Molecular Space-Time: Emergent Prediction from Local Interactions in an Evolved Graph Cellular Automaton

Anthony Nash, PhD

Independent Researcher, Warwickshire, United Kingdom

ORCID: https://orcid.org/0000-0001-8212-0302

info@acnash.co.uk

Abstract

We investigated how local interactions generate predictive information and complex dynamics by representing molecules as graph cellular automata. In this representation, numerical atom states evolve through interactions along chemical bonds, producing molecular space-time trajectories that provide fingerprints for predicting enzyme inhibition. Ridge regression converts these fingerprints into predictions. In the OpenADMET CYP Inhibition Challenge, blind MA-ST-RAE scores were 1.0092 for fixed-rule backpropagation, 1.0020 for genetic configuration search with backpropagation and 0.9946 for genetic programming with backpropagation. Separate dynamical analyses identified convergence, possible periodic behaviour and persistent complex variation. Two systems exhibited bounded hyperchaotic dynamics, with small initial differences growing in multiple independent directions. Discontinuities in the phase-based cellular automaton transition rule and unresolved fractal dimension leave their classification as strange attractors provisional. These findings support molecular space-time as a framework for studying how local interactions generate predictive information and dynamical complexity.

Keywords: cellular automata; complex systems; emergence; genetic programming; graph dynamics; molecular representation; nonlinear dynamics

1. Introduction

Cellular automata demonstrate how repeated local interactions produce organised behaviour, including fixed configurations, oscillators, propagating structures, long transients, computational universality and transitions between order and disorder [1–3]. Reaction–diffusion systems provide a related physical example: local chemical reactions change species concentrations, while diffusion redistributes those species through space, together generating larger-scale patterns [4]. This principle motivates our computational investigation: atoms carry numerical states, bonds define communication routes, and training adjusts their local interactions so that trajectory summaries predict enzyme inhibition.

Our molecular representation brings together three related approaches to learning through repeated local interactions. Learned graph cellular automata provide the network structure by extending shared cellular automaton transition rules to irregular connections such as those between bonded atoms [5]. Within this structure, we draw on neural cellular automata to train the local interactions and allow the numerical state channels to acquire useful roles collectively [6]. Molecular message-passing networks connect this local exchange of information to the prediction task by showing how

communication between bonded atoms can support molecular-property prediction [7]. Together, these ideas motivate our use of trained local interactions on molecular graphs and the retention of their atom-state trajectories as predictive features.

We call the retained atom-state trajectory **molecular space-time**. At generation t, each atom carries a vector of learned numerical values. All possible combinations of these values across the molecule define its computational *state space*. A point in this mathematical space represents one complete set of atom-state values, providing a numerical snapshot of the whole molecule at one generation. Changing these values moves the snapshot to another point, and successive snapshots trace a trajectory through state space. A shared nonlinear cellular automaton transition rule uses messages along typed bonds to advance the graph to generation $t + 1$. Summaries of early changes, intermediate states and final states form a molecular fingerprint, a numerical feature vector for predicting pIC50.

We compared four approaches to learning this representation. Backpropagation trained numerical weights within fixed transition-rule families, while an evolutionary strategy adjusted those weights by testing small changes and using their prediction errors to guide further updates. A genetic algorithm selected model configurations, and genetic programming evolved the mathematical form of the cellular automaton transition rule; both used backpropagation to train the numerical weights within each candidate. All four approaches used fitted ridge regression to combine trajectory features into predictions. The backpropagation-based approaches calculated weight updates through the differentiable ridge fit, while the evolutionary strategy used prediction errors directly to guide its weight changes. Configuration choices included the rule family, number of generations and channels, chemical descriptors and trajectory summaries, as explained in Sections 2.2–2.4 and Appendix A. Genetic algorithms search encoded combinations of model settings, while genetic programming applies selection, crossover and mutation to mathematical expressions defining the cellular automaton transition rule [8, 9].

The 2026 OpenADMET CYP Inhibition Challenge provided an external validation task. Cytochrome P450 (CYP) enzymes influence the metabolism and clearance of many medicines, and inhibition can contribute to clinically important drug interactions [10]. The challenge examined four CYP isoforms, CYP1A2, CYP2C9, CYP2D6, and CYP3A4, which are distinct enzyme members of this family with different substrate preferences. Each isoform defined a separate prediction task: using the compound's molecular structure to predict how potently it inhibits that enzyme. The organisers compared the resulting direct-inhibition pIC50 predictions with concealed experimental measurements. This independent evaluation tests the predictive usefulness of the molecular space-time fingerprints.

We also examined the trained trajectories for convergence, repeating oscillations, persistent complex variation and chaos [1–3, 11–13]. With model parameters fixed, we followed atom states for thousands of generations and measured recurrence, frequency spectra, perturbation growth, Lyapunov exponents and attraction basins. Introducing small initial differences tests whether nearby computational states progressively separate, providing evidence relevant to chaos. We also examined whether trajectories approach a bounded set with fractal geometry, whose dimension describes how its detail changes with measurement scale. Reliable estimates of this fractal dimension require excluding pairs of states recorded too close together in generation number and checking that the estimated dimension remains consistent when we represent the trajectory using different numbers of numerical coordinates [14–16]. These coordinates are the numbers used to locate each state in the chosen representation, such as individual atom-state values or combinations of them that summarise variation across the molecule.

Genetic programming modestly improved blind prediction. The dynamical analyses identified convergence, possible periodic behaviour and persistent complex variation, including two bounded hyperchaotic systems. Phase-boundary discontinuities and unresolved fractal dimension leave their classification as strange attractors provisional.

2. Materials and Methods

2.1. Dataset and Prediction Task

The study used the primary direct-inhibition dataset released for the 2026 OpenADMET CYP Inhibition Blind Challenge (https://openadmet.org/blindchallenges/). It comprised 4,905 unique compounds represented by a molecule identifier and a Simplified Molecular Input Line Entry System (SMILES) string. The dataset provided experimental direct-inhibition pIC50 values for four drug-metabolising cytochrome P450 enzyme isoforms: CYP1A2, CYP2C9, CYP2D6, and CYP3A4. Each value described inhibition of one specific isoform by one compound. Here, pIC50 denotes the negative base-10 logarithm of the half-maximal inhibitory concentration expressed in molar units. Each reported measurement included lower and upper uncertainty bounds and an estimated standard deviation from the fitted concentration–response experiment. The experiments did not necessarily measure each compound against every CYP isoform. The dataset contained 6,525 observed compound–CYP pairs, distributed as summarised in Table 1.

CYP isoform	Compounds with an observed direct-inhibition pIC50
CYP1A2	1,412
CYP2C9	1,285
CYP2D6	1,493
CYP3A4	2,335
Total	**6,525**

Table 1. Number of observed direct-inhibition pIC50 measurements for each CYP isoform.

Each observed compound–CYP pair formed a training case, with molecular structure and isoform identity as inputs and experimental pIC50 as the target. An *endpoint* denotes the direct-inhibition prediction task for one isoform. The CYP-specific systems used observations for their assigned endpoint throughout fitting and selection (Section 2.4). Unmeasured compound–CYP pairs remained missing and contributed neither targets nor loss terms.

We grouped compounds by their standardised Bemis–Murcko scaffold, the structural skeleton comprising rings and their connecting chains after removal of side branches. Reserving 20% of scaffold groups as a sealed holdout left 5,216 observed compound–CYP pairs for fitting and internal selection and 1,309 for final validation. Scaffold groups were exclusive to each partition, allowing evaluation beyond closely related chemistry. Hyperparameter selection used the fitting pool alone.

The challenge test set contained 750 additional compounds for which the organisers withheld all direct-inhibition labels. The challenge required the four trained CYP-specific graph cellular automaton models to generate one finite pIC50 prediction per isoform for each compound, giving 3,000 blinded compound–CYP predictions. We excluded blinded compounds and their unreleased outcomes from parameter estimation, hyperparameter selection, early stopping, and dynamical candidate selection.

Predictive evaluation followed the challenge formulation: the primary measure was the macro-averaged soft-threshold relative absolute error (MA-ST-RAE) across the four CYP isoforms, with equal weight for each isoform and zero error for predictions falling within the reported experimental uncertainty interval. Lower MA-ST-RAE indicates better prediction under this uncertainty-aware measure. We retained conventional regression statistics as complementary measures of predictive agreement, including root mean squared error (RMSE), which summarises prediction-error magnitude, and rank correlations, which assess how well predictions preserve the ordering of experimental values.

2.2. Molecular Graph Representation

We used RDKit to standardise SMILES structures, including common groups of bonded atoms and their formal atomic charges (integer charges assigned to individual atoms, such as +1 or −1). We then retained the largest organic fragment and applied RDKit's sanitisation procedure to check the consistency of the molecular representation, including whether each atom's number and types of bonds were compatible with its assigned chemical state. We converted the resulting structure to canonical isomeric SMILES. We represented each standardised molecule as a graph G=(V,E), where every heavy atom (an atom other than hydrogen) i ∈ V was a cellular-automata cell and every covalent bond defined two directed message-passing edges, j → i and i → j. The predictive input, meaning the model inputs for predicting pIC50, comprised the standardized two-dimensional graph connectivity, atom and bond descriptors, and CYP context; it contained no molecular geometry.

The standard set of atomic descriptors contained one-hot element identity for H, C, N, O, F, P, S, Cl, Br, I, and other elements; for example, with this ordering the encoding represented carbon as (0,1,0,0,0,0,0,0,0,0) and oxygen as (0,0,0,1,0,0,0,0,0,0). Further descriptors comprised formal charge; aromaticity; sp, sp2, sp3 (types of atomic orbital hybridisation), and other hybridisation states; degree; total attached hydrogens; ring membership; hydrogen-bond donor and acceptor status; and tetrahedral chirality. We also tested additional atomic descriptors, organised into the five groups in Table 2. To choose which groups to include, we compared their predictive performance using training and validation subsets drawn entirely from the fitting data. The sealed holdout and blinded challenge set played no part in that selection.

Feature group	Atom-level quantities
Periodic	Atomic number, atomic mass, covalent radius, van der Waals radius, outer electrons, period, group, and approximate atomic volume.
Valence	Total and implicit valence, heavy-atom degree, radical electrons, and absolute formal charge.
Electronic	Pauling electronegativity, approximate polarizability, heteroatom and halogen indicators, conjugated-bond fraction, first ionization energy, electron affinity, and missing-property indicator.
Ring geometry	Number of rings containing the atom and indicators for ring sizes 3, 4, 5, 6, 7, and 8 or greater.
Local environment	Mean neighbouring electronegativity, electronegativity difference from the neighbourhood, mean neighbouring atomic number, and mean neighbouring formal charge.

Table 2. Chemically organised atom-feature groups considered during training-only model selection. A feature group is a named collection of related atom-level descriptors that we added to the baseline encoding as one candidate block.

The *feature profile* specifies which chemical descriptors form each atom's input vector: the baseline encoding together with the selected additional feature groups in Table 2. We stored the selected feature profile with every frozen checkpoint, meaning the saved model whose parameters we held fixed for evaluation. This preserved the exact feature names and order required to predict new molecules. We scaled continuous quantities by fixed chemically meaningful constants during graph construction; for example, we divided atomic mass in daltons by 250, converting carbon's mass of approximately 12.01 into a dimensionless input of approximately 0.048. The bond vector e_{ji} contained one-hot indicators for single, double, triple, or aromatic bonds, together with indicators for conjugation, ring membership, and three stereochemical properties. We attached the same bond vector to both directed representations of an undirected bond.

A four-entry context vector, $\boldsymbol{c}$, identified the target CYP isoform: its entry was 1 and the remaining entries were 0. This one-hot encoding supplied target identity to the cellular automaton and prediction readout, allowing one molecular graph to generate isoform-specific trajectories and predictions.

Each atom carries H numerical state values, called channels, which change over generations. They are latent because training determines how they combine the chemical descriptors to support prediction. The N-methylacetamide example shows how each heavy atom forms a cell and how bonds allow information to pass in both directions, with x_i recording the atom's chemical descriptors, including its attached hydrogen count, and e_{ji} recording the bond descriptors (Figure 1A). To form an atom's initial state, the model combines its F input descriptors using learned weights that determine how strongly each descriptor contributes to each of the H channels (Figure 1B). Scaling and the hyperbolic tangent keep the initial values between −1 and 1, and the same learned transformation applies to every atom. The illustrated conversion of 20 descriptors into 8 channels is an example of how the input and state sizes can differ. The resulting channel values subsequently change through interactions with bonded neighbours, forming the molecular space-time trajectory.

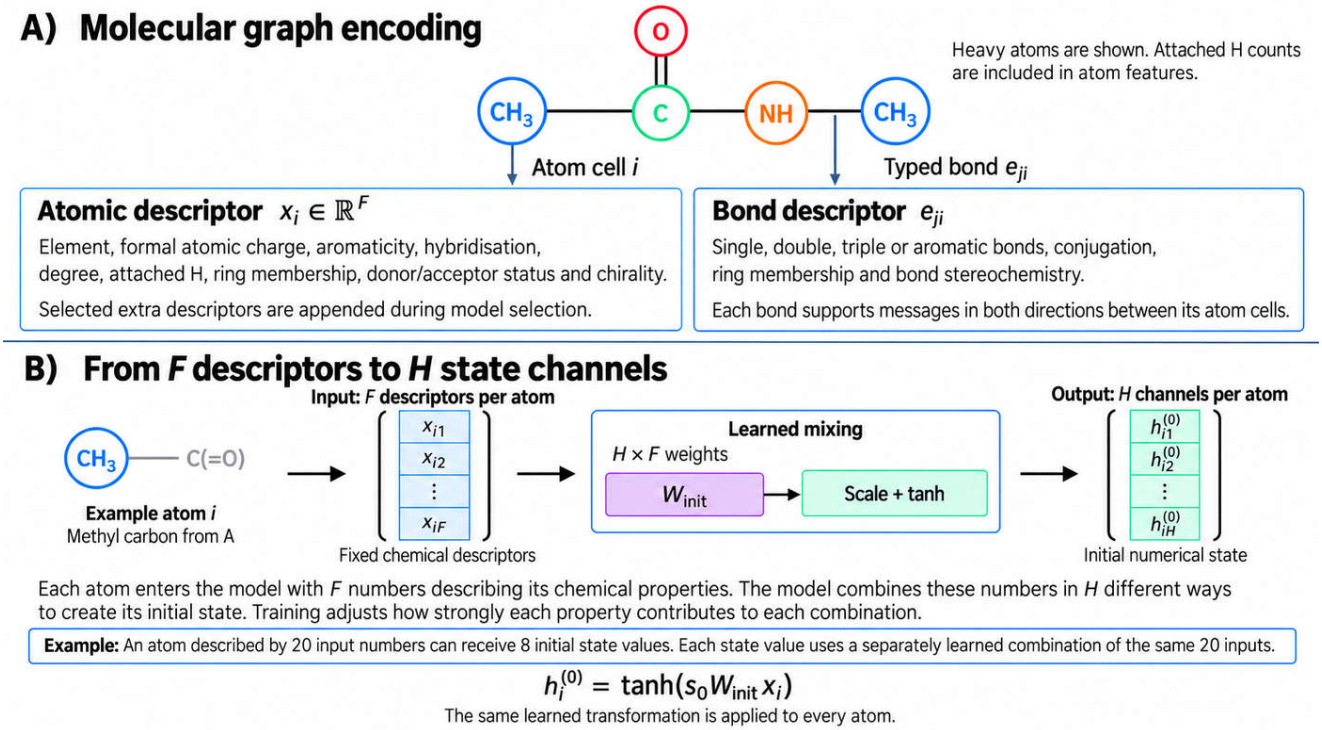


Figure 1. Molecular graph encoding and formation of the initial dynamical state. A) N-methylacetamide illustrates the atomic cells and bond descriptors. **B)** The highlighted methyl carbon illustrates how F chemical descriptors become H initial state values through a learned transformation. The pale bond indicates its connection to the rest of the molecule. The example of 20 descriptors and 8 channels is illustrative.

