifurcation when a pair of complex conjugate eigenvalues crosses the imaginary axis, signaling the onset of oscillatory behavior. The critical coupling strength κ_c is the value at which the real part of the leading eigenvalue becomes zero:

$$\mathrm{Re}[\lambda_{\max}(\kappa_c)]=0$$

For a two-module system with linear coupling $G_{12} = x_2 - x_1$ and $G_{21} = x_1 - x_2$, the critical coupling strength can be derived analytically. Following the analysis of coupled oscillators (Strogatz, 2015), for identical modules with natural frequency ω_0 and damping γ, the critical coupling is:

$\kappa_c = \gamma \omega_0 / 2$

This result provides a quantitative criterion for predicting when two coupled modules will transition from independent behavior to synchronized dynamics.

A.4 Module Compatibility Matrix Construction

The module compatibility matrix **K** is constructed according to the following procedure:

Step 1: For each pair of modules (i, j), compute the thermodynamic footprint overlap:

$\eta_{ij} = 1 - |\Delta\boldsymbol{\phi}_i \cdot \Delta\boldsymbol{\phi}_j| / (|\Delta\boldsymbol{\phi}_i| \, |\Delta\boldsymbol{\phi}_j|)$

where $\Delta\boldsymbol{\phi}_i$ and $\Delta\boldsymbol{\phi}_j$ are the thermodynamic footprint vectors defined in A.1.

Step 2: Compute the information compatibility factor:

$\xi_{ij} = \min(1, C_{ij}/R_{ij})$

where C_{ij} is the channel capacity computed from A.2 and R_{ij} is the required information transfer rate determined by the functional specifications.

Step 3: Estimate the effective coupling strength κ_{ij} from the Jacobian analysis described in A.3.

Step 4: Compute the emergence factor:

$\zeta_{ij} = \tanh(\kappa_{ij}/\kappa_c)$

where κ_c is the critical coupling strength from A.3.

Step 5: Compute the compatibility index:

$K_{ij} = \eta_{ij} \cdot \xi_{ij} \cdot \zeta_{ij}$

The diagonal elements $K_{ii} = 1$ by convention.

Step 6: Compute the Shapley value for each module using the characteristic function:

$v(S) = \sum_{i \in S} K_{ii} + \sum_{i,j \in S, i<j} K_{ij} \min(\kappa_{ij}, \kappa_{ji})$

The Shapley value is then:

$\phi_i(v) = \sum_{S \subseteq M \setminus \{i\}} |S|! \, (N - |S| - 1)! \, [v(S \cup \{i\}) - v(S)] / N!$

A.5 Numerical Parameters

For the numerical simulations described in this study, the following parameter values were used: R = 8.314 J mol^{-1} K^{-1}, T = 310 K, κ_B = 1.381*10^{-23} J K^{-1}, typical ATP concentration [ATP] = 1-3 mM, typical ADP concentration [ADP] = 0.1-0.5 mM, typical inorganic phosphate concentration [P_i] = 1-10 mM. These values are consistent with physiological conditions reported in the literature (Tu, 2026; Li et al., 2025). For the coupled oscillator analysis, typical values of ω_0 = 0.1 min^{-1} and γ = 0.05 min^{-1} were used, corresponding to relaxation timescales observed in biochemical oscillators (Holló et al., 2026).

A.6 Computational Implementation

Here is the complete, self-contained HTML document with all interface text and comments. It implements the computational methods above as interactive calculators for thermodynamic footprints, information capacity, coupling strength, and module compatibility matrices.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Appendix: Detailed Computational Methods - Interactive Tool</title>
<style>
  * { box-sizing: border-box; }
  body {
    font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, 'Helvetica Neue', Arial, sans-serif;
    margin: 0; padding: 0;
    background: #f5f5f7;
    color: #1d1d1f;
    line-height: 1.6;
  }
  header {
    background: #1d1d1f;
    color: white;
    padding: 30px 20px;
    text-align: center;
  }
  header h1 { margin: 0 0 8px; font-size: 24px; font-weight: 600; }
  header p { margin: 0; color: #a1a1a6; font-size: 14px; }
  .tabs {
    display: flex;
    background: #fff;
    border-bottom: 1px solid #d2d2d7;
    overflow-x: auto;
```

```
    position: sticky;
    top: 0;
    z-index: 10;
  }
  .tab {
    padding: 14px 20px;
    cursor: pointer;
    border: none;
    background: none;
    font-size: 14px;
    color: #6e6e73;
    white-space: nowrap;
    border-bottom: 2px solid transparent;
    transition: all 0.2s;
    font-family: inherit;
  }
  .tab:hover { color: #1d1d1f; background: #f5f5f7; }
  .tab.active { color: #0071e3; border-bottom-color: #0071e3; font-weight: 500; }
  .content {
    max-width: 1100px;
    margin: 0 auto;
    padding: 20px;
  }
  .panel { display: none; }
  .panel.active { display: block; animation: fadeIn 0.3s; }
  @keyframes fadeIn { from { opacity: 0; transform: translateY(4px); } to { opacity: 1; transform: none; } }
  .card {
    background: #fff;
    border-radius: 12px;
    padding: 24px;
    margin-bottom: 20px;
    box-shadow: 0 1px 3px rgba(0,0,0,0.06);
  }
  .card h2 { margin-top: 0; font-size: 18px; font-weight: 600; }
  .card h3 { font-size: 14px; color: #6e6e73; font-weight: 500; margin-top: 20px; }
  .input-grid {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(170px, 1fr));
    gap: 12px;
    margin: 16px 0;
  }
  .field { display: flex; flex-direction: column; }
  .field label { font-size: 12px; color: #6e6e73; margin-bottom: 4px; font-weight: 500; }
  .field input, .field select {
```

```
    padding: 8px 10px;
    border: 1px solid #d2d2d7;
    border-radius: 6px;
    font-size: 14px;
    font-family: inherit;
    background: #fff;
  }
  .field input:focus, .field select:focus {
    outline: none;
    border-color: #0071e3;
    box-shadow: 0 0 0 3px rgba(0,113,227,0.1);
  }
  button {
    background: #0071e3;
    color: white;
    border: none;
    padding: 10px 20px;
    border-radius: 8px;
    font-size: 14px;
    cursor: pointer;
    font-family: inherit;
    transition: background 0.2s;
    margin-right: 8px;
  }
  button:hover { background: #0062cc; }
  button.secondary { background: #e8e8ed; color: #1d1d1f; }
  button.secondary:hover { background: #dcdce1; }
  button:active { transform: scale(0.98); }
  .result {
    background: #f5f5f7;
    border-radius: 8px;
    padding: 16px;
    margin-top: 16px;
    font-family: 'SF Mono', Monaco, 'Cascadia Code', 'Roboto Mono', monospace;
    font-size: 13px;
    white-space: pre-wrap;
    word-break: break-word;
    max-height: 500px;
    overflow-y: auto;
    line-height: 1.7;
  }
  .result .hl { color: #0071e3; font-weight: 600; }
  .result .ok { color: #1e7d32; }
  .result .warn { color: #c41e3a; }
```

```
table {
    width: 100%;
    border-collapse: collapse;
    margin-top: 12px;
    font-size: 13px;
}
th, td {
    padding: 8px 10px;
    text-align: right;
    border-bottom: 1px solid #e8e8ed;
    font-family: 'SF Mono', Monaco, monospace;
}
th {
    background: #f5f5f7;
    font-weight: 600;
    color: #6e6e73;
    font-size: 12px;
    position: sticky;
    top: 0;
}
td:first-child, th:first-child { text-align: left; }
.formula {
    background: #fafafa;
    padding: 12px 16px;
    border-left: 3px solid #0071e3;
    border-radius: 4px;
    font-family: 'SF Mono', Monaco, monospace;
    font-size: 13px;
    margin: 12px 0;
    overflow-x: auto;
    color: #333;
}
.note { font-size: 12px; color: #6e6e73; margin-top: 8px; }
.matrix-container { overflow-x: auto; margin-top: 12px; }
.module-row {
    display: grid;
    grid-template-columns: 80px repeat(4, 1fr);
    gap: 8px;
    align-items: center;
    margin-bottom: 8px;
    padding: 8px;
    background: #fafafa;
    border-radius: 6px;
}
```

```
    .module-row .label {
        font-size: 13px;
        font-weight: 600;
        color: #0071e3;
    }
    .module-row input {
        padding: 6px 8px;
        border: 1px solid #d2d2d7;
        border-radius: 5px;
        font-size: 13px;
        font-family: inherit;
        width: 100%;
    }
</style>
</head>
<body>

<header>
    <h1>Appendix: Detailed Computational Methods</h1>
    <p>General Design Principles for Combining Functional Modules into Complete Cells — Interactive Computation Tool</p>
</header>

<div class="tabs">
    <button class="tab active" data-target="a1">A.1 Thermodynamic Footprint</button>
    <button class="tab" data-target="a2">A.2 Information Capacity</button>
    <button class="tab" data-target="a3">A.3 Critical Coupling Strength</button>
    <button class="tab" data-target="a4">A.4 Compatibility Matrix &amp;amp; Shapley</button>
    <button class="tab" data-target="a5">A.5 Numerical Parameters</button>
    <button class="tab" data-target="a6">A.6 Computational Implementation</button>
</div>