2.3. Nonlinear Graph Cellular Automaton

The Graph Cellular Automaton (Graph-CA) advances the initial atom states defined in Section 2.2 through repeated local interactions. Each state's role depends on its effects on neighbours, subsequent generations and prediction. We screened ten Graph-CA transition-rule families, illustrated in Figure 2 and defined through their shared calculations and individual implementations in Appendix A. Section 2.4.1 identifies those selected for prediction, and Table 6 reports their dynamical screening results. The ten families were gated residual, inertial reaction–diffusion, FitzHugh–Nagumo, Kuramoto–Sakaguchi, delayed memory, Gray–Scott, coupled map, activator–inhibitor, damped symplectic and conservative graph flux.

A rule family specifies how an atom calculates its next numerical state. All ten use atomic cells connected by chemical bonds, with one local rule shared across atoms and repeated at every generation. They differ in how they use the atom states, bonded-neighbour information and any retained earlier states. Chemical descriptors set the initial states; most rules also use chemical descriptors and CYP context during subsequent updates. The conservative graph flux rule subsequently exchanges state values directly along bonds. We distinguish this rule here because its updates redistribute

the existing state through equal-and-opposite transfers that preserve the molecule-wide total in each channel, with atomic descriptors contributing through the initial states. Repeated updates allow information to spread through the molecular graph.

Gated residual. This rule forms a bounded mixture of the current state and a proposed value calculated from local inputs [17] (Figure 2A). A learned gate controls the mixture separately for each atom, channel and generation, providing adaptive memory through selective retention or change. The approach relates to gated recurrent neural networks [18] and neural cellular automata [19].

Inertial reaction–diffusion. This rule combines locally generated change, numerical exchange with bonded neighbours and restoration towards a central state [4] (Figure 2B). A velocity-like variable carries earlier change into the next generation [20], allowing overshoot, damping and propagation of disturbances. Here, diffusion describes numerical information exchanged across bonds; physical reaction–diffusion supplies the mathematical analogy.

FitzHugh–Nagumo. This rule divides each atom's state into faster excitation and slower recovery components [21, 22] (Figure 2C). Local input can move excitation away from rest, with recovery regulating or reversing the change later. Chemical information and bonded-neighbour exchange shape both components, allowing threshold responses, pulses and oscillatory return.

Kuramoto–Sakaguchi. Each channel represents a phase within a repeating cycle [23, 24] (Figure 2D). Bonded atoms interact through phase differences, while chemical information and CYP context determine their preferred rates of advance. Neighbours may synchronise, retain an offset or resist alignment. Phases repeat after a complete turn, creating a circular state space whose numerical boundary requires care in the dynamical analyses (Sections 2.5 and 3.3).

Delayed memory. Each atom accesses a retained earlier state [25] (Figure 2E). The next state combines current local inputs with a learned response to that earlier state and the difference between past and present. The delay controls when information returns, allowing feedback to reinforce or oppose recent changes after several rounds of neighbour communication.

Gray–Scott. This rule (Figure 2F) divides each atom's channels into two interacting groups. A nonlinear term transfers numerical activity from one group to the other, while replenishment, removal and exchange with bonded neighbours shape their evolution. The chemical descriptors and CYP context also contribute to the updates.

Coupled map. This rule (Figure 2G) applies a curved, nonlinear function to each current state value and combines the result with a corresponding calculation from the mean neighbour state. Repeating this simple calculation can generate steady, repeating or irregular patterns, depending on the settings. This rule supplies the candidate period-two oscillator shown later in Figures 6 and 9.

Activator–inhibitor. This rule (Figure 2H) divides the channels into an activating group and a regulating group. The first can amplify local changes, while the second feeds back on that activity; both groups also exchange information with bonded neighbours. It is closely related to the FitzHugh–Nagumo implementation, with different settings for the strengths of feedback, neighbour exchange and local inputs.

Damped symplectic. This rule (Figure 2I) divides the state into position-like and momentum-like channel groups. It first updates the momentum-like values from local inputs, neighbour differences and a restoring term, then uses them to update the position-like values. Damping reduces retained momentum, and bounding functions limit both channel groups; these are numerical analogues of motion used to organise information on the molecular graph.

Conservative graph flux. This rule (Figure 2J) transfers numerical state between bonded atoms. Each transfer adds to one atom what it subtracts from the other, preserving the molecule-wide sum in each channel up to numerical rounding. The atom-state differences, learned weights and bond descriptors determine the transfers.

The rule names identify the mathematical ideas that inspired these methods, all of which generate learned numerical trajectories on molecular graphs. The ten families share graph construction, available descriptors, bond information and the prediction task, with selected features and use of CYP context depending on the configuration. Their different local calculations allow us to investigate how the transition rule influences prediction and dynamical behaviour. Section 4.4 describes the descriptor-only control still needed to establish the additional predictive value of repeated interactions.

Each Graph-CA transition rule produces a trajectory containing the state of every atom at every generation. Molecules contain different numbers of atoms, so the complete trajectories cannot be passed directly to one fixed-size prediction model. We therefore summarised each trajectory across atoms and across selected generations. These

summaries captured final state, variation between atoms, change through time and behaviour at several temporal scales. We collected these summaries into a molecular fingerprint, a fixed-length list of numerical features representing the molecule's trajectory. We supplied this fingerprint to ridge regression. Optional summaries allowed the genetic search to test whether persistence, correlation, squared changes in numerical states or the direction of change added useful information. Appendix A defines these quantities and explains the CYP-specific fingerprint construction. Figure 2 illustrates these trajectories using separately trained systems and different example molecules, with some molecules appearing under multiple rules.

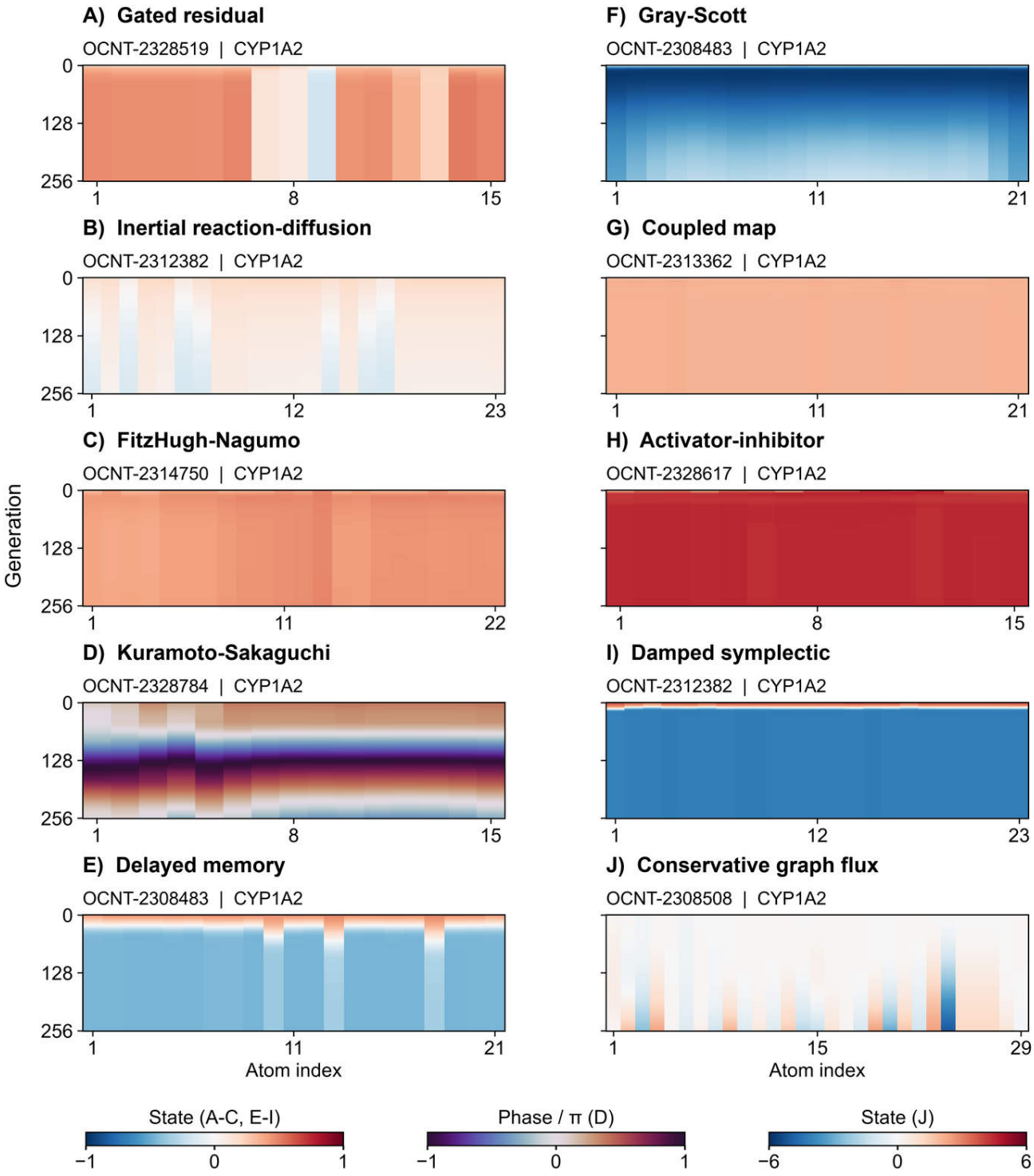


Figure 2. Recorded molecular cellular evolution under ten transition-rule families. Panels A-E run down the left column and F-J down the right. Each title identifies the rule and the example compound's OCNT identifier from the OpenADMET dataset. All examples use CYP1A2 context. Columns show heavy atoms in stored graph order; rows show every generation from 0 to 256. Interactions follow molecular bonds. Colour shows the first state channel. Panels A-C and E-I share a scale from −1 to 1; panel J uses −6 to 6 because its recorded values extend beyond that range. Panel D shows the first phase angle divided by π, with −1 and 1 representing the same phase. Values retain their original amplitudes. The panels illustrate separately trained systems and are not a matched comparison on one molecule. Rule families C, E, F, I and J contributed to the BP-Graph-CA submission across its four CYP targets; these example trajectories are illustrations, not the submission predictions.

2.4. Learning the Molecular Space-Time Representation

All four learning methods used ridge regression to convert molecular trajectory fingerprints into pIC50 predictions (Figure 3). Pooling supplied the fixed-length fingerprints described in Section 2.3. Ridge regression combined their values using coefficients calculated from training observations, with a penalty discouraging very large coefficients to help control overfitting. The methods differed in how they trained or selected the Graph-CA producing these fingerprints. Backpropagation works backwards from prediction error to calculate how trainable numerical weights should change. Here, it passed through the ridge fit and every retained Graph-CA generation. The four methods used this calculation or an evolutionary alternative as follows:

- **Fixed-rule Graph-CA (BP-Graph-CA):** backpropagation trained numerical weights within a chosen cellular automaton transition rule and the other Graph-CA components; the rule's mathematical form stayed fixed.
- **Evolutionary strategy (ES-Graph-CA):** small random changes to the weights produced candidate models. Their prediction errors indicated useful directions of change, guiding updates without backpropagation.
- **Hybrid genetic search (GA-Graph-CA):** a genetic algorithm selected and recombined model *configurations*: choices of rule family, number of generations and channels, chemical descriptors and trajectory summaries. Backpropagation trained each candidate's numerical weights.
- **Genetic programming (GP-Graph-CA):** genetic search changed the mathematical transition rule, including how addition, multiplication or averaging combined local inputs and neighbour-state differences. Backpropagation trained each candidate's numerical weights, and validation error guided selection. Each candidate rule remained fixed throughout its molecular trajectories.

All training and model selection used the fitting data and its internal scaffold-separated validation folds, keeping structurally related molecular frameworks together. The sealed holdout and blind challenge labels were excluded from these decisions. Sections 2.4.1–2.4.5 describe the training procedures; Section 2.4.6 describes final evaluation and blinded inference.

2.4.1. Fixed-rule Graph-CA with backpropagation

The fixed-rule reference, BP-Graph-CA, used a stacked ensemble for each CYP isoform. Each component Graph-CA was trained separately with a particular transition rule and model configuration, such as its channel count and number of generations. A second regression model then learned how to combine their predictions. Training used backpropagation through time (Figure 3A), credible-interval alignment, measurements for the CYP isoform being predicted and cross-validation, as detailed below.

The Adam optimiser used these gradients to adjust the numerical weights that set the initial atom states, pass information along bonds, combine each atom's state with neighbour information, chemical descriptors and CYP context, and calculate the next state using the selected cellular automaton transition rule. We gradually reduced the learning rate, which controls the size of the adjustments to the model's weights, along a smooth curve defined by the cosine function. This allowed progressively finer adjustments as training continued. Gradient clipping limited exceptionally large updates, and an L2 penalty discouraged large cellular automaton weights. Using only the training data and its internal validation folds, we selected the number of generations and dynamical channels, atom-feature profile, learning rate, ridge penalty, update scale, proportion assigned to the support subset, batch size, bond temperature, initialisation scale and noise, pooling design, and constants specific to each transition rule.

BP-Graph-CA trained four independent nonlinear prediction systems, one for each CYP isoform. A CYP-specific system used only measurements for its target isoform when fitting the ridge readout, calculating prediction error, deciding when to stop training and choosing which saved checkpoint to retain. Mean squared error measured the average squared prediction error. We supplemented it with a smooth credible-interval penalty, which added a cost when a prediction fell outside the experimental uncertainty interval reported for that measurement.

For each CYP prediction target, we first screened all ten transition-rule families described in Section 2.3 using two scaffold-separated folds. The three strongest combinations of rule and model configuration then advanced to confirmation using five scaffold folds and two independent training seeds. Each out-of-fold prediction came from a model that had not used the corresponding validation fold for fitting. We used the validation predictions to select the Graph-CA models included in the final stacked ensemble and trained a ridge-regression stacking model to determine how their predictions should be weighted. Once the models and weights had been fixed, we evaluated the ensemble on the previously untouched holdout set. The final CYP1A2 system combined individual Graph-CA models using FitzHugh–Nagumo, Gray–Scott and conservative graph flux. CYP2C9 combined Gray–Scott and damped symplectic; CYP2D6 used delayed memory and FitzHugh–Nagumo; and CYP3A4 used damped symplectic, FitzHugh–Nagumo and delayed memory. Internal validation selected these five families from the ten-rule search illustrated in Figure 2.