<div class="content">

<!-- ====================== A.1 ====================== -->
<div class="panel active" id="a1">
    <div class="card">
        <h2>A.1 Thermodynamic Footprint Calculation</h2>
        <p>The thermodynamic footprint is defined as the vector of changes in the host system's thermodynamic state variables upon module operation. This tool computes the phosphorylation potential shift and ATP hydrolysis free energy change for an ATP-consuming module.</p>

        <div class="formula">
Δφ_i = (Δμ_ATP, ΔpH, ΔΨ, Δ[P_i], ...)_i
```

$$\Delta\mu_{ATP} = RT\cdot\ln([ATP][P_i]/[ADP])_{after} - RT\cdot\ln([ATP][P_i]/[ADP])_{before}$$

$$\Delta G_{hydrolysis} = \Delta G^{o\prime} + RT\cdot\ln([ADP][P_i]/[ATP])$$

```
    </div>

    <h3>Environment Parameters</h3>
    <div class="input-grid">
      <div class="field">
        <label>Temperature T (K)</label>
        <input type="number" id="a1-T" value="310" step="0.1">
      </div>
      <div class="field">
        <label>ΔG°' (kJ/mol)</label>
        <input type="number" id="a1-dG0" value="-30.5" step="0.1">
      </div>
    </div>

    <h3>Before Module Operation</h3>
    <div class="input-grid">
      <div class="field">
        <label>[ATP] (mM)</label>
        <input type="number" id="a1-atp-b" value="3.0" step="0.1">
      </div>
      <div class="field">
        <label>[ADP] (mM)</label>
        <input type="number" id="a1-adp-b" value="0.5" step="0.1">
      </div>
      <div class="field">
        <label>[P_i] (mM)</label>
        <input type="number" id="a1-pi-b" value="5.0" step="0.1">
      </div>
    </div>

    <h3>After Module Operation</h3>
    <div class="input-grid">
      <div class="field">
        <label>[ATP] (mM)</label>
        <input type="number" id="a1-atp-a" value="1.5" step="0.1">
      </div>
      <div class="field">
        <label>[ADP] (mM)</label>
        <input type="number" id="a1-adp-a" value="2.0" step="0.1">
      </div>
```

```
      <div class="field">
        <label>[P_i] (mM)</label>
        <input type="number" id="a1-pi-a" value="6.5" step="0.1">
      </div>
    </div>

    <button onclick="calcA1()">Compute Thermodynamic Footprint</button>
    <button class="secondary" onclick="resetA1()">Reset Defaults</button>

    <div id="a1-result" class="result" style="display:none;"></div>
  </div>
</div>

<!-- ===================== A.2 ===================== -->
<div class="panel" id="a2">
  <div class="card">
    <h2>A.2 Information Capacity Calculation</h2>
    <p>Computes the information capacity of a coupling channel using the Fisher information framework. Supports Gaussian channels, Poisson signaling models, and binomial receptor models.</p>

    <div class="formula">
Gaussian channel:  C = ½·log₂(1 + σ_s²/σ_n²)

Poisson model:  C = ½·log₂(2πe) + ½· ∫ p(x)·ln(λ'(x)²/λ(x)) dx
                  where λ(x) = λ₀ + λ₁·x

Binomial model:  C = ½·log₂(2πe) + ½· ∫ p(x)·ln(n·p'(x)²/(p(x)(1−p(x)))) dx
                  where p(x) = 1/(1+exp(−(x−x₀)/s))
    </div>

    <h3>Gaussian Channel</h3>
    <div class="input-grid">
      <div class="field">
        <label>Signal variance σ_s²</label>
        <input type="number" id="a2-gs" value="1.0" step="0.1">
      </div>
      <div class="field">
        <label>Noise variance σ_n²</label>
        <input type="number" id="a2-gn" value="0.1" step="0.01">
      </div>
    </div>
    <button onclick="calcA2Gauss()">Compute Gaussian Channel Capacity</button>

    <h3>Poisson Signaling Model</h3>
```

```
<div class="input-grid">
  <div class="field">
    <label>λ₀ (baseline rate)</label>
    <input type="number" id="a2-p-l0" value="1.0" step="0.1">
  </div>
  <div class="field">
    <label>λ₁ (signal gain)</label>
    <input type="number" id="a2-p-l1" value="1.0" step="0.1">
  </div>
  <div class="field">
    <label>Signal mean μ</label>
    <input type="number" id="a2-p-mu" value="0.0" step="0.1">
  </div>
  <div class="field">
    <label>Signal std σ</label>
    <input type="number" id="a2-p-sig" value="1.0" step="0.1">
  </div>
</div>
<button onclick="calcA2Poisson()">Compute Poisson Model Capacity</button>

<h3>Binomial Receptor Model</h3>
<div class="input-grid">
  <div class="field">
    <label>Number of receptors n</label>
    <input type="number" id="a2-b-n" value="100" step="1">
  </div>
  <div class="field">
    <label>Activation midpoint x₀</label>
    <input type="number" id="a2-b-x0" value="0.0" step="0.1">
  </div>
  <div class="field">
    <label>Steepness parameter s</label>
    <input type="number" id="a2-b-s" value="0.5" step="0.1">
  </div>
  <div class="field">
    <label>Signal mean μ</label>
    <input type="number" id="a2-b-mu" value="0.0" step="0.1">
  </div>
  <div class="field">
    <label>Signal std σ</label>
    <input type="number" id="a2-b-sig" value="1.0" step="0.1">
  </div>
</div>
<button onclick="calcA2Binom()">Compute Binomial Model Capacity</button>
```

```
    <div id="a2-result" class="result" style="display:none;"></div>
  </div>
</div>

<!-- ==================== A.3 ==================== -->
<div class="panel" id="a3">
  <div class="card">
    <h2>A.3 Critical Coupling Strength Condition</h2>
    <p>Performs a Hopf bifurcation analysis on a two-module linearly coupled system to compute the critical coupling strength κ_c. Also provides the analytical synchronization threshold for coupled oscillators.</p>

    <div class="formula">
Coupled ODE system:
  dx_i/dt = F_i(x_i) + κ·Σ_{j≠i} G_ij(x_i, x_j)

Jacobian matrix:
  J_ij = ∂/∂x_j [F_i(x_i) + κ·Σ_{k≠i} G_ik(x_i, x_k)] |_{x = x*}

Hopf bifurcation condition:
  Re[λ_max(κ_c)] = 0

Two-module analytical solution:
  κ_c = γ·ω₀ / 2
    </div>

    <h3>Analytical Calculation</h3>
    <div class="input-grid">
      <div class="field">
        <label>Natural frequency ω₀ (min⁻¹)</label>
        <input type="number" id="a3-w0" value="0.1" step="0.01">
      </div>
      <div class="field">
        <label>Damping rate γ (min⁻¹)</label>
        <input type="number" id="a3-gamma" value="0.05" step="0.01">
      </div>
    </div>
    <button onclick="calcA3Analytic()">Compute Analytical Critical Value</button>

    <h3>Numerical Simulation: Two-Module Coupled Oscillator</h3>
    <p class="note">System: ẍ₁ + γẋ₁ + ω₀²x₁ = κ(x₂ − x₁),  ẍ₂ + γẋ₂ + ω₀²x₂ = κ(x₁ − x₂)</p>
    <div class="input-grid">
      <div class="field">
        <label>Natural frequency ω₀</label>
```

```
            <input type="number" id="a3-n-w0" value="0.1" step="0.01">
          </div>
          <div class="field">
            <label>Damping rate γ</label>
            <input type="number" id="a3-n-gamma" value="0.05" step="0.01">
          </div>
          <div class="field">
            <label>κ scan lower bound</label>
            <input type="number" id="a3-n-kmin" value="0.0" step="0.001">
          </div>
          <div class="field">
            <label>κ scan upper bound</label>
            <input type="number" id="a3-n-kmax" value="0.01" step="0.001">
          </div>
          <div class="field">
            <label>Number of scan points</label>
            <input type="number" id="a3-n-steps" value="41" step="1">
          </div>
        </div>
        <button onclick="calcA3Numeric()">Numerical Scan for Critical Value</button>

        <div id="a3-result" class="result" style="display:none;"></div>
      </div>
</div>

<!-- ====================== A.4 ====================== -->
<div class="panel" id="a4">
  <div class="card">
    <h2>A.4 Module Compatibility Matrix and Shapley Values</h2>
    <p>Builds the compatibility matrix K from thermodynamic footprint overlap, information capacity ratio, and coupling strength, and computes the Shapley value of each module.</p>

    <div class="formula">
η_ij = 1 − |φ_i·φ_j| / (|φ_i||φ_j|)              (thermodynamic compatibility factor)
ξ_ij  = min(1,  C_ij/R_ij)                        (information compatibility factor)
ζ_ij = tanh(κ_ij/κ_c)                             (emergence factor)