2.4.2. Shared ridge readout and its role in training

For every training batch, we divided the molecules into a readout-fitting subset, hereafter called the *support subset*, and a *query subset* used to assess the fitted readout. The Graph-CA trajectory of each molecule was summarised as a fixed-length numerical fingerprint. We used the fingerprints and measured pIC50 values of the molecules in the support subset to fit the ridge-regression readout. We then used that fitted readout to predict the query molecules, which had not contributed to the ridge fit. This tested whether the relationship learned between the trajectory fingerprints and pIC50 could predict molecules that had not been used to fit the ridge-regression readout. BP-Graph-CA, GA-Graph-CA and GP-Graph-CA differentiated through this fit when training numerical weights. By contrast, ES-Graph-CA used error on the query subset as the score used to compare candidate weight sets and estimated updates to its numerical weights by comparing perturbed candidates. Separately, GA-Graph-CA and GP-Graph-CA generated alternative model designs, trained the numerical weights of each design, and compared the completed models using internal validation folds.

The following matrix construction, feature standardisation, target centring and ridge calculation allowed fingerprint features with different numerical scales to contribute fairly, separated the average pIC50 from feature-dependent variation, and provided a stable, differentiable relationship between the trajectory fingerprints and pIC50. The fingerprints of the molecules in the support subset formed the matrix F_s: each row represented one molecule in the support subset and each column represented one fingerprint feature. We standardised every column by subtracting its support-set mean and dividing by its support-set scale, producing Z_s. We also centred the measured support targets by subtracting their mean, giving $y_s - \bar{y}_s$. The following closed-form calculation then produced the ridge coefficients β [26]:

$$\beta = \left(Z_s^T Z_s + \lambda I\right)^{-1} Z_s^T (y_s - \bar{y}_s).$$

In this expression, $Z_s^T Z_s$ describes relationships among the standardised fingerprint features. The ridge term λI adds the non-negative penalty λ to the diagonal through the identity matrix I, reducing sensitivity to large or highly correlated coefficients. The inverse matrix then gives the coefficient vector that balances fit to the centred support targets against this penalty. For a query molecule with fingerprint f_q, the predicted pIC50 was

$$\hat{y}_q = \bar{y}_s + \left(\frac{f_q - \bar{F}_s}{s_F}\right)^T \beta,$$

Here, $\bar{F}_s$ and s_F are the support-set mean and scale for each fingerprint feature. Subtracting $\bar{F}_s$ and dividing by s_F applies the same standardisation to the query fingerprint that was applied to the support fingerprints. The transposed expression in parentheses multiplied by β is the dot product: it multiplies each standardised fingerprint feature by its corresponding ridge coefficient and adds the products to produce the centred prediction. Adding $\bar{y}_s$ restores the mean pIC50. The ridge penalty was applied only to the fingerprint coefficients. It was not applied to the intercept, which represented the mean pIC50 of the support subset rather than the influence of a fingerprint feature.

The training software tracked each mathematical operation used to produce the predictions. It then worked backwards from the prediction error to calculate how each Graph-CA weight should change. Because the ridge coefficients depended on the molecular fingerprints, this calculation passed through both the ridge-regression fit and the successive Graph-CA generations that produced the fingerprints. When several trajectory features contained nearly the same information, the ordinary matrix calculation could become numerically unstable. In these cases, we used a pseudoinverse, which is a stable alternative to direct matrix inversion, to obtain coefficients for the same ridge-regression objective.

After Graph-CA training, we refitted the ridge readout using all observations permitted for fitting and stored it with the selected checkpoint. MA-ST-RAE on the internal scaffold-separated validation fold determined early stopping and checkpoint selection; RMSE provided a secondary diagnostic.

2.4.3. Evolutionary-strategy optimisation of numerical weights

We constructed ES-Graph-CA as a derivative-free comparison method that updated parameters without calculating gradients (Figure 3B). To isolate this difference in training, it retained the same molecular graphs, chemical encodings, recurrent transition rules, stored trajectories, scaffold partitions and ridge-regression prediction step as the backpropagation system.

The evolutionary strategy created paired candidates by adding and subtracting the same random change from the current weights. These opposite, or *antithetic*, perturbations allowed a direct comparison of directions. Each candidate underwent the support/query ridge procedure in Section 2.4.2. Lower query error identified the more useful direction, and the optimiser combined evidence across pairs using Adam-style running averages to update the weights.

The evolutionary-strategy campaign used a fixed CYP3A4 damped-symplectic Graph-CA with 16 channels and 32 recurrent generations. Its atom inputs comprised the baseline descriptors supplemented by the Periodic and Electronic feature groups in Table 2. We compared 32 evolutionary configurations that varied the number and size of the candidate weight perturbations, the evolutionary learning rate, training-batch size and composition, and the stopping criteria. Appendix B defines these settings and reports their tested values.

Each evolutionary configuration was screened using two scaffold folds, producing 64 runs. The five strongest configurations then underwent confirmation across five scaffold folds and two independent seeds, producing 50 additional runs. Performance of the backpropagation-trained model on the same two screening folds provided a matched reference, while the confirmation stage tested whether the evolutionary results were reproducible. ES-Graph-CA remained an internal comparison and was not evaluated on the sealed holdout or blind challenge set.

2.4.4. Hybrid genetic Graph-CA structure search

We next used a genetic algorithm to search the design of the model predicting CYP3A4 inhibition while retaining backpropagation to train its numerical weights (Figure 3C). Each candidate design specified the transition-rule family, number of generations and channels, atom features, trajectory pooling and optional model components. We evaluated four successive populations, each containing 24 candidate model configurations, using two scaffold folds. Each population represents one round of the genetic search, which we call a genetic generation. Within each candidate model, cellular automaton generations describe the successive updates of the atoms' numerical states that form a molecular trajectory. The four strongest designs and the existing backpropagation-trained CYP3A4 model configuration used for comparison then advanced to five-fold, two-seed confirmation. We used MA-ST-RAE on the internal validation folds to compare candidate model configurations, with lower error favouring selection for the next search population and advancement to confirmation. We also recorded RMSE as a complementary measure of prediction error. Appendix C defines the design choices, genetic operations and training budget.

The hybrid genetic search selected a model configuration using the FitzHugh–Nagumo cellular automaton transition rule after comparing candidate designs on the scaffold-validation folds. This selected configuration used 128 cellular automaton generations, 24 channels and the comprehensive atom-feature profile. Appendix C lists the design choices tested (Table C1) and the complete design chosen by the search (Table C2), including which trajectory summaries it enabled or disabled.

2.4.5. Genetic programming of the cellular automaton transition rule

GP-Graph-CA evolved mathematical expressions for the cellular automaton transition rule used by the model predicting CYP3A4 inhibition (Figure 3D), retaining the selected configuration in Section 2.4.4 for the other model components. This isolated changes to the rule from changes in graph encoding, channel count and training procedure.

We represented each candidate transition rule as an *expression tree*, a branching structure that specifies a mathematical calculation. Its outermost points, called leaves, hold inputs, while the points joining branches specify operations such as addition, multiplication or averaging. Evaluating the tree combines these inputs in the order defined by its branches to produce the rule's output. For example, a tree that averages the local-input vector and the bonded-neighbour difference has two input leaves joined by one averaging operation. Candidate trees could draw on the atom's local-input vector, current state, bonded-neighbour difference and delayed-state difference, and genetic programming could modify or exchange branches to produce new rules. We screened four successive populations, each containing 24 candidate cellular automaton transition rules, using two scaffold folds, then confirmed five finalists using five folds and two seeds. Backpropagation trained the numerical weights within the model using each candidate cellular automaton transition rule. We used MA-ST-RAE on the internal scaffold-separated validation folds to compare candidate cellular automaton transition rules, with lower prediction error favouring selection for the next search population and advancement to confirmation. We also recorded RMSE as a complementary measure of prediction error. Appendix D defines the candidate inputs (Table D1), permitted operations (Table D2), genetic-programming procedure and training budget.

The cellular automaton transition rule selected by genetic programming used a single averaging operation to combine its two inputs

$$u_i^{(t)} = \frac{1}{2}\left(r_i^{(t)} + \bar{h}_i^{(t)} - h_i^{(t)} \right),$$

The expression averages two H-channel vectors: $r_i^{(t)}$, which combines the atom's local inputs, and $\bar{h}_i^{(t)} - h_i^{(t)}$, the bonded neighbours' mean state minus the atom's own state. Selection and recombination of executable expressions followed the genetic-programming approach introduced above.

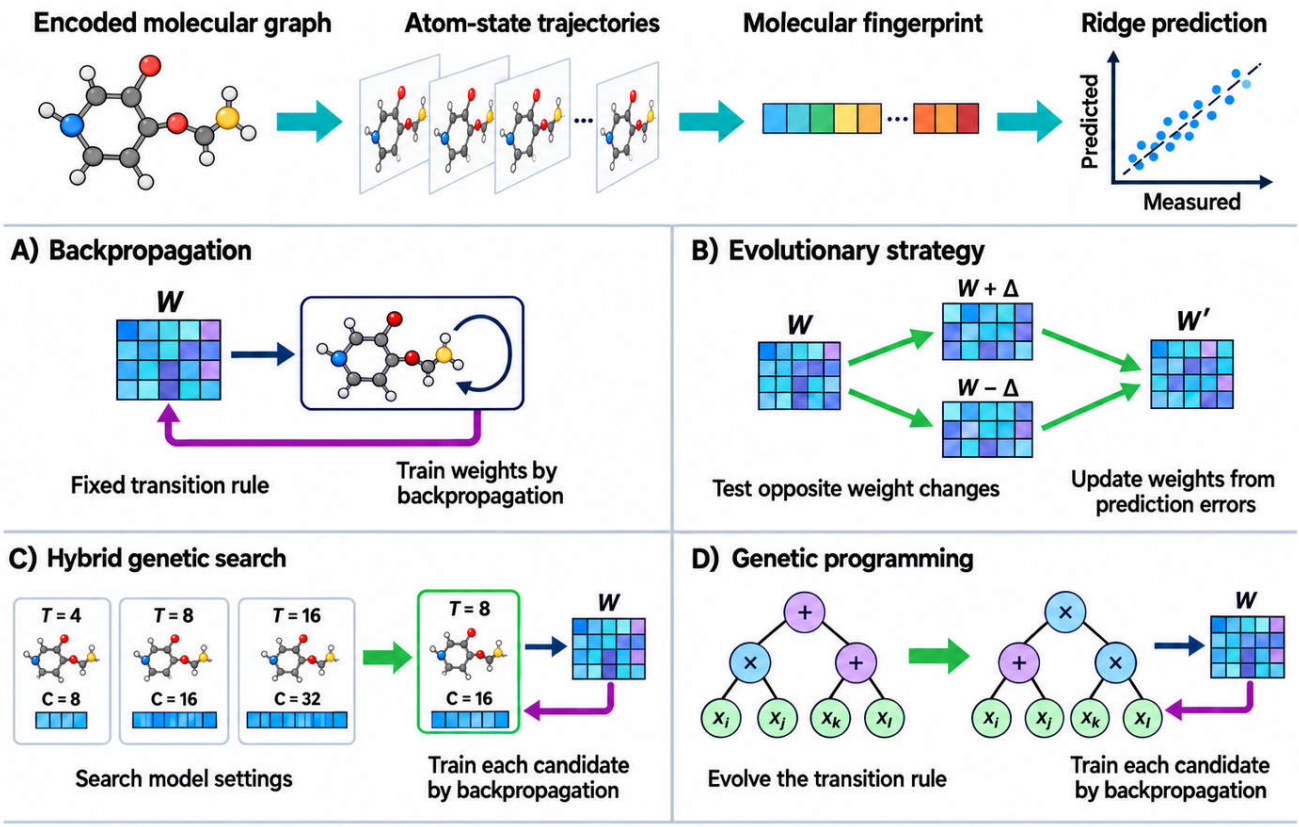


Figure 3. Four strategies for learning molecular space-time representations. The shared workflow converts encoded molecular graphs into atom-state trajectories, summarises these as molecular fingerprints and fits ridge regression to predict pIC50. **A**, BP-Graph-CA trains numerical weights by backpropagation through the ridge fit and successive cellular automaton generations, using a fixed form of transition rule. **B**, ES-Graph-CA tests equal and opposite changes to the weights and uses their prediction errors to determine the next weight update. W denotes the numerical weights, Δ a trial change and W′ the updated weights. **C**, GA-Graph-CA searches model-design settings and trains each candidate by backpropagation. The example cards illustrate the generation count T and channel count C; C corresponds to H in the manuscript. **D**, GP-Graph-CA evolves the cellular automaton transition rule by changing mathematical expression trees, while backpropagation trains each candidate's numerical weights. The tree leaves represent illustrative numerical inputs. Both genetic searches use internal scaffold-validation error to guide selection, with the expression-size penalty also applied in genetic programming. Purple arrows indicate backpropagation and green arrows indicate evolutionary updates or genetic search. Transition rules remain fixed within individual trajectories.

2.4.6. Final validation and blinded inference

Model development used five scaffold-grouped folds drawn from the fitting pool. Structurally related molecular frameworks were therefore kept within the same fold. We opened the previously defined sealed holdout only once, after selecting the individual Graph-CA models and the endpoint-specific stacked ensembles. The hybrid genetic CYP3A4 structure and genetic-programming rule were fixed after five-fold internal-validation confirmation with two seeds. Each was then evaluated once on the sealed holdout. We used the primary and complementary metrics defined in Section 2.1 and estimated final uncertainty with 1,000 bootstrap resamples. ES-Graph-CA served only as an internal comparison, while BP-Graph-CA, GA-Graph-CA and GP-Graph-CA were submitted for blinded evaluation.

After all model parameters had been fixed, checkpoints trained across five scaffold folds and two seeds generated predictions for every blinded molecule-CYP pair under each selected rule. We first averaged the predictions from checkpoints using the same rule. The saved endpoint-specific ridge-regression stacking model then produced the final pIC50 prediction. GA-Graph-CA retained the frozen fixed-rule predictions for CYP1A2, CYP2C9 and CYP2D6, but replaced CYP3A4 with a combination of ten trained models predicting CYP3A4 inhibition, selected by the hybrid genetic search. GP-Graph-CA used the same three frozen fixed-rule endpoint ensembles and replaced CYP3A4 with the

ten-checkpoint genetic-programming ensemble. The experimentally measured pIC50 values for the competition's blind test compounds remained hidden from us, and we trained, selected and locally validated our models using only the labelled data available before submission.

2.5. Long-Horizon Dynamical Analysis

We examined the molecular space-time trajectories to understand what kinds of behaviour emerge when atoms repeatedly exchange information along molecular bonds. By continuing the trained cellular automata for many more generations, we tested whether their numerical states settled to fixed values, repeated in cycles or showed sustained sensitivity to small changes in the starting state. We also investigated whether the resulting trajectories provided evidence for strange attractors, including attraction towards a bounded region and reproducible fractal geometry, and whether molecular structure influenced this behaviour. These analyses examined the dynamics of the learned numerical states separately from the models' accuracy in predicting pIC50.