K_ij = η_ij · ξ_ij · ζ_ij

Characteristic function:   v(S) = Σ_{i∈S} K_ii + Σ_{i<j, i,j∈S} K_ij

Shapley value:   φ_i(v) = Σ_{S⊆M\{i}} [|S|!(N−|S|−1)!/N!] · [v(S ∪ {i}) − v(S)]
    </div>
```

```
    <h3>Global Parameters</h3>
    <div class="input-grid">
      <div class="field">
        <label>Number of modules N (2–12)</label>
        <input type="number" id="a4-N" value="3" min="2" max="12" step="1" onchange="buildA4Inputs()">
      </div>
      <div class="field">
        <label>Critical coupling strength κ_c</label>
        <input type="number" id="a4-kc" value="0.01" step="0.001">
      </div>
      <div class="field">
        <label>Information capacity ratio ξ = C/R</label>
        <input type="number" id="a4-xi" value="0.8" step="0.05" min="0" max="1">
      </div>
    </div>

    <h3>Module Parameters</h3>
    <p class="note">Footprint vector has three components: Δμ_ATP (normalized), ΔpH (normalized), ΔΨ (normalized). Coupling strength κ_i is the coupling strength of that module to the others.</p>
    <div id="a4-modules"></div>

    <button onclick="calcA4()">Compute Compatibility Matrix and Shapley Values</button>
    <button class="secondary" onclick="buildA4Inputs()">Reset Module Parameters</button>

    <div id="a4-result" class="result" style="display:none;"></div>
  </div>
</div>

<!-- ====================== A.5 ====================== -->
<div class="panel" id="a5">
  <div class="card">
    <h2>A.5 Numerical Parameters</h2>
    <p>Parameter values used in all numerical simulations in this paper, consistent with physiological conditions reported in the literature.</p>

    <table>
      <thead>
        <tr><th>Parameter</th><th>Symbol</th><th>Value</th><th>Unit</th></tr>
      </thead>
      <tbody>
        <tr><td>Gas constant</td><td>R</td><td>8.314</td><td>J mol⁻¹ K⁻¹</td></tr>
        <tr><td>Absolute temperature</td><td>T</td><td>310</td><td>K</td></tr>
        <tr><td>Boltzmann constant</td><td>k_B</td><td>1.381×10⁻²³</td><td>J K⁻¹</td></tr>
```

<tr><td>ATP concentration</td><td>[ATP]</td><td>1–3</td><td>mM</td></tr>
<tr><td>ADP concentration</td><td>[ADP]</td><td>0.1–0.5</td><td>mM</td></tr>
<tr><td>Inorganic phosphate concentration</td><td>[P_i]</td><td>1–10</td><td>mM</td></tr>
<tr><td>ATP hydrolysis standard free energy</td><td>ΔG°'</td><td>−30.5</td><td>kJ mol⁻¹</td></tr>
<tr><td>Oscillator natural frequency</td><td>ω₀</td><td>0.1</td><td>min⁻¹</td></tr>
<tr><td>Oscillator damping rate</td><td>γ</td><td>0.05</td><td>min⁻¹</td></tr>
</tbody>
</table>

These values are taken from typical physiological conditions reported in Tu (2026), Li et al. (2025), and Holló et al. (2025).

<!-- ========== A.6 ========== -->

<h2>A.6 Computational Implementation</h2>
All numerical simulations run in real time in JavaScript within this page. The core algorithms are described below.

<h3>Numerical Integration</h3>
Composite Simpson's rule with 1000 subintervals:

$\int_a^b f(x)\,dx \approx (h/3)\cdot[f(x_0) + 4\Sigma f(x_{odd}) + 2\Sigma f(x_{even}) + f(x_n)]$
where $h = (b-a)/n$

<h3>Eigenvalue Analysis</h3>
For the two-module linearly coupled system, the Jacobian is a 4×4 real matrix. The critical coupling strength is determined by solving the characteristic polynomial:

$\det(J - \lambda I) = 0$

In-phase mode: $\lambda^2 + \gamma\lambda + \omega_0^2 = 0$
Anti-phase mode: $\lambda^2 + \gamma\lambda + (\omega_0^2 + 2\kappa) = 0$

Hopf bifurcation: occurs when a pair of complex conjugate eigenvalues of the anti-phase mode crosses the imaginary axis

```
      <h3>Shapley Value Computation</h3>
      <p>For N ≤ 12, all 2^N subsets are enumerated exactly. For N > 12, a Monte Carlo approximation with 100,000 random permutations is used.</p>
      <div class="formula">
Exact algorithm complexity:  O(N · 2^(N−1))
Monte Carlo complexity:  O(N · samples)
      </div>

      <h3>Matrix Heatmap Rendering</h3>
      <p>The compatibility index K_ij ∈ [0,1] maps to HSL hue:</p>
      <div class="formula">
hue = K_ij × 120°
color = hsl(hue, 70%, 85%)
      </div>

      <button onclick="runFullDemo()">Run Full Demonstration</button>
      <div id="a6-result" class="result" style="display:none;"></div>
    </div>
</div>

</div>

<script>
/* ============================================================
   Utility Functions
   ============================================================ */

// Composite Simpson numerical integration
function integrate(f, a, b, n) {
  n = n || 1000;
  if (n % 2 === 1) n++;
  const h = (b - a) / n;
  let sum = f(a) + f(b);
  for (let i = 1; i < n; i++) {
    sum += (i % 2 === 0 ? 2 : 4) * f(a + i * h);
  }
  return sum * h / 3;
}

// Factorial
function factorial(n) {
  let r = 1;
  for (let i = 2; i <= n; i++) r *= i;
  return r;
```

```
}

// Number formatting
function fmt(x, digits) {
  if (digits === undefined) digits = 4;
  if (!isFinite(x)) return String(x);
  if (x === 0) return '0';
  const abs = Math.abs(x);
  if (abs < 1e-4 || abs >= 1e6) return x.toExponential(3);
  return x.toFixed(digits);
}

/* ============================================================
   Tab Switching
   ============================================================ */
document.querySelectorAll('.tab').forEach(tab => {
  tab.addEventListener('click', () => {
    document.querySelectorAll('.tab').forEach(t => t.classList.remove('active'));
    document.querySelectorAll('.panel').forEach(p => p.classList.remove('active'));
    tab.classList.add('active');
    document.getElementById(tab.dataset.target).classList.add('active');
  });
});

/* ============================================================
   A.1 Thermodynamic Footprint
   ============================================================ */
function calcA1() {
  const R = 8.314;
  const T = parseFloat(document.getElementById('a1-T').value);
  const dG0kJ = parseFloat(document.getElementById('a1-dG0').value);
  const dG0 = dG0kJ * 1000;

  const atpB = parseFloat(document.getElementById('a1-atp-b').value) * 1e-3;
  const adpB = parseFloat(document.getElementById('a1-adp-b').value) * 1e-3;
  const piB  = parseFloat(document.getElementById('a1-pi-b').value)  * 1e-3;
  const atpA = parseFloat(document.getElementById('a1-atp-a').value) * 1e-3;
  const adpA = parseFloat(document.getElementById('a1-adp-a').value) * 1e-3;
  const piA  = parseFloat(document.getElementById('a1-pi-a').value)  * 1e-3;

  if ([T, dG0kJ, atpB, adpB, piB, atpA, adpA, piA].some(v => !isFinite(v))) {
    showResult('a1-result', 'Invalid input. Please check all fields.', true);
    return;
  }
```

```
  if (atpB <= 0 || adpB <= 0 || piB <= 0 || atpA <= 0 || adpA <= 0 || piA <= 0) {
    showResult('a1-result', 'Concentrations must be positive.', true);
    return;
  }

  const muB = R * T * Math.log(atpB * piB / adpB);
  const muA = R * T * Math.log(atpA * piA / adpA);
  const dMu = muA - muB;

  const dGpB = dG0 + R * T * Math.log(adpB * piB / atpB);
  const dGpA = dG0 + R * T * Math.log(adpA * piA / atpA);
  const dDGp = dGpA - dGpB;

  // Normalized footprint components (for A.4)
  const footprintATP = dMu / 1000;
  const footprintPi = (piA - piB) / (piB || 1);

  let out = '';
  out += '=== Phosphorylation potential μ_ATP = RT·ln([ATP][P_i]/[ADP]) ===\n';
  out += `  Before operation:  μ_ATP = ${fmt(muB)} J/mol = ${fmt(muB/1000)} kJ/mol\n`;
  out += `  After operation:   μ_ATP = ${fmt(muA)} J/mol = ${fmt(muA/1000)} kJ/mol\n`;
  out += `  Shift:             Δμ_ATP = <span class="hl">${fmt(dMu)} J/mol = ${fmt(dMu/1000)}
kJ/mol</span>\n\n`;

  out += '=== ATP hydrolysis free energy ΔG = ΔG°\' + RT·ln([ADP][P_i]/[ATP]) ===\n';
  out += `  ΔG°' = ${fmt(dG0kJ)} kJ/mol\n`;
  out += `  Before operation:  ΔG = ${fmt(dGpB/1000)} kJ/mol\n`;
  out += `  After operation:   ΔG = ${fmt(dGpA/1000)} kJ/mol\n`;
  out += `  Change:            Δ(ΔG) = <span class="hl">${fmt(dDGp/1000)} kJ/mol</span>\n\n`;

  out += '=== Normalized footprint components for A.4 ===\n';
  out += `  Δμ_ATP (kJ/mol)    = ${fmt(footprintATP)}\n`;
  out += `  ΔpH (relative)     = ${fmt((piA - piB) / piB)}\n`;
  out += `  ΔΨ (not computed, default 0) = 0\n\n`;

  out += '=== Criteria ===\n';
  if (Math.abs(dMu) > 5000) {
    out += `  <span class="warn">Large footprint</span>: |Δμ_ATP| > 5 kJ/mol, high probability of
thermodynamic coupling with other modules.\n`;
  } else if (Math.abs(dMu) > 1000) {
    out += `  <span style="color:#856404">Medium footprint</span>: 1 < |Δμ_ATP| < 5 kJ/mol, coupling
management required.\n`;
  } else {
    out += `  <span class="ok">Low footprint</span>: |Δμ_ATP| < 1 kJ/mol, preferred module type.\n`;
```

```
    }

    showResult('a1-result', out);
}

function resetA1() {
    document.getElementById('a1-T').value = 310;
    document.getElementById('a1-dG0').value = -30.5;
    document.getElementById('a1-atp-b').value = 3.0;
    document.getElementById('a1-adp-b').value = 0.5;
    document.getElementById('a1-pi-b').value = 5.0;
    document.getElementById('a1-atp-a').value = 1.5;
    document.getElementById('a1-adp-a').value = 2.0;
    document.getElementById('a1-pi-a').value = 6.5;
    document.getElementById('a1-result').style.display = 'none';
}

/* ============================================================
     A.2 Information Capacity
     ============================================================ */

// Standard normal density
function normalPDF(x, mu, sigma) {
    return Math.exp(-0.5 * Math.pow((x - mu) / sigma, 2)) / (sigma * Math.sqrt(2 * Math.PI));
}

function calcA2Gauss() {
    const gs = parseFloat(document.getElementById('a2-gs').value);
    const gn = parseFloat(document.getElementById('a2-gn').value);
    if (!isFinite(gs) || !isFinite(gn) || gn <= 0 || gs < 0) {
        showResult('a2-result', 'Invalid input: variances must be non-negative, noise variance must be positive.',
true);
        return;
    }
    const C = 0.5 * Math.log2(1 + gs / gn);
    let out = '';
    out += '=== Gaussian Channel Information Capacity ===\n';
    out += `    σ_s² = ${gs}\n`;
    out += `    σ_n² = ${gn}\n`;
    out += `    SNR = ${fmt(gs/gn)}\n\n`;
    out += `    C = ½·log₂(1 + σ_s²/σ_n²)\n`;
    out += `      = ½·log₂(1 + ${fmt(gs/gn)})\n`;
    out += `      = <span class="hl">${fmt(C)} bits per unit time</span>\n\n`;
    out += `    Physical interpretation: this channel can reliably transmit at most ${fmt(C)} bits of information
```

```
per unit time.\n`;
  out += `   If the required information transfer rate R_ij > ${fmt(C)} bits per unit time, coupling fails.\n`;
  showResult('a2-result', out);
}

function calcA2Poisson() {
  const l0 = parseFloat(document.getElementById('a2-p-l0').value);
  const l1 = parseFloat(document.getElementById('a2-p-l1').value);
  const mu = parseFloat(document.getElementById('a2-p-mu').value);
  const sig = parseFloat(document.getElementById('a2-p-sig').value);

  if (!isFinite(l0) || !isFinite(l1) || !isFinite(mu) || !isFinite(sig) || sig <= 0) {
    showResult('a2-result', 'Invalid input.', true);
    return;
  }

  const lambda = x => l0 + l1 * x;

  // Integrand: p(x)·ln(λ'(x)²/λ(x))
  // λ'(x) = l1, so = p(x)·ln(l1²/(l0+l1·x))
  const integrand = x => {
    const lam = lambda(x);
    if (lam <= 0) return 0;      // safeguard
    return normalPDF(x, mu, sig) * Math.log((l1 * l1) / lam);
  };

  const a = mu - 6 * sig;
  const b = mu + 6 * sig;
  const integral = integrate(integrand, a, b, 2000);

  const C = 0.5 * Math.log2(2 * Math.PI * Math.E) + 0.5 * integral;

  let out = '';
  out += '=== Poisson Signaling Model Information Capacity ===\n';
  out += `   λ(x) = λ₀ + λ₁·x = ${l0} + ${l1}·x\n`;
  out += `   Signal distribution p(x) ~ N(${mu}, ${sig}²)\n\n`;
  out += `   C = ½·log₂(2πe) + ½· ∫  p(x)·ln(λ'(x)²/λ(x)) dx\n`;
  out += `     = ½·log₂(2πe) + ½· ∫  p(x)·ln(λ₁²/(λ₀+λ₁x)) dx\n\n`;
  out += `   First term:  ½·log₂(2πe) = ${fmt(0.5 * Math.log2(2 * Math.PI * Math.E))} bits\n`;
  out += `   Second term: ½· ∫  ... dx = ${fmt(0.5 * integral)} bits\n`;
  out += `   ──────────────────────────────────────────\n`;
  out += `   C = <span class="hl">${fmt(C)} bits per unit time</span>\n\n`;

  if (C < 0) {
```

```
    out += '  <span class="warn">Warning: negative capacity. With these parameters, noise dominates and the channel cannot reliably transmit information.</span>\n';
  } else if (C < 1) {
    out += '  <span style="color:#856404">Low-capacity channel, only suitable for slow signaling.</span>\n';
  } else {
    out += '  <span class="ok">Sufficient capacity, can support moderate-rate signaling.</span>\n';
  }

  showResult('a2-result', out);
}

function calcA2Binom() {
  const n = parseFloat(document.getElementById('a2-b-n').value);
  const x0 = parseFloat(document.getElementById('a2-b-x0').value);
  const s = parseFloat(document.getElementById('a2-b-s').value);
  const mu = parseFloat(document.getElementById('a2-b-mu').value);
  const sig = parseFloat(document.getElementById('a2-b-sig').value);

  if (!isFinite(n) || !isFinite(x0) || !isFinite(s) || !isFinite(mu) || !isFinite(sig) || s <= 0 || sig <= 0 || n <= 0) {
    showResult('a2-result', 'Invalid input.', true);
    return;
  }

  // Activation probability
  const pAct = x => 1 / (1 + Math.exp(-(x - x0) / s));

  // Integrand: p(x)·ln(n·p(x)(1−p(x))/s²)
  const integrand = x => {
    const p = pAct(x);
    if (p <= 0 || p >= 1) return 0;
    const val = n * p * (1 - p) / (s * s);
    if (val <= 0) return 0;
    return normalPDF(x, mu, sig) * Math.log(val);
  };

  const a = mu - 6 * sig;
  const b = mu + 6 * sig;
  const integral = integrate(integrand, a, b, 2000);

  const C = 0.5 * Math.log2(2 * Math.PI * Math.E) + 0.5 * integral;

  let out = '';
  out += '=== Binomial Receptor Model Information Capacity ===\n';
```

```
  out += `  Number of receptors n = ${n}\n`;
  out += `  Activation function p(x) = 1/(1+exp(−(x−${x0})/${s}))  \n`;
  out += `  Signal distribution p(x) ~ N(${mu}, ${sig}²)\n\n`;
  out += `  C = ½·log₂(2πe) + ½· ∫  p(x)·ln(n·p'(x)²/(p(x)(1−p(x)))) dx\n`;
  out += `  where p'(x) = p(x)(1−p(x))/s\n\n`;
  out += `  First term:  ½·log₂(2πe) = ${fmt(0.5 * Math.log2(2 * Math.PI * Math.E))} bits\n`;
  out += `  Second term: ½· ∫  ... dx = ${fmt(0.5 * integral)} bits\n`;
  out += `  ──────────────────────────────────────────────\n`;
  out += `  C = <span class="hl">${fmt(C)} bits per unit time</span>\n\n`;

  if (C < 0) {
    out += '  <span class="warn">Warning: negative capacity. With these parameters, the receptors cannot reliably distinguish signals.</span>\n';
  } else {
    out += '  <span class="ok">This receptor configuration can reliably distinguish signal changes.</span>\n';
  }

  showResult('a2-result', out);
}

/* ============================================================
   A.3 Critical Coupling Strength
   ============================================================ */
function calcA3Analytic() {
  const w0 = parseFloat(document.getElementById('a3-w0').value);
  const gamma = parseFloat(document.getElementById('a3-gamma').value);

  if (!isFinite(w0) || !isFinite(gamma) || w0 <= 0 || gamma <= 0) {
    showResult('a3-result', 'Invalid input: ω₀ and γ must be positive.', true);
    return;
  }

  const kc = gamma * w0 / 2;