2.5.1. Selecting trajectories for tests of convergence, oscillation and chaos

We analysed trajectories after training, keeping the learned numerical weights, atomic descriptors and bond connections unchanged. We repeatedly applied the trained cellular automaton transition rule to follow how the atoms' numerical states evolved over successive generations, without further training.

We used several broad descriptions of behaviour. *Convergence* means that a trajectory approaches a fixed state. *Periodic behaviour* means that it repeats a sequence of states. *Persistent complex variation* means that change continues without an immediately established classification. A *bounded* trajectory remains within a finite numerical range. To quantify sensitivity to small changes in the initial state, we used Lyapunov exponents, which measure the average exponential rates at which small differences between trajectories grow or shrink as the system evolves [27]. A positive exponent indicates that an initially small difference grows exponentially on average, revealing sensitivity to initial conditions. More than one positive exponent indicates expansion along several independent directions, termed hyperchaotic dynamics over the measured interval. Dynamical tests used the complete atom-by-channel state, including auxiliary variables such as velocity and retained earlier states whenever the rule required them to calculate the next generation.

We prioritised continued motion, recurrence, variation across a range of timescales and sensitivity to small initial changes when selecting trajectories for detailed analysis. The initial screen used complete atom-by-channel validation trajectories. For each trajectory H_t, we first calculated the mean state across all atoms at every generation. To measure continued change, we calculated the average Euclidean distance between successive molecule-wide mean states during the second half of each trajectory. We refer to this measure as *late-time motion*. Recurrence analysis examines whether a system returns close to an earlier state [28]. We used this principle to define a study-specific screening measure, which we called the *recurrence ratio*. For each interval from 2 to 64 generations, subject to the available trajectory length, we calculated the average distance between molecule-wide mean states separated by that interval. We divided this distance by the interval multiplied by the average change per generation, then retained the smallest ratio. Smaller values indicate less separation relative to this estimate of accumulated change. Dividing by the interval can favour longer intervals, and averaging across atoms can conceal differences between individual atom states. We therefore used the ratio to select trajectories for further analysis; a small value alone cannot establish a genuine return or periodic behaviour.

We calculated spectral entropy to measure how widely the trajectory's changes were spread across different frequencies, from slow variations to rapid fluctuations [29]. Here, frequency describes the number of cycles of rising and falling numerical state values per generation: slow changes extend over many generations, while rapid changes occur over fewer generations. For each channel, we used a Fourier transform to separate the molecule-wide mean state into these frequency components and measured how much each contributed to its variation. We excluded the zero-frequency component, which represents the constant average level. Low spectral entropy indicates that a few

frequencies dominate, while high spectral entropy indicates that many frequencies contribute. We calculated it for every state channel and then averaged across channels. The screening score combined continued motion, frequency diversity and recurrence:

$$S = \frac{v_{\text{late}}\left(1 + H_{\text{spectral}}\right)}{R_{\text{recurrence}}},$$

Here, v_{late} is late-time motion, H_{spectral} is normalised spectral entropy, and $R_{\text{recurrence}}$ is the recurrence ratio. We combined these measurements to prioritise trajectories that continued to change, varied across several timescales and returned close to earlier states. Continued change and greater spectral entropy increased the score, while a smaller recurrence ratio increased it by indicating closer returns. The +1 prevented low spectral entropy from reducing the complete score towards zero. Trajectories dominated by a few frequencies could therefore still receive a high score through continued motion and recurrence, while greater frequency diversity increased the score further. We used this study-specific score to select candidates for further dynamical tests. The population screen covered all ten families; detailed analyses included gated residual, delayed memory, inertial reaction–diffusion, Kuramoto–Sakaguchi and FitzHugh–Nagumo, together with the coupled-map oscillator example.

We propagated 20 complete molecular states for 5,000 generations using frozen Graph-CA parameters. Every channel of every atom was retained at every generation. The first 1,000 generations were treated as *burn-in* and excluded from late-time measurements, allowing initial transient behaviour to pass before analysis. We examined ten representative cases in greater detail using principal-component projections, which provide lower-dimensional coordinates that capture major patterns of variation, together with recurrence plots, frequency spectra, estimates of correlation dimension and curves describing the separation of neighbouring trajectories. We used these coordinates only to display the trajectories. All perturbation and Lyapunov calculations used the complete atom-by-channel state, so dimensional reduction did not remove information from the quantitative dynamical tests.

2.5.2. Direct perturbation screen

For each selected reference trajectory, we generated eight independent random perturbation vectors across all atom-by-channel state values. Each vector specified a different pattern of small increases and decreases and was scaled to a total Euclidean magnitude of 10^{-5}. We applied each perturbation to the initial complete molecular state to create a companion trajectory, then advanced it alongside the unchanged reference under exactly the same frozen transition rule, molecular graph, bond features, atom features and CYP context. At every generation, we recorded the distance between the reference and companion states.

Candidates were prioritised when their companion trajectories repeatedly moved away from the reference across several perturbation vectors. In Figure 4A, the upper diagram shows the reference and a nearby companion, while the lower curves illustrate how we followed their separation through successive generations. We looked for separation that increased across several initial perturbations; a later plateau can arise as the trajectories explore the available state range, so the initial screen was followed by the confirmatory tests described in Sections 2.5.5–2.5.7.

2.5.3. Circular state-space distance

In the Kuramoto–Sakaguchi rule, each state channel records a position around a repeating cycle rather than a position on a straight line. The phase angle is rescaled so that 0 represents zero radians, −1 represents −π and +1 represents +π. Because −π and +π meet at the same point on the cycle, values near opposite ends of the numerical range are close in phase, and crossing one end wraps the state to the other. An ordinary numerical subtraction would incorrectly treat states close to opposite ends as far apart. We therefore measured their shortest separation around the circle. For reference state h and companion state h', the circular difference in each channel was

$$\Delta(h', h) = \frac{1}{\pi}\text{atan2}\left[\sin\left(\pi(h' - h)\right), \cos\left(\pi(h' - h)\right)\right],$$

The sine and cosine convert the numerical difference into a circular direction, and atan2 returns its signed shortest angle. Dividing by π returns the result to the model's state scale. The full-state separation $d=\|\Delta(h',h)\|_2$ is the Euclidean length of all these elementwise circular differences. It treats states near opposite ends of the interval as nearby phases and avoids an artificial distance caused only by the wrapped numerical representation. We also tested whether the learned update changed smoothly when a phase crossed between +1 and −1, the point at which the numerical representation wraps around the cycle.

2.5.4. Confirmatory protocols

The numerical settings for the confirmatory and structural-intervention experiments are summarised in Table 3. A *repeat* used either an independently oriented initial perturbation or an independently constructed orthogonal perturbation basis, depending on the calculation.

Analysis	Burn-in generations	Measured generations	Renormalization or sampling interval	Perturbation scale	Repeats
Largest Lyapunov exponent	1,000	4,000	10	10^{-4}, 10^{-5}, and 10^{-6}	8 per scale and molecule
Float64 Lyapunov spectrum	1,000	4,000	5, 10, and 20	10^{-7}	2 per interval and molecule
Attraction basin	1,000	6,000	Sampled every 10	Displacement radii 0.1, 0.5, 1.0, and 2.0	8 per radius and molecule
Structural interventions	1,000	2,000	10	10^{-7}	2 per system

Table 3. Numerical settings for the confirmatory and structural-intervention analyses of Graph-CA dynamics.

2.5.5. Renormalized largest Lyapunov exponent

The first confirmatory analysis tested whether the separation observed in the direct perturbation screen persisted over time. A single separation event may be transient, whereas a positive Lyapunov exponent indicates that nearby states separate exponentially on average over sustained evolution [27]. We therefore estimated the largest Lyapunov exponent using the repeated renormalisation method of Benettin et al. [30]. After burn-in, we placed a companion state at a small distance ε from the reference state. Both states advanced for τ generations. We then measured their circular separation d_k and recorded how much it had grown relative to ε. Before the next interval, we moved the companion back to distance ε while preserving the direction in which the separation had grown. Repeating this procedure tested whether local expansion was continually regenerated rather than caused by a single transient event. The largest Lyapunov exponent was estimated as

$$\lambda_1 = \frac{1}{K\tau}\sum_{k=1}^{K}\log\left(\frac{d_k}{\varepsilon}\right).$$

Here, K is the number of measured intervals, τ is the number of generations in each interval, d_k is the separation at the end of interval k, and ε is the separation imposed at its start. The logarithm converts multiplicative growth into an additive rate, and division by Kτ gives the mean rate per generation. The complete design produced 48 estimates. A positive value indicates that nearby states separated exponentially on average over the measured interval. In Figure 4B, the rising segments illustrate separation during each interval and the dashed arrows return its magnitude to ε. The lower curves show the running estimate as intervals accumulate; we examined whether it settled towards a consistent value across independently oriented starting perturbations.

2.5.6. Float64 Lyapunov spectrum

A single largest Lyapunov exponent measures the fastest average expansion. A *Lyapunov spectrum* extends this idea by measuring growth or contraction along several independent directions through state space. We calculated the eight leading exponents using double-precision float64 arithmetic and eight simultaneous perturbation vectors that began mutually perpendicular.

After the reference trajectory and its eight perturbations had evolved for the specified interval of 5, 10 or 20 generations, the perturbation directions were no longer guaranteed to remain mutually perpendicular. We arranged the circular differences between the reference state and its eight perturbed states as columns of a matrix, then applied reduced QR decomposition. This calculation produced Q, a refreshed set of mutually perpendicular perturbation directions, and R, whose diagonal values recorded the stretching or contraction along those directions. The logarithm of each absolute diagonal value contributed the expansion or contraction measured for that interval. We compared results across the renormalisation intervals and repeats listed above. Stability across these settings helped distinguish persistent expansion in several directions from numerical precision effects or one isolated unstable direction.

In Figure 4C, the upper diagram illustrates how evolving perturbations can become aligned and how QR decomposition restores mutually perpendicular directions before the next interval. The lower plot places the estimated exponents in order, allowing us to examine whether several directions showed expansion and whether independent repeats gave similar estimates. These examples illustrate how we read the diagnostics; the dynamical findings and their limitations are assessed in the Results.

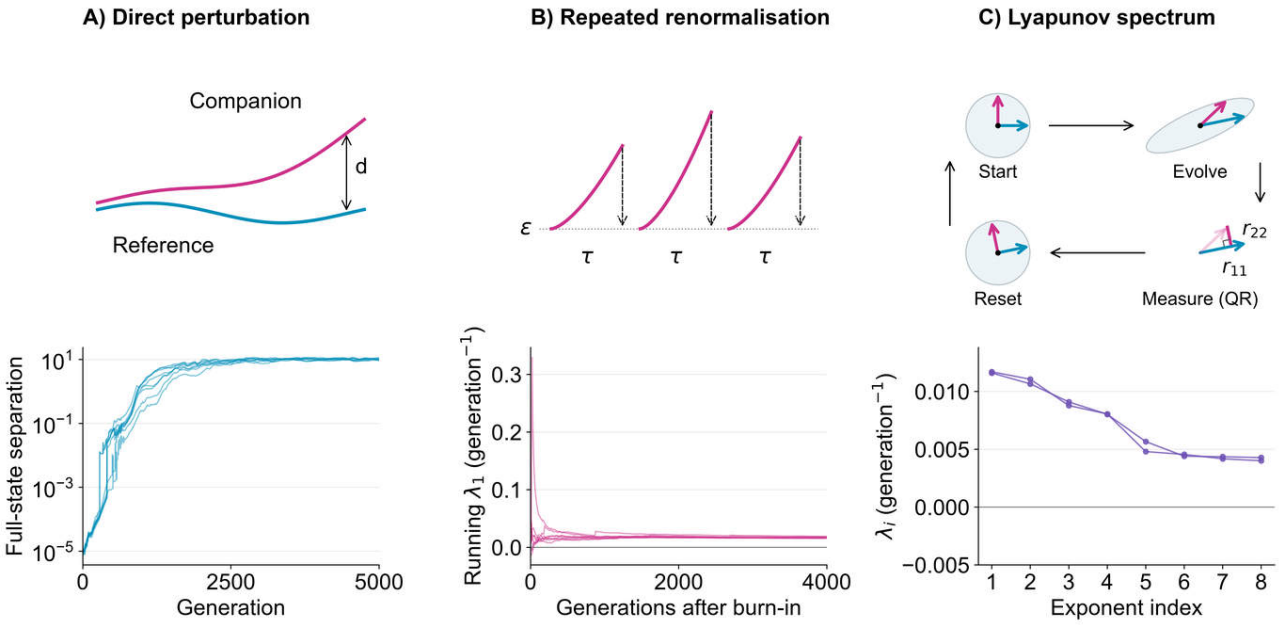


Figure 4. Measuring sensitivity to small changes in molecular states. Upper diagrams illustrate the procedures; lower plots use archived measurements for the Kuramoto–Sakaguchi system `OCNT-0494110` with CYP2C9 context. The molecular graph and trained weights remained fixed. **A)** A reference state and eight slightly perturbed companions evolve together; d denotes their full-state separation. Each initial perturbation had magnitude 10^{-5}. **B)** After each interval τ, we record the growth in separation and return the companion to distance ε while retaining its direction. The lower curves show eight running estimates of the largest Lyapunov exponent, using $\varepsilon = 10^{-5}$ and $\tau = 10$ generations. **C)** QR decomposition restores perpendicular perturbation directions and records their stretching or contraction. The schematic circle represents equal-sized starting perturbations, and the ellipse illustrates their deformation during evolution. QR measures the first vector's length, r_{11}, and the second vector's component perpendicular to the first, r_{22}; resetting restores equal-sized perpendicular vectors for the next interval. Two directions are drawn for clarity; the calculation followed eight and used float64 arithmetic. The lower curves show the eight leading exponents from two repeats, using $\varepsilon = 10^{-7}$ and $\tau = 10$ generations. Calculations in B and C used 1,000 burn-in generations followed by 4,000 measured generations. Separations use dimensionless model-state units; exponents give rates per generation.

2.5.7. Boundedness and attraction-basin test

An attractor is a set of computational states towards which nearby trajectories evolve. Its basin of attraction is the set of starting states whose trajectories approach that attractor. We tested for this behaviour by displacing the complete post-burn-in state by Euclidean distances of 0.1, 0.5, 1.0 and 2.0, then propagating the resulting float64 trajectories. For each of the two molecules, we generated eight displacements at each of the four distances. This produced 64 displaced trajectories in total. To check whether a trajectory escaped to progressively larger numerical values, we recorded the largest absolute state value that it reached. In the Kuramoto–Sakaguchi Graph-CA used for this test, every atom carried cyclic state channels called phases. Because each phase wraps between −1 and +1, we represented it by its sine and cosine before comparing the collections of computational states sampled by different trajectories. This circular embedding removes the artificial boundary between the two ends of the phase interval.

Directly comparing every phase value from every atom at once is difficult because each molecular state contains one coordinate for every atom-channel combination. We therefore examined the states from 32 randomly chosen one-dimensional directions. For each direction, every complete molecular state was reduced to one projected value, allowing us to compare the distributions sampled by the reference and displaced trajectories. For each time window, the reference distribution was formed from the projected states sampled along the unchanged reference trajectory. We calculated the distance from the displaced distribution to this reference distribution during both early and late portions of the trajectory. A late-to-early ratio below one meant that the displaced trajectory sampled states more similar to those of the reference trajectory at late times than at early times. We also measured the distance from each displaced state to its nearest sampled reference state. This supplied a complementary test of whether the displaced trajectory approached the set of computational states sampled by the reference trajectory.