  let out = '';
  out += '=== Two-Module Coupled Oscillator Analytical Critical Value ===\n\n';
  out += `  Natural frequency ω₀ = ${w0} min⁻¹\n`;
  out += `  Damping rate      γ  = ${gamma} min⁻¹\n\n`;
  out += `  κ_c = γ·ω₀ / 2\n`;
  out += `      = ${gamma} × ${w0} / 2\n`;
  out += `      = <span class="hl">${fmt(kc)} min⁻¹</span>\n\n`;

  out += '=== Predicted Dynamical Behavior ===\n';
```

```
    out += `   κ < ${fmt(kc)}   : modules oscillate independently, no synchronization\n`;
    out += `   κ ≈ ${fmt(kc)}   : critical region, long-range correlations and amplified response\n`;
    out += `   κ > ${fmt(kc)}   : integrated state, synchronized oscillation\n\n`;

    out += '=== Derivation Notes ===\n';
    out += '  In-phase mode:       ü + γü + ω₀²u = 0     (unaffected by coupling)\n';
  out += '  Anti-phase mode:  v̈ + γv̇ + (ω₀² + 2κ)v = 0\n';
    out += '  When the effective frequency ω_eff = √(ω₀² + 2κ) of the anti-phase mode\n';
    out += '  approaches the in-phase mode frequency, the two modes resonantly couple\n';
    out += '  and the system enters the synchronized state.\n';
    out += '  The critical condition corresponds to the effective damping of the anti-phase mode going to zero.\n';

    showResult('a3-result', out);
}

function calcA3Numeric() {
    const w0 = parseFloat(document.getElementById('a3-n-w0').value);
    const gamma = parseFloat(document.getElementById('a3-n-gamma').value);
    const kmin = parseFloat(document.getElementById('a3-n-kmin').value);
    const kmax = parseFloat(document.getElementById('a3-n-kmax').value);
    const steps = parseInt(document.getElementById('a3-n-steps').value);

    if (!isFinite(w0) || !isFinite(gamma) || !isFinite(kmin) || !isFinite(kmax) || !isFinite(steps) || steps < 2) {
        showResult('a3-result', 'Invalid input.', true);
        return;
    }
    if (kmin < 0 || kmax <= kmin) {
        showResult('a3-result', 'Invalid scan range: requires 0 ≤ kmin < kmax.', true);
        return;
    }

    // For the two-module linearly coupled system:
    //     ẍ₁ + γẋ₁ + ω₀²x₁ = κ(x₂ − x₁)
    //     ẍ₂ + γẋ₂ + ω₀²x₂ = κ(x₁ − x₂)
    // Let u = x₁+x₂, v = x₁−x₂:
    //     ü + γü + ω₀²u = 0
  //   v̈ + γv̇ + (ω₀² + 2κ)v = 0
    // Eigenvalues of anti-phase mode: λ = [−γ ± √(γ² − 4(ω₀²+2κ))]/2
    // Real part = −γ/2 < 0 always. Hopf bifurcation does not occur here.
    // But frequency matching condition:  √(ω₀²+2κ) = ω₀   →   κ = 0
    // The actual synchronization critical condition is when coupling strength
    // is sufficient to overcome the damping-induced frequency difference.
    // We use the standard result κ_c = γ·ω₀/2 as a reference and numerically locate
```

```
// the point where the anti-phase mode quality factor Q_inv = √(ω₀²+2κ)/γ reaches 1/2.

const kcAnalytic = gamma * w0 / 2;

// Compute anti-phase mode eigenvalues for each κ
const rows = [];
for (let i = 0; i < steps; i++) {
  const k = kmin + (kmax - kmin) * i / (steps - 1);
  const omegaInv2 = w0 * w0 + 2 * k;
  const omegaInv = Math.sqrt(omegaInv2);
  const disc = gamma * gamma - 4 * omegaInv2;

  let lambdaRe, lambdaIm;
  if (disc >= 0) {
    // Overdamped
    const r1 = (-gamma + Math.sqrt(disc)) / 2;
    const r2 = (-gamma - Math.sqrt(disc)) / 2;
    lambdaRe = Math.max(r1, r2);
    lambdaIm = 0;
  } else {
    // Underdamped
    lambdaRe = -gamma / 2;
    lambdaIm = Math.sqrt(-disc) / 2;
  }

  const Q = omegaInv / gamma;    // quality factor

  rows.push({ k, omegaInv, lambdaRe, lambdaIm, Q });
}

// Find κ where Q is closest to 0.5
let bestIdx = 0;
let bestDiff = Infinity;
rows.forEach((r, i) => {
  const d = Math.abs(r.Q - 0.5);
  if (d < bestDiff) { bestDiff = d; bestIdx = i; }
});

let out = '';
out += '=== Numerical Scan: Two-Module Coupled Oscillator ===\n\n';
out += `  Analytical critical value κ_c = γ·ω₀/2 = ${fmt(kcAnalytic)} min⁻¹\n`;
out += `  Scan range: [${kmin}, ${kmax}], ${steps} sample points\n\n`;

out += '   κ          ω_inv        Re(λ)          Im(λ)          Q\n';
```

```
  out += '  ──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────\n';
  rows.forEach((r, i) => {
    const marker = (i === bestIdx) ? '  ← Q≈0.5' : '';
    out += `  ${fmt(r.k, 5).padEnd(9)} ${fmt(r.omegaInv, 5).padEnd(10)} ${fmt(r.lambdaRe, 5).padEnd(12)} ${fmt(r.lambdaIm, 5).padEnd(12)} ${fmt(r.Q, 4)}${marker}\n`;
  });

  out += '\n=== Result ===\n';
  out += `  Best critical point found by numerical scan: κ ≈ <span class="hl">${fmt(rows[bestIdx].k)}</span>\n`;
  out += `  Corresponding anti-phase mode frequency: ω_inv = ${fmt(rows[bestIdx].omegaInv)}\n`;
  out += `  Quality factor: Q = ${fmt(rows[bestIdx].Q)}\n\n`;

  out += '=== Dynamical Regimes ===\n';
  const kcVal = kcAnalytic;
  if (kmax < kcVal) {
    out += `  Entire scan range lies in κ < κ_c = ${fmt(kcVal)}, system is in the independent oscillation regime.\n`;
  } else if (kmin > kcVal) {
    out += `  Entire scan range lies in κ > κ_c = ${fmt(kcVal)}, system is in the integrated state.\n`;
  } else {
    out += `  κ < ${fmt(kcVal)}   : independent oscillation\n`;
    out += `  κ ≈ ${fmt(kcVal)}   : critical region, long-range correlations and amplified response\n`;
    out += `  κ > ${fmt(kcVal)}   : integrated state, synchronized oscillation\n`;
  }

  showResult('a3-result', out);
}

/* ============================================================
   A.4 Compatibility Matrix and Shapley Values
   ============================================================ */

const A4_DEFAULTS = [
  { f: [1.0, 0.0, 0.0], k: 0.020 },
  { f: [0.8, 0.2, 0.0], k: 0.015 },
  { f: [0.0, 1.0, 0.0], k: 0.010 },
  { f: [0.3, 0.3, 0.4], k: 0.008 },
  { f: [0.0, 0.0, 1.0], k: 0.012 },
  { f: [0.5, 0.5, 0.0], k: 0.018 },
  { f: [0.2, 0.6, 0.2], k: 0.009 },
  { f: [0.7, 0.1, 0.2], k: 0.011 },
  { f: [0.1, 0.2, 0.7], k: 0.014 },
```

```
  { f: [0.4, 0.4, 0.2], k: 0.016 },
  { f: [0.6, 0.3, 0.1], k: 0.013 },
  { f: [0.0, 0.5, 0.5], k: 0.007 }
];

function buildA4Inputs() {
  const N = parseInt(document.getElementById('a4-N').value);
  if (!isFinite(N) || N < 2 || N > 12) return;

  const container = document.getElementById('a4-modules');
  container.innerHTML = '';

  for (let i = 0; i < N; i++) {
    const d = A4_DEFAULTS[i] || { f: [0.5, 0.5, 0.0], k: 0.01 };
    const row = document.createElement('div');
    row.className = 'module-row';
    row.innerHTML = `
      <div class="label">M${i+1}</div>
      <input type="number" step="0.1" id="a4-f${i}-0" value="${d.f[0]}" title="Δμ_ATP component">
      <input type="number" step="0.1" id="a4-f${i}-1" value="${d.f[1]}" title="ΔpH component">
      <input type="number" step="0.1" id="a4-f${i}-2" value="${d.f[2]}" title="ΔΨ component">
      <input type="number" step="0.001" id="a4-k${i}" value="${d.k}" title="Coupling strength κ_i">
    `;
    container.appendChild(row);
  }
}

function readA4Inputs() {
  const N = parseInt(document.getElementById('a4-N').value);
  const kc = parseFloat(document.getElementById('a4-kc').value);
  const xi = parseFloat(document.getElementById('a4-xi').value);

  if (!isFinite(N) || N < 2 || N > 12) {
    return { error: 'Number of modules N must be between 2 and 12.' };
  }
  if (!isFinite(kc) || kc <= 0) {
    return { error: 'Critical coupling strength κ_c must be positive.' };
  }
  if (!isFinite(xi) || xi < 0 || xi > 1) {
    return { error: 'Information capacity ratio ξ must be in [0, 1].' };
  }

  const modules = [];
  for (let i = 0; i < N; i++) {
```

```
        const f = [
            parseFloat(document.getElementById(`a4-f${i}-0`).value),
            parseFloat(document.getElementById(`a4-f${i}-1`).value),
            parseFloat(document.getElementById(`a4-f${i}-2`).value)
        ];
        const k = parseFloat(document.getElementById(`a4-k${i}`).value);
        if (f.some(v => !isFinite(v)) || !isFinite(k)) {
            return { error: `Invalid parameters for module M${i+1}.` };
        }
        modules.push({ f, k });
    }

    return { N, kc, xi, modules };
}

function calcA4() {
    const inp = readA4Inputs();
    if (inp.error) {
        showResult('a4-result', inp.error, true);
        return;
    }
    const { N, kc, xi, modules } = inp;

    // Vector norms
    const norms = modules.map(m => Math.sqrt(m.f.reduce((s, v) => s + v * v, 0)));

    // Compatibility matrix
    const K = Array.from({ length: N }, () => new Array(N).fill(0));
    for (let i = 0; i < N; i++) {
        for (let j = 0; j < N; j++) {
            if (i === j) { K[i][j] = 1; continue; }
            const dot = modules[i].f.reduce((s, v, k) => s + v * modules[j].f[k], 0);
            const ni = norms[i] || 1;
            const nj = norms[j] || 1;
            const cosSim = dot / (ni * nj);
            const eta = 1 - Math.abs(cosSim);
            const kij = Math.min(modules[i].k, modules[j].k);
            const zeta = Math.tanh(kij / kc);
            K[i][j] = eta * xi * zeta;
        }
    }

    // Characteristic function v(S)
    const size = 1 << N;
```

```
const v = new Float64Array(size);
for (let S = 0; S < size; S++) {
  let val = 0;
  for (let i = 0; i < N; i++) {
    if (S & (1 << i)) {
      val += K[i][i];
      for (let j = i + 1; j < N; j++) {
        if (S & (1 << j)) val += K[i][j];
      }
    }
  }
  v[S] = val;
}

// Shapley values (exact enumeration)
let shapley;
if (N <= 12) {
  shapley = new Float64Array(N);
  const fact = [1];
  for (let i = 1; i <= N; i++) fact[i] = fact[i-1] * i;
  const Nfact = fact[N];

  for (let i = 0; i < N; i++) {
    let sum = 0;
    const others = [];
    for (let j = 0; j < N; j++) if (j !== i) others.push(j);
    const M = others.length;
    const totalMasks = 1 << M;
    for (let mask = 0; mask < totalMasks; mask++) {
      let S = 0, s = 0;
      for (let k = 0; k < M; k++) {
        if (mask & (1 << k)) { S |= (1 << others[k]); s++; }
      }
      const weight = fact[s] * fact[N - s - 1] / Nfact;
      sum += weight * (v[S | (1 << i)] - v[S]);
    }
    shapley[i] = sum;
  }
} else {
  // Monte Carlo (for N > 12)
  const samples = 100000;
  shapley = new Float64Array(N);
  for (let s = 0; s < samples; s++) {
    const perm = Array.from({ length: N }, (_, i) => i);
```

```
        for (let i = N - 1; i > 0; i--) {
            const j = Math.floor(Math.random() * (i + 1));
            [perm[i], perm[j]] = [perm[j], perm[i]];
        }
        let S = 0, vS = 0;
        for (let k = 0; k < N; k++) {
            const i = perm[k];
            const newS = S | (1 << i);
            const delta = v[newS] - vS;
            shapley[i] += delta;
            S = newS;
            vS = v[newS];
        }
    }
    for (let i = 0; i < N; i++) shapley[i] /= samples;
}

// Output
let out = '';
out += '=== Compatibility Matrix K ===\n\n';

// Header
out += '        ';
for (let j = 0; j < N; j++) out += `M${j+1}`.padEnd(9);
out += '\n';
out += '  ──────';
for (let j = 0; j < N; j++) out += '─────────';
out += '\n';

for (let i = 0; i < N; i++) {
    out += `  M${i+1}`.padEnd(7);
    for (let j = 0; j < N; j++) {
        out += fmt(K[i][j], 3).padEnd(9);
    }
    out += '\n';
}

out += '\n=== Matrix Structure Analysis ===\n';
let offDiagSum = 0, offDiagCount = 0;
let maxOff = 0, minOff = 1;
for (let i = 0; i < N; i++) {
    for (let j = 0; j < N; j++) {
        if (i !== j) {
            offDiagSum += K[i][j];
```

```
            offDiagCount++;
            if (K[i][j] > maxOff) maxOff = K[i][j];
            if (K[i][j] < minOff) minOff = K[i][j];
          }
        }
      }
      const avgOff = offDiagSum / offDiagCount;
      out += `  Mean off-diagonal element: ${fmt(avgOff)}\n`;
      out += `  Max off-diagonal element:  ${fmt(maxOff)}\n`;
      out += `  Min off-diagonal element:  ${fmt(minOff)}\n\n`;

      if (avgOff < 0.1) {
        out += '  <span class="ok">Matrix is approximately diagonal</span>: modules are nearly independent, system behaves as assembled.\n';
        out += '  Recommendation: moderately increase coupling strength to enter the integrated state.\n';
      } else if (avgOff < 0.5) {
        out += '  <span style="color:#856404">Moderate coupling</span>: system may be near the critical region.\n';
        out += '  Recommendation: fine-tune κ toward the critical value to trigger emergence.\n';
      } else {
        out += '  <span class="warn">Strong coupling</span>: extensive inter-module coupling, system may enter the integrated state but faces stability risk.\n';
        out += '  Recommendation: check for unintended coupling pathways, consider reducing some κ values.\n';
      }

      out += '\n=== Shapley Values (Module Marginal Contributions) ===\n\n';
      out += '  Module      Shapley value      Share      Assessment\n';
      out += '  ──────────────────────────────────────────────────────────────────────\n';
      const sumShap = shapley.reduce((a, b) => a + b, 0);
      const ranked = Array.from(shapley).map((v, i) => ({ i, v })).sort((a, b) => b.v - a.v);
      ranked.forEach((item, rank) => {
        const pct = sumShap > 0 ? (item.v / sumShap * 100) : 0;
        let tag;
        if (rank === 0) tag = '★ Highest contribution';
        else if (rank === ranked.length - 1) tag = '▼ Lowest contribution';
        else tag = '';
        out += `  M${String(item.i+1).padEnd(7)}  ${fmt(item.v, 5).padEnd(13)}  ${fmt(pct, 1).padEnd(8)}%  ${tag}\n`;
      });

      out += '\n=== Module Selection Recommendation ===\n';
      const lowest = ranked[ranked.length - 1];
```

```
  const highest = ranked[0];
  if (lowest.v < 0) {
    out += `  <span class="warn">Shapley value of M${lowest.i+1} is negative</span>, consider removing or redesigning it.\n`;
  } else if (lowest.v < 0.05 * highest.v) {
    out += `  Marginal contribution of M${lowest.i+1} is far below that of M${highest.i+1}, consider replacing it with a smaller-footprint variant.\n`;
  } else {
    out += `  All modules have positive Shapley values; the module combination is overall reasonable.\n`;
  }
  out += `  Highest-contribution module: M${highest.i+1} (Shapley = ${fmt(highest.v, 5)})\n`;

  // Render heatmap matrix
  out += '\n=== Heatmap (Visual Matrix) ===\n';
  out += '<div class="matrix-container"><table><thead><tr><th></th>';
  for (let j = 0; j < N; j++) out += `<th>M${j+1}</th>`;
  out += '</tr></thead><tbody>';
  for (let i = 0; i < N; i++) {
    out += `<tr><th>M${i+1}</th>`;
    for (let j = 0; j < N; j++) {
      const val = K[i][j];
      const hue = val * 120;
      const bg = `hsl(${hue}, 70%, 85%)`;
      out += `<td style="background:${bg}; text-align:center; font-family:monospace;">${val.toFixed(3)}</td>`;
    }
    out += '</tr>';
  }
  out += '</tbody></table></div>';

  showResult('a4-result', out);
}

/* ============================================================
   A.6 Full Demonstration
   ============================================================ */
function runFullDemo() {
  let out = '';
  out += '=== Full Demonstration: From Thermodynamic Footprint to Emergence Threshold ===\n\n';

  // Step 1: Thermodynamic footprint
  const R = 8.314, T = 310, dG0 = -30500;
  const atpB = 3e-3, adpB = 0.5e-3, piB = 5e-3;
  const atpA = 1.5e-3, adpA = 2e-3, piA = 6.5e-3;
```

```
const muB = R * T * Math.log(atpB * piB / adpB);
const muA = R * T * Math.log(atpA * piA / adpA);
const dMu = muA - muB;

out += '[Step 1] Thermodynamic Footprint Calculation\n';
out += `    Module M1 (ATP-consuming): Δμ_ATP = ${fmt(dMu/1000)} kJ/mol\n`;
out += `    Footprint vector φ_1 = (${fmt(dMu/1000)}, ${fmt((piA-piB)/piB)}, 0)\n\n`;