2.5.8. Frozen-model structural interventions

We performed controlled computational interventions on the two candidates that advanced from the confirmatory dynamical analyses. These interventions tested whether changing particular structural inputs altered the sensitivity to initial conditions produced by the trained model. All trained weights, the transition rule, CYP context and Lyapunov-calculation settings remained fixed. Because messages travel in both directions across a chemical bond, each undirected bond was represented by two directed message-passing edges. Deleting a bond removed both directions.

For a bond-identity intervention, we retained the connection but replaced its encoding as single, double, triple or aromatic with each alternative type. These categories describe how atoms share electrons and whether the bond belongs to an aromatic electron system. A ring-opening intervention removed one selected connection within a molecular ring and also removed the atom feature stating that the affected atoms belonged to that ring. The atom-feature interventions removed one interpretable group at a time. The tested groups described elemental identity; charge and aromaticity; hybridisation; hydrogen-bond donor or acceptor state; chirality; or the chemical properties of neighbouring atoms. We also tested local valence, which describes an atom's bonding capacity and current bonding pattern.

The campaign estimated the largest Lyapunov exponent for 187 systems, comprising unmodified baselines and structurally modified versions, using a 1,000-generation burn-in followed by 2,000 measured generations, renormalisation every ten generations, perturbations of magnitude 10^{-7} and two repeats per system (Table 3). Effect size was the change in the mean largest Lyapunov exponent relative to the corresponding unmodified molecule-CYP system. A positive value meant that nearby trajectories separated faster after the intervention; a negative value meant that they separated more slowly. The workflow, numerical tables and publication figures are available through the repository described in Data Availability.

2.5.9. Extended analysis of the two strange-attractor candidates

We extended the analysis of candidates selected by the initial trajectory screening and direct perturbation tests. These candidates had subsequently undergone the largest-Lyapunov-exponent, Lyapunov-spectrum and attraction-basin tests described in Sections 2.5.5–2.5.7. For each candidate, we retained the frozen Graph-CA checkpoint that had generated its trajectory. We made no changes to the checkpoint's trained parameters or transition rule. The first extended test used

double-precision float64 trajectories with 5,000 burn-in generations followed by 20,000 measured generations. We propagated eight analytic tangent vectors alongside each trajectory. A tangent vector represents an infinitesimally small displacement and the local Jacobian describes how one update transforms that displacement. Every ten generations, QR decomposition restored eight mutually perpendicular directions. We checked the implementation in two ways. First, its one-step state update agreed with that of the original model, confirming that both implementations advanced the reference trajectory identically. Second, its analytic Jacobian-vector product agreed to within 10^{-12} with automatic differentiation, a software procedure that calculates derivatives by following the model's mathematical operations. This tested whether the analytic calculation correctly described how one update transformed an infinitesimal displacement.

We also estimated the largest exponent using finite rather than infinitesimal separations. The three repeatedly renormalised starting distances were 10^{-5}, 10^{-7} and 10^{-9}. We examined the estimates in four consecutive windows of 5,000 generations to test whether expansion persisted through time. Before calculating the complete Lyapunov spectrum, we allowed the tangent basis to evolve for 1,000 generations so that its directions could align with the local dynamics. We then measured the complete spectrum over a further 10,000 generations. Unlike the earlier leading spectrum, the complete spectrum included one exponent for every independent state-space direction. The sum of all Lyapunov exponents measures whether a small volume of nearby states grows or shrinks on average along a trajectory. We also calculated the Kaplan–Yorke estimate from the ordered Lyapunov exponents [31]. Here, dimension means the effective number of independent state-space directions occupied by the long-term dynamics. The estimate counts each direction up to the point at which cumulative expansion gives way to cumulative contraction, then adds a fractional contribution from the next contracting direction. Interpreting this estimate as a fractal dimension requires additional assumptions about the underlying dynamics.

For each reference trajectory, we also propagated four additional displaced states. Each used an independent direction and one of the radii 0.1, 0.5, 1.0 or 2.0. We represented every wrapped phase using its sine and cosine coordinates. We then examined the resulting complete states from 32 random one-dimensional projections and compared the distributions sampled by the reference and displaced trajectories. For each displaced trajectory, we measured the distance between its late distribution and the late distribution of the reference trajectory. We compared this value with the distance between two consecutive windows of the reference trajectory. Both reference windows contained 500 states sampled at intervals of ten generations. The reference's window-to-window distance provided a measure of sampling variation against which to assess displaced trajectories. Wrapped phases are bounded by construction, so the extended boundedness test served as a consistency check.

Finally, we tested the sensitivity of the correlation-dimension analysis. We sampled 2,001 states from each trajectory. From these states, we drew 100,000 random pairs. We then excluded pairs whose members occurred too close together in time. A *Theiler window* performs this exclusion: windows of 100, 500 and 1,000 generations prevented successive or near-successive states from dominating the geometric estimate. We repeated the calculation in projections containing 6, 12 and 24 components and in the complete circular embedding [14–16]. The correlation integral measures the fraction of state pairs within a given distance. Its scaling with distance can estimate fractal dimension when a stable scaling region exists. We retained local slopes and fitting-radius ranges to test this requirement.

We also examined the phase boundary, where the numerical representation wraps from one end of the phase interval to the other. We placed one state coordinate on opposite sides of this boundary at circular separations ranging from 2×10^{-4} to 2×10^{-10} and measured the separation after one update. For reproducibility and later figure preparation, we archived every reference atom-by-channel state, displaced trajectories sampled every ten generations, molecular bond connectivity, chemical features and diagnostic curves.

3. Results

We present the results in three parts. In Section 3.1, we report the scaffold-validation scores used to select models and their subsequent predictive performance on the reserved holdout set, which remained separate from training and model selection. In Section 3.2, we report the competition organisers' blind evaluation of our submitted predictions

against concealed experimental pIC50 measurements. In Section 3.3, we examine how the trained cellular automata behave over successive generations, including convergence, oscillation and sensitivity to small changes in initial states, and assess the evidence for strange attractors in molecular space-time.

3.1. Predictive performance on the reserved holdout set

We report the four-endpoint predictive performance on the reserved holdout set in Table 4 and describe the separate CYP3A4 scaffold-validation comparisons below. We report CYP3A4 separately because the evolutionary-strategy, hybrid genetic and genetic-programming searches trained and selected models specifically for this isoform; the four-endpoint GA and GP systems retained the existing BP models for the other three isoforms. The evaluation workflow links candidate screening, confirmation and reserved-holdout assessment for each method (Figure 5A–D). The reserved holdout set was kept separate from training and model selection. For BP-Graph-CA, 1,000 bootstrap resamples of the 1,309 reserved-holdout observations gave a mean MA-ST-RAE of 0.749064 and a 95% interval of 0.710791–0.789424. Individual CYP ST-RAE values were 0.816136 for CYP1A2, 0.713018 for CYP2C9, 0.936774 for CYP2D6 and 0.528032 for CYP3A4.

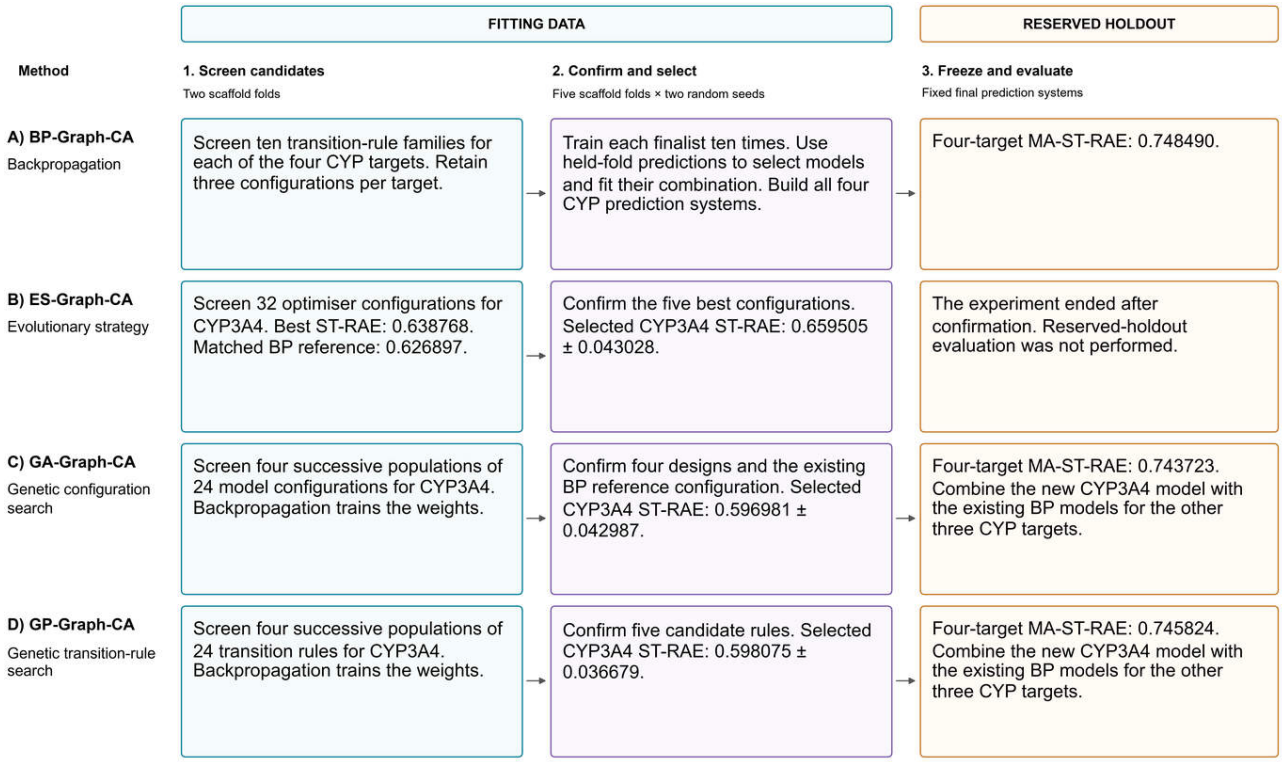


Figure 5. How the predictive scores were generated. Rows A–D follow BP-Graph-CA, ES-Graph-CA, GA-Graph-CA and GP-Graph-CA from two-fold screening through five-fold confirmation to reserved-holdout evaluation where performed. During five-fold validation, each training run uses four scaffold-separated folds for fitting and the remaining fold for scoring. Rotating the scoring fold through all five groups and repeating with two random seeds produces ten training runs per candidate. The confirmation scores shown for ES, GA and GP are means ± sample standard deviations across those ten runs. ST-RAE measures uncertainty-aware prediction error for one CYP target; MA-ST-RAE averages ST-RAE equally across the four CYP targets. The holdout values are scores calculated directly from predictions made after model selection and training were complete. GA and GP combine their selected CYP3A4 model with the existing BP prediction systems for the other three targets. ES concluded after confirmation.

The evolutionary-strategy campaign completed 114 training and scaffold-validation runs. Its best-performing training settings achieved CYP3A4 ST-RAE 0.638768 and RMSE 0.803659 pIC50 in the initial two-fold screen. Subsequent evaluation using five scaffold folds and two random seeds gave CYP3A4 ST-RAE 0.659505 ± 0.043028 and RMSE 0.817975 ± 0.042879 pIC50. For all three search methods reported here, confirmation values are means ± sample standard deviations across ten runs comprising five scaffold folds and two random seeds. We calculated each standard

deviation using a denominator of nine; it describes variation across these runs, rather than a confidence interval. Each evolutionary update tested 128 candidate sets of model weights, the numerical values that control how the cellular automaton processes information. The selected training settings used a perturbation scale of 0.02 to control the size of trial changes to those weights, an evolutionary learning rate of 0.01 to control the resulting weight adjustment, and batches of 256 molecules balanced by measured activity. Training continued for at most 80 updates, stopping earlier if validation ST-RAE failed to improve by at least 0.002 over 20 consecutive updates.

In the matched two-fold CYP3A4 screen, backpropagation training achieved ST-RAE 0.626897, compared with 0.638768 for evolutionary-strategy training. This comparison favoured backpropagation, and we ended the evolutionary-strategy experiment after scaffold validation, without advancing it to reserved-holdout or competition blind evaluation.

The hybrid genetic search selected the model-design settings specified in Section 2.4.4, including the transition-rule family, channel count and number of cellular automaton generations. Backpropagation trained the numerical weights within each candidate design. During model selection, scaffold validation using five folds and two random seeds gave CYP3A4 ST-RAE 0.596981 ± 0.042987 and RMSE 0.781116 ± 0.035890 pIC50. After selection, we evaluated the fixed CYP3A4 prediction system on the separate reserved-holdout molecules, obtaining CYP3A4 ST-RAE 0.508964. Combining this system with the existing BP systems for the other three isoforms gave GA-Graph-CA the lowest four-endpoint reserved-holdout MA-ST-RAE in Table 4.

Genetic programming selected a cellular automaton transition rule that averages the atom's local-input vector and the difference between its bonded neighbours' average state and its own state (Section 2.4.5). Genetic programming searched the mathematical form of the rule, while backpropagation trained the numerical weights within each candidate model. During model selection, scaffold validation using five folds and two random seeds gave CYP3A4 ST-RAE 0.598075 ± 0.036679 and RMSE 0.772706 ± 0.032415 pIC50, the lowest confirmation RMSE among the genetic searches. After selection, we evaluated the fixed CYP3A4 prediction system on the separate reserved-holdout molecules, obtaining CYP3A4 ST-RAE 0.517368. Combining this system with the existing BP systems for the other three isoforms gave a four-endpoint reserved-holdout MA-ST-RAE between the fixed-rule and genetic-structure results (Table 4). Across 1,000 bootstrap samples of the reserved-holdout observations, mean MA-ST-RAE was 0.746602, with a 95% interval of 0.708890–0.788464.

Method	Local-rule treatment	Reserved-holdout MA-ST-RAE	Reserved-holdout RMSE, pIC50
BP-Graph-CA	Fixed rule families; model weights trained by backpropagation	0.748490	0.847738
ES-Graph-CA	Fixed rule and model design; model weights trained by evolutionary strategy	Not evaluated	Not evaluated
GA-Graph-CA	Genetic algorithm selects model-design settings; backpropagation trains model weights	**0.743723**	**0.841572**
GP-Graph-CA	Genetic programming evolves the cellular automaton transition rule; backpropagation trains model weights	0.745824	0.848265

Table 4. Predictive performance of Graph-CA learning strategies on the reserved holdout set. Values describe the four-endpoint ensembles evaluated on data excluded from training and model selection. MA-ST-RAE averages ST-RAE across the four CYP isoforms. The displayed scores were calculated directly from the reserved-holdout predictions; the text reports bootstrap means and intervals separately. ES-Graph-CA concluded after scaffold validation and did not undergo reserved-holdout evaluation.