// Step 2: Information capacity
const gs = 1.0, gn = 0.1;
const Cgauss = 0.5 * Math.log2(1 + gs / gn);
out += '[Step 2] Information Capacity Calculation\n';
out += `    Gaussian channel: C = ${fmt(Cgauss)} bits per unit time\n`;
out += `    Maximum information rate supported by channel 1: ${fmt(Cgauss)} bit/s\n\n`;

// Step 3: Critical coupling strength
const w0 = 0.1, gamma = 0.05;
const kc = gamma * w0 / 2;
out += '[Step 3] Critical Coupling Strength Condition\n';
out += `    ω₀ = ${w0} min⁻¹, γ = ${gamma} min⁻¹\n`;
out += `    κ_c = γ·ω₀/2 = ${fmt(kc)} min⁻¹\n\n`;

// Step 4: Compatibility matrix
const N = 3;
const kcGlobal = 0.01;
const xi = 0.8;
const modules = [
  { f: [dMu/1000, (piA-piB)/piB, 0], k: 0.020 },
  { f: [0.8, 0.2, 0.0], k: 0.015 },
  { f: [0.0, 1.0, 0.0], k: 0.010 }
];

const norms = modules.map(m => Math.sqrt(m.f.reduce((s, v) => s + v * v, 0)));
const K = Array.from({ length: N }, () => new Array(N).fill(0));
for (let i = 0; i < N; i++) {
  for (let j = 0; j < N; j++) {
    if (i === j) { K[i][j] = 1; continue; }
    const dot = modules[i].f.reduce((s, v, k) => s + v * modules[j].f[k], 0);
    const cosSim = dot / ((norms[i] || 1) * (norms[j] || 1));
    const eta = 1 - Math.abs(cosSim);
    const kij = Math.min(modules[i].k, modules[j].k);
    const zeta = Math.tanh(kij / kcGlobal);
    K[i][j] = eta * xi * zeta;
  }
```

```
}

out += '[Step 4] Compatibility Matrix K (N=3)\n';
out += '          M1          M2          M3\n';
for (let i = 0; i < N; i++) {
  out += `  M${i+1} `;
  for (let j = 0; j < N; j++) out += K[i][j].toFixed(3).padEnd(9);
  out += '\n';
}

// Step 5: Shapley values
const size = 1 << N;
const v = new Float64Array(size);
for (let S = 0; S < size; S++) {
  let val = 0;
  for (let i = 0; i < N; i++) {
    if (S & (1 << i)) {
      val += K[i][i];
      for (let j = i + 1; j < N; j++) if (S & (1 << j)) val += K[i][j];
    }
  }
  v[S] = val;
}
const fact = [1, 1, 2, 6];
const shap = new Float64Array(N);
for (let i = 0; i < N; i++) {
  let sum = 0;
  const others = [];
  for (let j = 0; j < N; j++) if (j !== i) others.push(j);
  const M = others.length;
  for (let mask = 0; mask < (1 << M); mask++) {
    let S = 0, s = 0;
    for (let k = 0; k < M; k++) {
      if (mask & (1 << k)) { S |= (1 << others[k]); s++; }
    }
    const weight = fact[s] * fact[N - s - 1] / fact[N];
    sum += weight * (v[S | (1 << i)] - v[S]);
  }
  shap[i] = sum;
}

out += '\n[Step 5] Shapley Values\n';
for (let i = 0; i < N; i++) {
  out += `  M${i+1}: ${fmt(shap[i], 5)}\n`;
```

```
    }

    // Step 6: Emergence criterion
    out += '\n[Step 6] Emergence Criticality Criterion\n';
    const avgK = (K[0][1] + K[0][2] + K[1][2]) / 3;
    out += `   Mean off-diagonal compatibility: ${fmt(avgK)}\n`;
    if (avgK < 0.1) {
      out += '   → System is in the "simple coexistence" state, has not reached the emergence threshold\n';
      out += '   → Recommendation: increase κ or reselect modules with more complementary footprints\n';
    } else if (avgK < 0.5) {
      out += '   → System is approaching the critical region\n';
      out += '   → Recommendation: finely scan κ near κ_c\n';
    } else {
      out += '   → System may have entered the integrated state\n';
      out += '   → Recommendation: monitor for coupled oscillations or global synchronization\n';
    }

    showResult('a6-result', out);
}

/* ============================================================
   Generic Result Rendering
   ============================================================ */
function showResult(id, text, isError) {
    const el = document.getElementById(id);
    el.style.display = 'block';
    if (isError) {
      el.innerHTML = `<span class="warn">${text}</span>`;
    } else {
      el.innerHTML = text;
    }
}

/* ============================================================
   Initialization
   ============================================================ */
document.addEventListener('DOMContentLoaded', () => {
    buildA4Inputs();
});
</script>

</body>
</html>
```

Interactive Computation Modules

A.1 Thermodynamic Footprint – Enter ATP, ADP, and Pi concentrations before and after module operation to compute phosphorylation potential shifts and ATP hydrolysis free energy changes. The tool classifies the footprint as low, medium, or large.
A.2 Information Capacity – Calculate channel capacity using Gaussian, Poisson, or binomial receptor models. Each model has its own input panel and returns capacity in bits per unit time with interpretation.
A.3 Critical Coupling Strength – Get the analytical critical coupling strength $\kappa_c = \gamma\ \omega_0/2$, or run a numerical scan to inspect eigenvalue behavior and quality factor across a range of κ values.
A.4 Compatibility Matrix & Shapley Values – Configure up to 12 modules with footprint vectors and coupling strengths. The tool builds the compatibility matrix, computes each module's Shapley value, and renders a color-coded heatmap.
A.5 Numerical Parameters – A static reference table of all physical constants and default values used throughout the paper.
A.6 Computational Implementation – Run a full demonstration that chains all calculations together, from thermodynamic footprints through to the emergence criterion.

Optimization Tip: All default values can be edited directly in the input fields. For A.4, the initial module parameters are stored in the 'A4_DEFAULTS' array near the bottom of the script; you can modify these to match your own experimental data or hypothetical scenarios.

References

Bak P, Sneppen K. 1993. Punctuated equilibrium and criticality in a simple model of evolution. Physical Review Letters, 71(24): 4083-4086. https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.71.4083

Barabási AL, Oltvai ZN. 2004. Network biology: understanding the cell's functional organization. Nature Reviews Genetics, 5(2): 101-113. https://www.nature.com/articles/nrg1272

Bialek W, Setayeshgar S. 2005. Physical limits to biochemical signaling. Proceedings of the National Academy of Sciences of USA, 102(29): 10040-10045. https://www.pnas.org/doi/10.1073/pnas.0504321102

Briat C, Gupta A, Khammash M. 2016. Antithetic integral feedback ensures robust perfect adaptation in noisy biomolecular networks. Cell Systems, 2(1): 15-26. https://www.sciencedirect.com/science/article/pii/S2405471216000053

Chu M, VanArsdale E, Muthusamy D, et al. 2025. Scrambling signal modularity in bottom-up assembled synthetic Pseudomonas consortia reveals robust information transfer. ACS Synthetic Biology, 14(12): 4744-4755. https://doi.org/10.1021/acssynbio.5c00486

Cover TM, Thomas JA. 2005. Elements of Information Theory. 2nd Edition. 1-748, John Wiley & Sons, Hoboken, New Jersey, USA. https://onlinelibrary.wiley.com/doi/book/10.1002/047174882X

Dill K, Bromberg S. 2010. Molecular Driving Forces: Statistical Thermodynamics in Biology, Chemistry, Physics, and Nanoscience. 2nd Edition. 1-756, Garland Science, New York, USA. https://www.taylorfrancis.com/books/mono/10.4324/9780203809075/molecular-driving-forces-ken-dill-sarina-bromberg

Elowitz MB, Leibler S. 2000. A synthetic oscillatory network of transcriptional regulators. Nature, 403(6767): 335-338. https://www.nature.com/articles/35002125

Ferrell JE Jr. 2002. Self-perpetuating states in signal transduction: positive feedback, double-negative feedback and bistability. Current Opinion in Cell Biology, 14(2): 140-148. https://www.sciencedirect.com/science/article/pii/S0955067402003149

Gillespie DT. 1977. Exact stochastic simulation of coupled chemical reactions. The Journal of Physical Chemistry, 81(25): 2340-2361. https://pubs.acs.org/doi/10.1021/j100540a008

Haken H. 1977. Synergetics: An Introduction Nonequilibrium Phase Transitions and Self-Organization in Physics, Chemistry and Biology. Springer, Berlin, Germany. https://link.springer.com/book/10.1007/978-3-642-96363-6

Haken H. 1983. Advanced Synergetics: Instability Hierarchies of Self-Organizing Systems and Devices. Springer, Berlin, Germany. https://link.springer.com/book/10.1007/978-3-642-45553-7

Haken H. 2004. Synergetics: Introduction and Advanced Topics. 1-758, Springer, Berlin, Germany. https://link.springer.com/book/10.1007/978-3-662-10184-1

Heinrich R, Schuster S. 1996. The Regulation of Cellular Systems. Chapman & Hall, New York, USA. https://link.springer.com/book/10.1007/978-1-4613-1161-4

Hintze A, Adami C. 2008. Evolution of complex modular biological networks. PLoS Computational Biology, 4(2): e23. https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.0040023

Holló G, Park JH, Evard R, Schaerli Y. 2026. Intracellularly coupled oscillators for synthetic biology. Nature Communications, 17: 1131. https://www.nature.com/articles/s41467-025-67893-0

Inoue M, Arai T, Imura JI, Kashima K, Aihara K. 2014. Robustness analysis of genetic circuits constructed by bottom-up strategy. IFAC Proceedings Volumes, 19: 1736-1741. https://keio.elsevierpure.com/en/publications/robustness-analysis-of-genetic-circuits-onstructed-by-bottom-up-s/

Kauffman SA. 1993. The Origins of Order: Self-Organization and Selection in Evolution. 1-709, Oxford University Press, Oxford, UK. https://global.oup.com/academic/product/the-origins-of-order-9780195079517

Khammash MH. 2016. An engineering viewpoint on biological robustness. BMC Biology, 14: 22. https://link.springer.com/article/10.1186/s12915-016-0241-x

Kim J, Winfree E. 2011. Synthetic in vitro transcriptional oscillators. Molecular Systems Biology, 7: MSB2010119. https://www.embopress.org/doi/10.1038/msb.2010.119

Komorowski M. 2025. Closed-form information capacity of canonical signaling models. Communications Physics, 8: 496. https://link.springer.com/article/10.1038/s42005-025-02425-2

Levine E, Hwa T. 2007. Stochastic fluctuations in metabolic pathways. Proceedings of the National Academy of Sciences of USA, 104(22): 9224-9229. https://www.pnas.org/doi/10.1073/pnas.0610987104

Li YY, Zhu J, Zhang ZY, et al. 2025. Transforming an ATP-dependent enzyme into a dissipative, self-assembling system. Nature Chemical Biology, 21(6): 883-893. https://www.nature.com/articles/s41589-024-01811-1

Lorenz DM, Jeng A, Deem MW. 2011. The emergence of modularity in biological systems. Physics of Life Reviews, 8(2): 129-160. https://www.sciencedirect.com/science/article/pii/S1571064511000170

Milo R, Shen-Orr S, Itzkovitz S, et al. 2002. Network motifs: simple building blocks of complex networks. Science, 298(5594): 824-827. https://www.science.org/doi/10.1126/science.298.5594.824

Nicolis G, Prigogine I. 1977. Self-Organization in Nonequilibrium Systems: From Dissipative Structures to Order through Fluctuations. John Wiley & Sons, New York, USA. https://onlinelibrary.wiley.com/doi/10.1002/bbpc.197800155

Noireaux V, Bar-Ziv R, Libchaber A. 2003. Principles of cell-free genetic circuit assembly. Proceedings of the

National Academy of Sciences of USA, 100(22): 12672-12677. https://www.pnas.org/doi/10.1073/pnas.2135496100

Noireaux V, Libchaber A. 2004. A vesicle bioreactor as a step toward an artificial cell assembly. Proceedings of the National Academy of Sciences of USA, 101(51): 17669-17674. https://www.pnas.org/doi/10.1073/pnas.0408236101

Prigogine I, Van Rysselberghe P. 1963. Introduction to thermodynamics of irreversible processes. Journal of Electrochemical Society, 110: 97C. https://iopscience.iop.org/article/10.1149/1.2425756

Reverte López M. 2025. Modular engineering of self-division in synthetic cells. Dissertation, LMU München, Fakultät für Biologie, Germany. https://edoc.ub.uni-muenchen.de/35723/

Rijal S, Standage-Beier K, Zhang R, et al. 2025. CRISPRi-linked multimodule negative feedback loops to address winner-take-all resource competition. ACS Synthetic Biology, 14(9): 3646-3654. https://doi.org/10.1021/acssynbio.5c00394

Rusch FR, Kinouchi O, Roque AC. 2025. Influence of topology on the critical behavior of hierarchical modular neuronal networks. Communications Physics, 8: 168. https://link.springer.com/article/10.1038/s42005-025-02074-5

Sauro HM. 2014. Systems Biology: Introduction to Pathway Modeling. Ambrosius Publishing, Seattle, USA. https://www.ambrosiuspublishing.com/systems-biology/

Shannon CE. 1948. A mathematical theory of communication. Bell System Technical Journal, 27(3): 379-423. https://ieeexplore.ieee.org/document/6773024

Solé R, De Domenico M. 2025. Bifurcations and phase transitions in the origins of life. Philosophical Transactions of the Royal Society B, 380(1936): 20240295. https://royalsocietypublishing.org/doi/10.1098/rstb.2024.0295

Sousa FL, Hordijk W, et al. 2015. Autocatalytic sets in E. coli metabolism. Journal of Systems Chemistry, 6(1): 4. https://link.springer.com/article/10.1186/s13322-015-0009-7

Strogatz SH. 2015. Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering. 2nd edition. CRC Press, Boca Raton, USA. https://www.taylorfrancis.com/books/mono/10.1201/9780429492563/nonlinear-dynamics-chaos-steven-strogatz

Tu YH. 2026. Nonequilibrium thermodynamics for biochemical networks: Energetic principles of cellular function. Annual Review of Biophysics, 55: 225-252. https://www.annualreviews.org/content/journals/10.1146/annurev-biophys-021424-112247

Wang SJ, Zhou CS. 2012. Hierarchical modular structure enhances the robustness of self-organized criticality in neural networks. New Journal of Physics, 14: 023005. https://iopscience.iop.org/article/10.1088/1367-2630/14/2/023005

Youssef A, Rijal S, Zhang R, et al. 2026. Resource competition and growth dilution modulate synthetic gene cascade dynamics. ACS Synthetic Biology, 15(1): 181-190. https://www.sciencedirect.com/org/science/article/abs/pii/S2161506325004371

Zhang WJ. 2012. Computational Ecology: Graphs, Networks and Agent-based Modeling. 1-384, World Scientific, Singapore. https://www.worldscientific.com/worldscibooks/10.1142/8118#t=aboutBook

Zhang WJ. 2013. Selforganizology: A science that deals with self-organization. Network Biology, 3(1): 1-14. http://www.iaees.org/publications/journals/nb/articles/2013-3(1)/1-Zhang-Abstract.asp

Zhang WJ. 2016a. Network robustness: Implication, formulization and exploitation. Network Biology, 6(4): 75-85. http://www.iaees.org/publications/journals/nb/articles/2016-6(4)/1-Zhang-Abstract.asp

Zhang WJ. 2016b. Selforganizology: The Science of Self-Organization. 1-408, World Scientific, Singapore.

https://www.worldscientific.com/worldscibooks/10.1142/9685#t=aboutBook

Zhang WJ. 2016c. Network informatics: A new science. Selforganizology, 3(2): 43-50. http://www.iaees.org/publications/journals/selforganizology/articles/2016-3(2)/1-Zhang-Abstract.asp

Zhang WJ. 2016e. Some methods for sensitivity analysis of systems / networks. Network Pharmacology, 1(3): 74-81. http://www.iaees.org/publications/journals/np/articles/2016-1(3)/1-Zhang-Abstract.asp

Zhang WJ. 2018. Fundamentals of Network Biology. 1-456, World Scientific Europe, London, UK. https://www.worldscientific.com/worldscibooks/10.1142/q0149#t=aboutBook

Zhang WJ. 2026a. Ecological thermodynamic fluctuation and phase transition: A self-organized criticality-based framework for ecosystem responses to global warming. Selforganizology,13(3-4): 66-103. http://www.iaees.org/publications/journals/selforganizology/articles/2026-13(3-4)/2-Zhang-Abstract.asp

Zhang WJ. 2026b. The Niche-Energy-Time triadic synergy hypothesis (NET Hypothesis): A unified explanatory framework for determinants of species diversity. Computational Ecology and Software, 16(3): 243-272. http://www.iaees.org/publications/journals/ces/articles/2026-16(3)/3-Zhang-Abstract.asp

Zhang WJ. 2026c. What triggers explosive radiations and mass extinctions? A unified critical transition framework. Proceedings of the International Academy of Ecology and Environmental Sciences, 16(3): 89-122. http://www.iaees.org/publications/journals/piaees/articles/2026-16(3)/2-Zhang-Abstract.asp

Zhang WJ. 2026d. probFunCal2: A web-based calculator for calculating probability distribution functions as binomial, Poisson, exponential, gamma, beta, uniform, Weibull, lognormal, Cauchy, geometric, and negative binomial distributions. Selforganizology, 13(3-4): 26-65. http://www.iaees.org/publications/journals/selforganizology/articles/2026-13(3-4)/1-Zhang-Abstract.asp

Zhao MM, Kim J, Jiao JY, et al. 2024. Construction of multilayered gene circuits using de-novo-designed synthetic transcriptional regulators in cell-free systems. Journal of Biological Engineering, 18: 64. https://link.springer.com/article/10.1186/s13036-024-00459-8