GA-Graph-CA achieved the lowest reserved-holdout MA-ST-RAE, while GP-Graph-CA achieved the strongest competition blind results reported in Section 3.2. The matched two-fold comparison specifically favoured backpropagation over evolutionary-strategy training for CYP3A4.

3.2. Blind Challenge Evaluation

The organisers evaluated the fixed submissions against concealed experimental measurements (Table 5). Each submission supplied four CYP predictions for each of 750 blinded molecules.

Official blind metric	Fixed-rule BP-Graph-CA	Genetic algorithm GA-Graph-CA	Genetic programming GP-Graph-CA
MA-ST-RAE	1.0092	1.0020	**0.9946**
Macro MAE, pIC50	1.0789	1.0745	**1.0688**
Macro R-squared	-0.0744	-0.0724	**-0.0656**
Macro Spearman rho	0.5345	0.5444	**0.5498**
Macro Kendall tau	0.3762	0.3840	**0.3878**

Table 5. Official blind OpenADMET CYP Inhibition Challenge results for the three submitted Graph-CA variants. Lower values are favourable for MA-ST-RAE and macro MAE; higher values are favourable for macro R-squared, Spearman rho, and Kendall tau.

Genetic programming produced the strongest blind result for every reported measure. Compared with the fixed-rule reference, its MA-ST-RAE decreased by 0.0146 and its macro mean absolute error (MAE) decreased by 0.0101 pIC50. Its Spearman rank correlation increased by 0.0153, and its Kendall rank correlation increased by 0.0116. The hybrid genetic structure search lay between the other two systems on all five measures. The improvement was modest but occurred consistently in both the error and rank-correlation measures.

3.3. Nonlinear Dynamics in Molecular Space-Time

We examined how the trained cellular automata's numerical atom states changed over successive generations, looking for convergence to fixed values, repeated oscillations and persistent complex variation. We first screened trajectories across all ten transition-rule families, then tested selected candidates for sustained sensitivity to small changes in initial states and evidence of strange-attractor geometry.

3.3.1. Population-level dynamical screen

Each of the ten production models retained summaries of 1,309 validation trajectories, yielding 13,090 short-horizon screens (Table 6). Point-attractor candidates required late motion below 10^{-4} and a final state change below 10^{-5}. Oscillator candidates required a recurrence ratio below 0.25 and spectral concentration above 0.5, indicating close returns and power concentrated at a small number of frequencies. Remaining trajectories entered the persistent or complex category.

Cellular automaton transition rule	Screened	Point-attractor candidates	Oscillator candidates	Persistent or complex	Analysed hyperchaotic candidates
Gated residual	1,309	0	0	1,309	0
Delayed memory	1,309	0	0	1,309	0
Inertial reaction–diffusion	1,309	0	0	1,309	0
Gray–Scott	1,309	0	0	1,309	0
Coupled map	1,309	0	1,309	0	0
Activator–inhibitor	1,309	0	0	1,309	0
FitzHugh–Nagumo	1,309	0	0	1,309	0
Kuramoto–Sakaguchi	1,309	0	0	1,309	2
Damped symplectic	1,309	0	0	1,309	0
Conservative graph flux	1,309	1,309	0	0	0
Total	**13,090**	**1,309**	**1,309**	**10,472**	**2**

Table 6. Population-level classification screen for frozen Graph-CA trajectories across all ten transition-rule families. Analysed hyperchaotic candidates form a tested subset of the persistent or complex screen.

The two tested Kuramoto–Sakaguchi candidates came from the persistent or complex class and showed sustained expansion of small differences. All 1,309 coupled-map trajectories met the screening criteria for oscillation, with longer periodicity tests required for confirmation. To illustrate how these behaviours develop over longer trajectories, we followed selected examples over thousands of cellular automaton generations. The examples show numerical atom states settling towards fixed values, continuing to vary within a bounded region or alternating between two states; they include a hyperchaotic candidate and a trajectory with persistent complex variation that requires further testing (Figure 6).

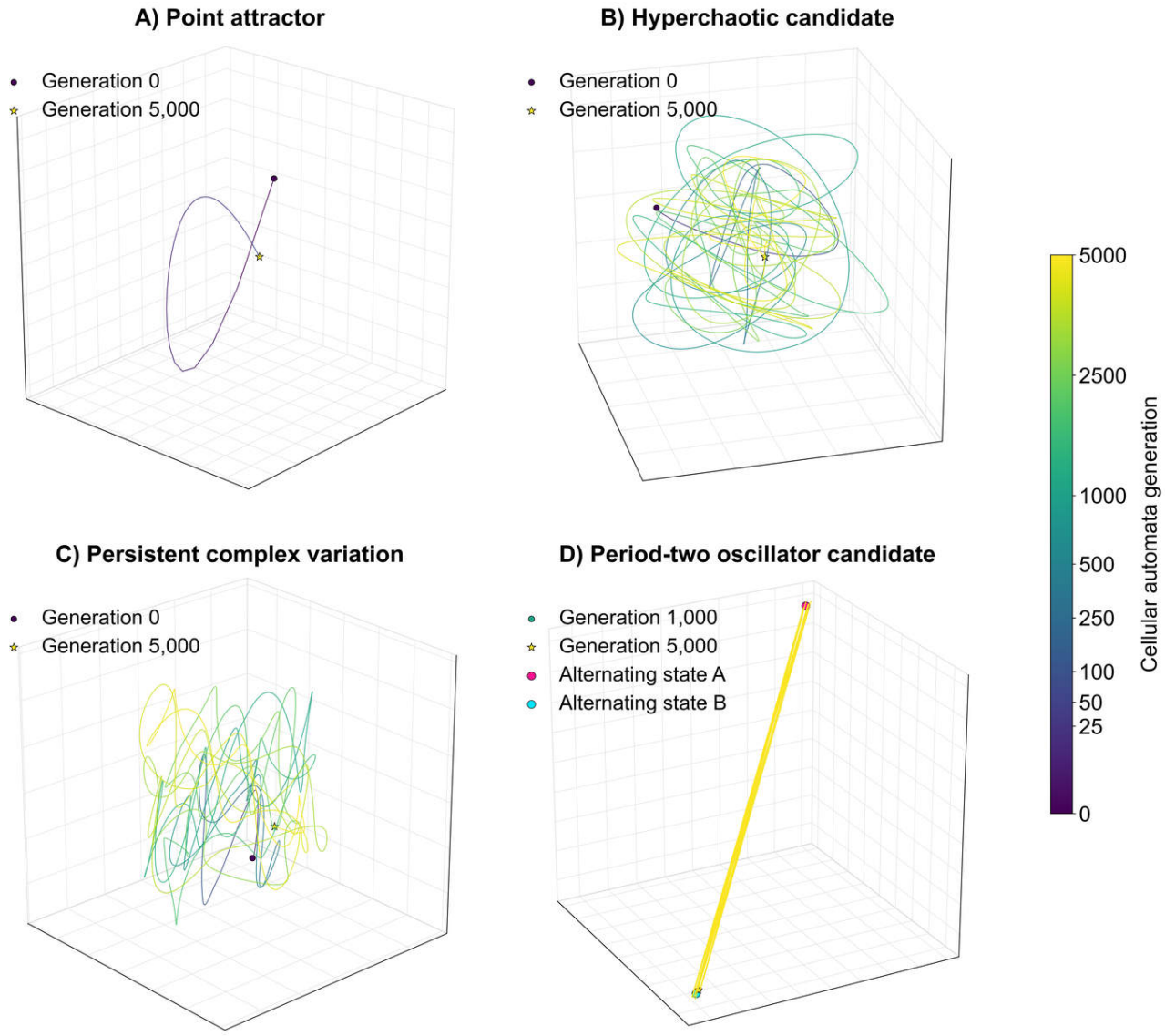


Figure 6. Four examples of molecular Graph-CA dynamics. A) Gated residual, `OCNT-2328519`, CYP1A2: convergence towards a fixed state. **B)** Kuramoto–Sakaguchi, `OCNT-0494110`, CYP2C9: a bounded hyperchaotic candidate. **C)** Kuramoto–Sakaguchi, `OCNT-0495493`, CYP2D6: persistent complex variation requiring further dynamical testing. **D)** Coupled map, `OCNT-0495275`, CYP3A4: a candidate period-two oscillation, with magenta and cyan markers identifying the alternating states. Panels A–C show three-dimensional principal-component projections of the numerical atom states over generations 0–5,000; circular phase states in B and C were represented by their sine and cosine before projection. Panel D shows generations 1,000–5,000 using the leading state component and its values one and two generations earlier. The axes describe numerical state coordinates, rather than physical molecular positions. Colour indicates generation using the nonlinear scale shown; circles and stars mark the first and last displayed generations.

3.3.2. Extended strange-attractor analysis

Following the screening described in Section 3.3.1, we examined the two selected Kuramoto–Sakaguchi candidates in greater detail to assess whether their persistent variation provided evidence of chaotic dynamics and strange-attractor geometry. We repeatedly introduced a small displacement between two states, measured how much it grew, and returned it to its original size before continuing. All 48 repeated estimates of the largest Lyapunov exponent were positive across initial displacement sizes from 10^{-6} to 10^{-4} (Figure 7A). The first eight Lyapunov exponents were also positive for both trajectories, indicating that nearby states separated along several independent directions (Figure 7B). To check whether the timing of the rescaling affected this result, we repeated the calculation with intervals of 5, 10 and 20 cellular automaton generations. The largest exponent remained positive and similar in magnitude across these intervals (Figure 7C).

We also tested how the trajectories responded to larger changes in their initial states, using four perturbation radii. All 64 repeated tests remained bounded. For each test, we compared the distance between the displaced and reference state distributions during early and late portions of the trajectories. The late-to-early distance ratios were below one, showing that these distributions became closer over time (Figure 7D). Together, these initial tests support sustained separation along several independent directions while displaced trajectories approach similar sampled distributions. The extended tests below examine the persistence of these findings and the additional evidence required to establish fractal strange-attractor geometry.

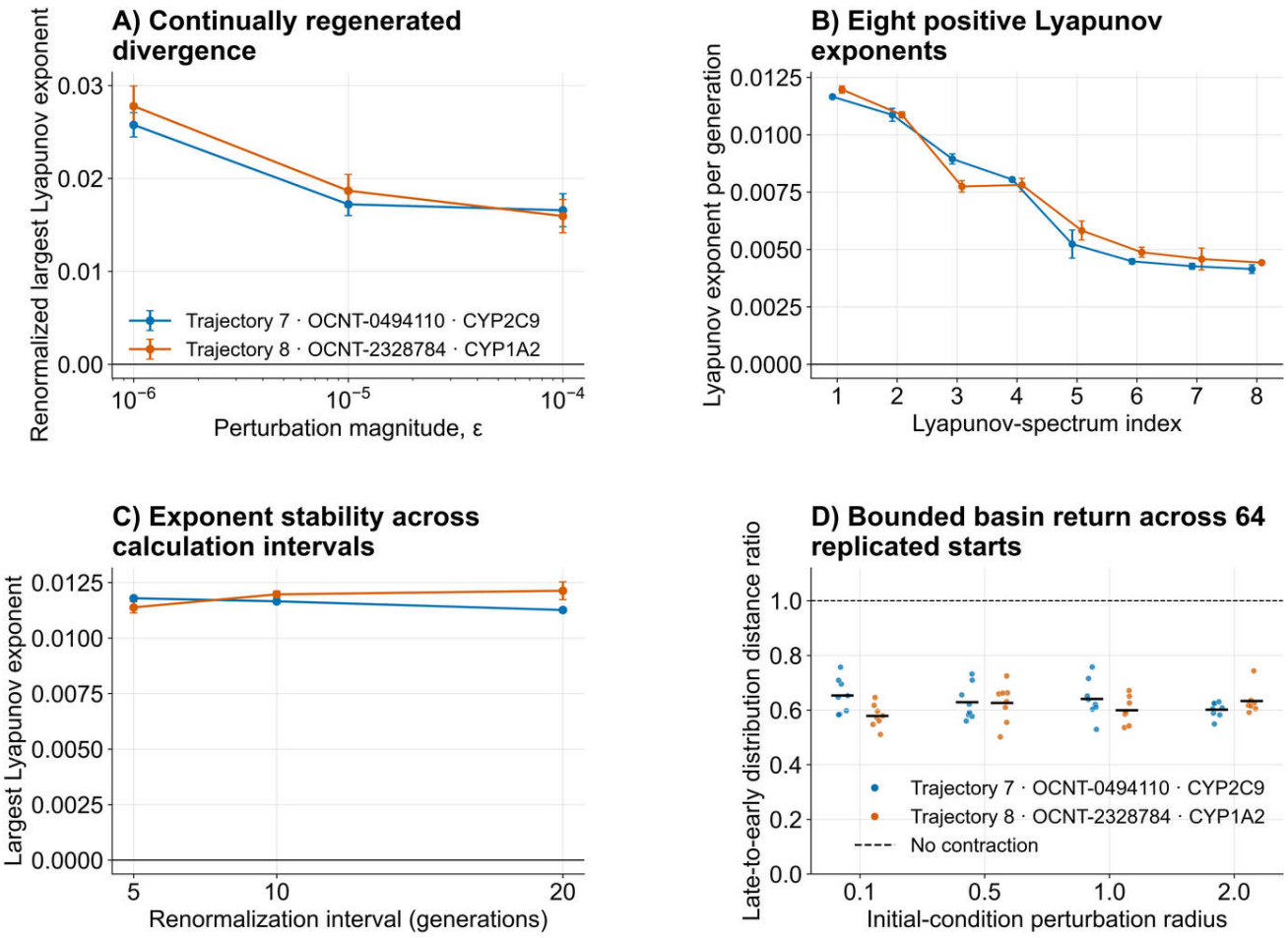


Figure 7. Initial dynamical tests of the two Kuramoto–Sakaguchi candidates. A) Largest Lyapunov exponent across initial displacement sizes. **B)** First eight Lyapunov exponents. **C)** Largest Lyapunov exponent across rescaling intervals. **D)** Late-to-early distribution-distance ratios across initial perturbation radii. Blue denotes trajectory 7, `OCNT-0494110`, CYP2C9; orange denotes trajectory 8, `OCNT-2328784`, CYP1A2. Error bars in A–C show standard deviations across repeated calculations. Horizontal black lines in A–C mark zero divergence, and short black bars in D mark mean ratios. In D, the dashed line labelled "No contraction" marks a ratio of 1, meaning that the displaced trajectory's distribution is equally far from the reference distribution at early and late times. Ratios below 1 indicate that these distributions became closer over time.

Building on these initial findings, we extended the analysis of `OCNT-2328784` with CYP1A2 context and `OCNT-0494110` with CYP2C9 context (Table 7). They corresponded to trajectories 8 and 7, respectively, in the original 20-trajectory analysis and were also selected for the frozen-model structural interventions described in Section 2.5.8. Both used the frozen Kuramoto–Sakaguchi checkpoint and retained eight positive leading tangent exponents during 20,000 measured generations following 5,000 burn-in generations. Largest-exponent estimates in successive 5,000-generation windows ranged from 0.011870 to 0.012341 per generation for `OCNT-2328784` and from 0.012169 to 0.013314 for `OCNT-0494110`. Finite-separation tests at 10^{-7} and 10^{-9} gave estimates of 0.012109 and 0.012704 for the respective systems, supporting persistence beyond the original analysis.

Quantity	OCNT-2328784, CYP1A2	OCNT-0494110, CYP2C9
Atom-by-channel state dimension	240	320
Largest leading exponent, per generation	0.012115	0.012641
Eighth leading exponent, per generation	0.004376	0.004421
Positive exponents in complete spectrum	29	30
Sum of complete spectrum, per generation	−3.606412	−5.144542
Kaplan–Yorke heuristic	68.19	73.91

Table 7. Extended float64 analysis of the two Kuramoto–Sakaguchi candidates. Leading exponents use the 20,000-generation analysis; complete Lyapunov-spectrum quantities use a subsequent 10,000-generation interval. Positive counts and Kaplan–Yorke estimates are finite-time numerical quantities.

The sum of each complete spectrum was negative. A small volume formed by nearby states therefore shrank on average as it moved along the sampled trajectory. At the same time, the individual positive exponents show that the states separated along several directions within that shrinking volume. The largest exponents in the complete spectra were 0.012118 and 0.012480, respectively. For the second system, the number of positive exponents changed from 28 to 30 between the two halves of the 10,000-generation measurement. This variation shows that some finite-time estimates remained close to zero and were not fully stable.

We also followed eight newly displaced trajectories. Their late distribution distances were 0.843–1.036 times the distance between two consecutive windows of the unperturbed reference. At this sampling resolution, the displaced trajectories approached distributions that differed from the reference by about as much as the reference differed from itself between successive time windows.

We examined two further questions before deciding whether these trajectories provide evidence of strange attractors: whether the transition rule behaves consistently where the numerical phase interval ends, and whether the trajectories have a measurable fractal geometry. The first question arises because a phase represents a position around a circle. The two ends of the numerical interval used to store that phase meet at the same point on the circle, just as the beginning and end of a clock face meet (Figure 8A). Here, h is the phase angle divided by π, and ε denotes a small offset from either end of the interval; the two marked states therefore have stored values of 1 − ε and −1 + ε. Their spacing in the schematic is exaggerated for visibility. We measured separation as a dimensionless circular state-space distance: for each atom-state channel, we calculated the shortest angular difference in radians and divided it by π, so a difference of 1 in a single channel represents half a turn. We then squared these differences, summed them across all atoms and channels, and took the square root to obtain the overall separation. We placed two initial states on opposite sides of the boundary in one channel, giving an overall separation of approximately 2×10^{-10} on this scale. After one cellular automaton update, their overall separations increased to 0.005422 and 0.005871 in the two systems, respectively. Across the tested initial separations, from approximately 2×10^{-10} to 2×10^{-4}, the separation after one update remained appreciable even as the initial separation became extremely small (Figure 8B). The learned calculation of how far each phase advances uses its stored numerical value, so phases that lie almost together on the circle can receive different updates when their stored values lie at opposite ends of the interval. This abrupt change at the boundary introduces a source of separation that complicates the interpretation of sensitivity to initial conditions as evidence of a smooth chaotic process.

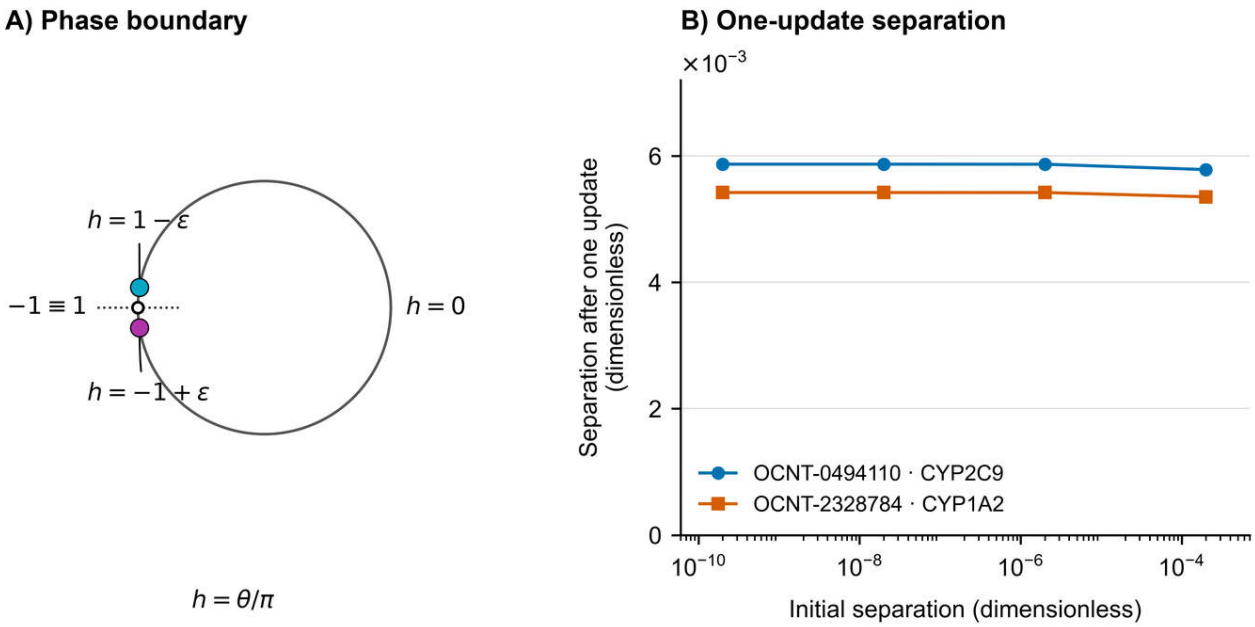


Figure 8. Phase-boundary test of the Kuramoto–Sakaguchi transition rule. A) Two nearby phases on opposite sides of the numerical boundary, with spacing exaggerated. **B)** Recorded separation after one update for the two analysed molecules. Distances are dimensionless; lines connect the measured points.

The second question concerns the geometry traced out by the system's successive states. We estimated the correlation dimension, which describes how the number of nearby pairs of states grows as we increase the distance used to count them. A reliable estimate requires a consistent relationship over a sufficiently broad range of distances, together with stability as the representation captures more of the system's state. We first represented each trajectory using six projected coordinates, which summarise the full state in fewer numbers, and then used the complete circular representation. The slopes used to estimate the dimension increased from approximately 4.6–4.8 to approximately 27–29 between these representations. Each fitted distance range covered less than 0.3 logarithmic decades, equivalent to less than a twofold increase in distance. These narrow ranges and the large changes in slope leave the fractal dimension unresolved. Taken together, our results support bounded trajectories in which small differences grow along several independent directions and displaced states later sample similar distributions. Establishing a strange attractor requires further evidence of a robust fractal dimension, alongside an assessment of how the phase-boundary discontinuity affects the dynamical interpretation [14–16]. The Data Availability section identifies the archived states and diagnostic results supporting this analysis.

To relate the trajectory displays to individual atoms, we also visualised the four examples from Figure 6 as molecular information cascades, with colour representing each atom's numerical state and vertical position representing successive generations. The gated-residual example, `OCNT-2328519` with CYP1A2 context, illustrates convergence towards a fixed state (Figure 9A). The Kuramoto–Sakaguchi example, `OCNT-0494110` with CYP2C9 context, shows the atom-state trajectory of the hyperchaotic candidate examined above (Figure 9B). A second Kuramoto–Sakaguchi example, `OCNT-0495493` with CYP2D6 context, illustrates persistent complex variation that requires further dynamical testing (Figure 9C). The coupled-map example, `OCNT-0495275` with CYP3A4 context, illustrates the candidate period-two oscillation (Figure 9D). These classifications draw on the dynamical tests described above; the cascades show how the corresponding numerical states develop across individual atoms. To construct each display, we rotated the three-dimensional molecule slightly at each generation, moved it downwards and retained its earlier atom positions, producing continuous atom ribbons.

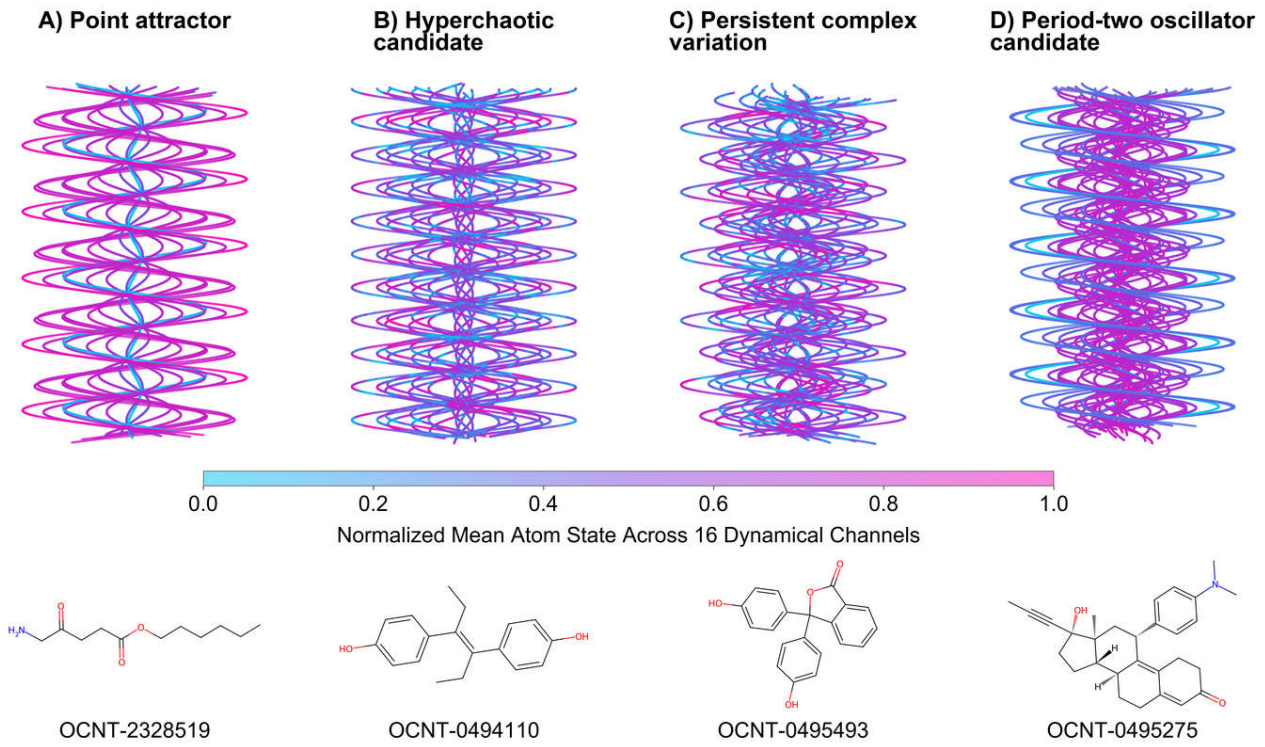


Figure 9. Molecular information cascades generated by the four Graph-CA trajectories in Figure 6. A) Point attractor. **B)** Hyperchaotic candidate. **C)** Persistent complex variation. **D)** Period-two oscillator candidate. Each panel shows the complete cascade retained at generation 5,000, with the corresponding two-dimensional molecular structure and identifier below it. Colour from cyan to magenta records the mean state of each atom across the 16 dynamical channels at that generation, after robust normalisation within the trajectory. Vertical position marks computational generations. Rotation serves solely as a visual aid, showing the three-dimensional molecule from different angles so that more atoms and their traces remain visible. Bonds are omitted from the cascades so that the accumulated atom-state structure remains visible; the molecular structures below were drawn using RDKit.

4. Discussion

To illustrate the aesthetic beauty of molecular space-time, Appendix E presents a ten-page visual library spanning all ten transition-rule families, pairing flat atom-by-generation displays with rotating molecular traces. Each page identifies its fixed-rule backpropagation training provenance and shows how the recorded atom states produce different visual patterns across molecular structures and CYP contexts.

4.1. Emergence from Local Molecular Interactions

Our operational test of emergence asks whether repeated interactions between bonded atomic cells generate a molecule-wide representation containing useful predictive information [32, 33]. All three submitted systems showed moderate agreement between predicted and experimental rankings on the blind challenge. This supports the usefulness of their trajectory fingerprints. A matched comparison with predictions based directly on the starting atom and bond descriptors would establish how much predictive information the repeated interactions add.

Blind performance improved from fixed-rule backpropagation to genetic structure search and then genetic programming, with GP-Graph-CA achieving the strongest result on all five official measures. This ordering suggests that the choice of local transition rule influences the information available to the molecular readout. The selected expression averages the combined local-input vector with the bonded-neighbour state difference, linking local processing to exchange across the graph. Its modest gain supports further investigation of this coupling under the present prediction task and training procedure.

4.2. Molecular Space-Time as a Dynamical Object

The numerical atom states provide coordinates for studying the trained model's behaviour. Their meaning comes from their effects on neighbour messages, subsequent states and predictions, rather than a direct correspondence with physical positions or measured chemical quantities. Such learned coordinates can collectively carry measurable information [34].

The long-horizon experiments identified convergence, candidate period-two oscillation and persistent complex variation. In two frozen Kuramoto–Sakaguchi systems, nearby states separated along several independent directions while small local volumes contracted on average. Displaced trajectories also approached similar sampled distributions. Together, the perturbation, Lyapunov-spectrum and attraction-basin tests support bounded hyperchaotic behaviour over the measured intervals [11–13]. These findings establish molecular space-time as an object of quantitative dynamical analysis. The larger persistent or complex group remains a screening category, whose members require individual testing. For the two detailed candidates, the estimated fractal dimension varied with the representation of the trajectory, and the transition rule changed abruptly at the phase boundary. Establishing a fractal strange attractor therefore requires stable dimension estimates and further assessment of this boundary effect.

4.3. Implications for Complex Systems and Drug Discovery

Molecular graphs provide chemically defined local connections, while measured properties supply an external test of the information generated through those connections. The same framework could be evaluated on solubility, permeability, binding and reactivity, with each task testing whether trajectory summaries capture useful information about the whole molecule.

Treating the representation as a dynamical system also allows comparisons between molecular structure and computational behaviour. Controlled changes to bonds or atomic descriptors, with trained weights held fixed, can test which inputs influence convergence, repetition or sensitivity to initial conditions. Relating these effects to measured chemical properties could provide interpretable trajectory-based descriptors, subject to prospective validation on new datasets and independently trained models.

The optimisation comparisons suggest complementary roles for gradient-based training and genetic search. Backpropagation achieved the stronger score on the matched evolutionary-strategy comparison, and the genetic methods improved blind performance while retaining backpropagation within each candidate. Gradients adjusted numerical weights, while population searches explored model configurations and mathematical transition rules. This division of tasks warrants further testing across datasets.

4.4. Limitations and Future Tests

Our primary aim was to investigate molecular space-time, using the CYP challenge as a testbed for studying how local interactions generate predictive information and dynamical behaviour. The predictive comparison provides a practical assessment of this representation, with state-of-the-art CYP prediction remaining a broader objective beyond the scope of the present study. The predictive results establish the performance of this approach within a dataset of 4,905 labelled compounds and four related CYP tasks. Blind MA-ST-RAE was close to the mean-prediction baseline, and macro R-squared was negative, indicating limited absolute predictive accuracy. Genetic programming produced a modest improvement, whose statistical reliability remains uncertain. The submitted system used an evolved transition rule for CYP3A4 and fixed-rule ensembles for the other targets. The extent to which this improvement generalises to other prediction tasks therefore remains open; evolving rules for additional targets or examining transfer between tasks could help clarify its scope.

The trajectory fingerprints carried useful predictive information, although the contribution of repeated local interactions beyond the starting chemical descriptors remains to be quantified. A matched descriptor-only comparison could address this by summarising the initial atom and bond descriptors directly into a fingerprint and fitting the same

readout, with identical data partitions, CYP context and comparable model-selection effort. Comparisons that alter the order of trajectory states or communication across bonds could also help explain how temporal organisation and local interactions contribute to prediction. These are possible extensions of the present analysis.

The dynamical results support bounded hyperchaotic behaviour over the measured intervals, while the fractal geometry remains unresolved. The estimated dimension varied with the trajectory representation and measurement scale. In addition, the learned phase rule can change abruptly where the numerical interval wraps, even though states on either side represent neighbouring phases. This boundary effect qualifies interpretations that assume smooth motion on a circle, including the use of Kaplan–Yorke estimates as fractal dimensions. The finite trajectory lengths, sampled initial states and numerical precision also define the scope of the evidence. Longer observations and comparisons across independently trained models could clarify the persistence and generality of these behaviours. Measures of entropy rate, recurrent-state relationships and computational structure offer further ways to characterise the trajectories [13–16].

5. Conclusions

Local interactions between chemically encoded atomic cells generated molecular space-time fingerprints with measurable predictive information for CYP inhibition. Blind MA-ST-RAE improved from 1.0092 for fixed-rule backpropagation to 1.0020 for genetic structure search and 0.9946 for genetic programming. The same ordering held across the other official metrics, although absolute accuracy remained limited.

Retained trajectories also supported direct dynamical analysis, revealing convergence, possible periodic behaviour and persistent complex variation. Two systems exhibited bounded hyperchaotic dynamics with expansion in multiple directions and average local-volume contraction. Phase-boundary discontinuities and unresolved fractal dimension keep their classification as strange attractors provisional.

The framework connects chemical graph structure, local computation and molecule-wide behaviour. Matched predictive controls, independently evolved and transferred rules, and reproducible dynamical classifications across molecular classes remain the principal next tests.

Author Contributions

A.N. conceived the study, developed and implemented the computational methodology, conducted the experiments, analysed and interpreted the results, prepared the figures, and wrote and revised the manuscript. A.N. used AI to prepare the code.

Funding

This research received no external funding.

Data Availability

The 2026 OpenADMET CYP Inhibition Challenge provided the dataset that we analysed in this study; the challenge organiser's access conditions govern its availability.

The research data are available on the main branch of the repository at https://github.com/acnash/Strange-Matter-Engine/tree/main. The file results/long_horizon_attractor_campaign_v1/validation_dynamics_population_summary.csv provides a machine-readable version of the screening counts in Table 6, together with the number of candidates tested

for strange-attractor behaviour. The two analysed hyperchaotic candidates retain an unresolved fractal dimension, as described in Section 3.3.2.

The directory results/strange_attractor_audit_20260929 contains the complete atom-state trajectories for the two Kuramoto–Sakaguchi candidates analysed in Section 3.3.2, displaced trajectories, molecular graph information, Lyapunov-spectrum arrays, correlation-integral data and numerical summaries. Its README describes the array contents and analysis limitations; provenance records and source-code snapshots document how the results were generated.

The directory results/structure_dynamics_publication_v1 contains the structural-intervention workflow description, numerical tables and publication figures. Its README documents the campaign and its supporting files.

Code Availability

Source code, trained model configurations and scripts required to reproduce the reported analyses are available at https://github.com/acnash/Strange-Matter-Engine.git

References

1. Wolfram S. Statistical mechanics of cellular automata. *Reviews of Modern Physics*. 1983;55(3):601–644. doi: 10.1103/RevModPhys.55.601.
2. Wolfram S. Universality and complexity in cellular automata. *Physica D: Nonlinear Phenomena*. 1984;10(1–2):1–35. doi: 10.1016/0167-2789(84)90245-8.
3. Langton CG. Computation at the edge of chaos: phase transitions and emergent computation. *Physica D: Nonlinear Phenomena*. 1990;42(1–3):12–37. doi: 10.1016/0167-2789(90)90064-V.
4. Turing AM. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society B*. 1952;237(641):37–72. doi: 10.1098/rstb.1952.0012.
5. Grattarola D, Livi L, Alippi C. Learning graph cellular automata. *Advances in Neural Information Processing Systems*. 2021;34:20983–20994. https://proceedings.neurips.cc/paper/2021/hash/af87f7cdcda223c41c3f3ef05a3aaeea-Abstract.html.
6. Mordvintsev A, Randazzo E, Niklasson E, Levin M. Growing neural cellular automata. *Distill*. 2020;5(2). doi: 10.23915/distill.00023.
7. Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE. Neural message passing for quantum chemistry. *Proceedings of Machine Learning Research*. 2017;70:1263–1272. https://proceedings.mlr.press/v70/gilmer17a.html.
8. Goldberg DE. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley; 1989.
9. Koza JR. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press; 1992.
10. Zanger UM, Schwab M. Cytochrome P450 enzymes in drug metabolism: regulation of gene expression, enzyme activities, and impact of genetic variation. *Pharmacology & Therapeutics*. 2013;138(1):103–141. doi: 10.1016/j.pharmthera.2012.12.007.
11. Benettin G, Galgani L, Giorgilli A, Strelcyn JM. Lyapunov characteristic exponents for smooth dynamical systems and for Hamiltonian systems; a method for computing all of them. Part 1: theory. *Meccanica*. 1980;15:9–20. doi: 10.1007/BF02128236.

12. Oseledec VI. A multiplicative ergodic theorem: Lyapunov characteristic numbers for dynamical systems. *Transactions of the Moscow Mathematical Society*. 1968;19:197–231.

13. Crutchfield JP, Young K. Inferring statistical complexity. *Physical Review Letters*. 1989;63(2):105–108. doi: 10.1103/PhysRevLett.63.105.

14. Grassberger P, Procaccia I. Measuring the strangeness of strange attractors. *Physica D*. 1983;9:189–208. doi: 10.1016/0167-2789(83)90298-1.

15. Theiler J. Spurious dimension from correlation algorithms applied to limited time-series data. *Physical Review A*. 1986;34:2427–2432. doi: 10.1103/PhysRevA.34.2427.

16. Eckmann J-P, Ruelle D. Fundamental limitations for estimating dimensions and Lyapunov exponents in dynamical systems. *Physica D*. 1992;56:185–187. doi: 10.1016/0167-2789(92)90023-G.

17. Cho K, van Merriënboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. 2014:1724–1734. doi: 10.3115/v1/D14-1179.

18. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Computation*. 1997;9(8):1735–1780. doi: 10.1162/neco.1997.9.8.1735.

19. Randazzo E, Mordvintsev A, Niklasson E, Levin M, Greydanus S. Self-classifying MNIST digits. *Distill*. 2020;5(8):e00027.002. doi: 10.23915/distill.00027.002.

20. Polyak BT. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*. 1964;4(5):1–17. doi: 10.1016/0041-5553(64)90137-5.

21. FitzHugh R. Impulses and physiological states in theoretical models of nerve membrane. *Biophysical Journal*. 1961;1(6):445–466. doi: 10.1016/S0006-3495(61)86902-6.

22. Nagumo J, Arimoto S, Yoshizawa S. An active pulse transmission line simulating nerve axon. *Proceedings of the IRE*. 1962;50(10):2061–2070. doi: 10.1109/JRPROC.1962.288235.

23. Kuramoto Y. *Chemical Oscillations, Waves, and Turbulence*. Berlin: Springer; 1984. doi: 10.1007/978-3-642-69689-3.

24. Sakaguchi H, Kuramoto Y. A soluble active rotator model showing phase transitions via mutual entrainment. *Progress of Theoretical Physics*. 1986;76(3):576–581. doi: 10.1143/PTP.76.576.

25. Hale JK, Verduyn Lunel SM. *Introduction to Functional Differential Equations*. New York: Springer; 1993. doi: 10.1007/978-1-4612-4342-7.

26. Hoerl AE, Kennard RW. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*. 1970;12(1):55–67. doi: 10.1080/00401706.1970.10488634.

27. Wolf A, Swift JB, Swinney HL, Vastano JA. Determining Lyapunov exponents from a time series. *Physica D: Nonlinear Phenomena*. 1985;16(3):285–317. doi: 10.1016/0167-2789(85)90011-9.

28. Eckmann J-P, Oliffson Kamphorst S, Ruelle D. Recurrence plots of dynamical systems. *Europhysics Letters*. 1987;4(9):973–977.

29. Miranda RA, Costa SGSP, Alves RA, Ferreira JL. Spectral entropy of plasma oscillations in numerical simulations of a Hall thruster during start-up transients. *Journal of Electric Propulsion*. 2026;5:7. doi: 10.1007/s44205-026-00180-2.

30. Benettin G, Galgani L, Giorgilli A, Strelcyn JM. Lyapunov characteristic exponents for smooth dynamical systems and for Hamiltonian systems; a method for computing all of them. Part 2: numerical application. *Meccanica*. 1980;15(1):21–30. doi: 10.1007/BF02128237.

31. Frederickson P, Kaplan JL, Yorke ED, Yorke JA. The Liapunov dimension of strange attractors. *Journal of Differential Equations*. 1983;49(2):185–207. doi: 10.1016/0022-0396(83)90011-6.

32. Gershenson C. Emergence in Artificial Life. *Artificial Life*. 2023;29(2):153–167. doi: 10.1162/artl_a_00397.

33. Bedau MA. Artificial life: organization, adaptation and complexity from the bottom up. *Trends in Cognitive Sciences*. 2003;7(11):505–512. doi: 10.1016/j.tics.2003.09.012.

34. Shannon CE. A mathematical theory of communication. *Bell System Technical Journal*. 1948;27:379–423, 623–656. doi: 10.1002/j.1538-7305.1948.tb01338.x; 10.1002/j.1538-7305.1948.tb00917.x.

Appendix A. Mathematical Definition of the Graph Cellular Automaton

This appendix explains the shared Graph-CA calculations and all ten transition rules introduced in Section 2.3. It gives equations for the rules in Figure 2A-E, followed by implementation details for the rules in Figure 2F-J. All descriptions refer to the numerical updates used in this study.

For atom i, x_i contains the chemical descriptors defined in Section 2.2. A shared trainable transformation converts them to H initial channel values (Figure 1B):

$$h_i^{(0)} = \tanh\left(s_0 W_{\text{init}} x_i\right),$$

Each row of the trainable matrix W_{init} forms a weighted sum of the F descriptors, producing one of the H channels. The superscript (0) denotes generation zero, so $h_i^{(0)} \in \mathbb{R}^H$ is the initial H-value state. The initialisation scale s_0 controls the sums before tanh bounds them between −1 and 1. Large inputs approach these limits, where further input changes have progressively less effect, termed saturation.

We applied the shared transition rule T times, with the same parameters for every atom and generation. The *trajectory depth* T sets the number of communication rounds; the *state width* H sets how many numerical values each atom carries.

The first step in an update was to calculate the message sent from a neighbouring atom j to atom i at generation t:

$$g_{ji} = \sigma\left(\frac{W_g e_{ji}}{\theta_b}\right), \qquad m_{ji}^{(t)} = g_{ji} \odot W_n h_j^{(t)} + W_e e_{ji},$$

The first expression calculates a *bond gate* g_{ji}, which controls how strongly neighbour j contributes to each of atom i's H channels. The vector e_{ji} describes the chemical bond connecting the two atoms. W_g forms learned weighted sums of those bond features, and the positive parameter θ_b, called the *bond temperature*, controls how sharply the gate responds. Dividing by a smaller θ_b moves gate values towards their extremes, so a contribution is more strongly passed or suppressed. A larger θ_b moves the values towards one half, making the gate less decisive. The logistic sigmoid $\sigma(z) = 1 / (1 + \exp(-z))$ converts every gate value to the interval between zero and one.

The second expression calculates the message $m_{ji}^{(t)}$. The vector $h_j^{(t)}$ is neighbour j's H-channel state at generation t. W_n, where n denotes *neighbour*, forms learned weighted sums of this state. The symbol $\odot$ means that corresponding elements are multiplied, so each bond-gate value scales the matching channel of the transformed neighbour state. W_e forms a further learned contribution directly from the bond features. In this context, a learned *linear map* such as W_g, W_n or W_e is a matrix of adjustable weights that forms new values by taking weighted sums of its inputs.

An atom may have more than one bonded neighbour. We therefore averaged its incoming messages and neighbouring states over its number of neighbours. This degree normalisation prevented atoms with more bonds from receiving larger totals merely because they had more neighbours:

$$a_i^{(t)} = \frac{1}{|N(i)|} \sum_{j \in N(i)} m_{ji}^{(t)}, \qquad \bar{h}_i^{(t)} = \frac{1}{|N(i)|} \sum_{j \in N(i)} h_j^{(t)}.$$

Here, $N(i)$ is the set of atoms directly bonded to atom i, and $|N(i)|$ is the number of atoms in that set. The vector $a_i^{(t)}$ is the mean incoming message. The state $h_i^{(t)}$ belongs to atom i itself, whereas $\bar{h}_i^{(t)}$ is the mean state of its directly bonded neighbours.

At every generation, the model combined the atom's present state, its mean incoming message, its fixed chemical features and the identity of the CYP isoform being predicted into a vector $r_i^{(t)}$:

$$r_i^{(t)} = \tanh\left(W_s h_i^{(t)} + a_i^{(t)} + W_x x_i + W_c c + b \right).$$

The resulting vector $r_i^{(t)} \in \mathbb{R}^H$ contains one value for each of the H channels, which the selected transition rule uses to calculate the next state. W_s forms learned weighted sums of the atom's present state $h_i^{(t)}$; W_x does the same for its fixed chemical features x_i; and W_c transforms the CYP context vector c. The mean incoming message $a_i^{(t)}$ adds information that has already been conditioned by the chemical bonds. The learned bias $b \in \mathbb{R}^H$ provides one adjustable offset for each channel. The hyperbolic tangent bounds each resulting value between −1 and 1.

Nine rule families use $r_i^{(t)}$ to calculate the next state. Conservative graph flux instead uses bonded-atom state differences directly, as specified below. The following equations define the rules in Figure 2A-E.

Gated residual. This rule was inspired by gates in recurrent neural networks [17]. For every atom and channel, it calculates how much of the current value to retain and how much of the calculated vector $r_i^{(t)}$ to accept (Figure 2A). The model recalculates this choice from the local molecular state at every generation:

$$\alpha_i^{(t)} = \sigma\left(W_\alpha [h_i^{(t)}, a_i^{(t)}, x_i, c] \right), \qquad h_i^{(t+1)} = (1 - s\alpha_i^{(t)}) \odot h_i^{(t)} + s\alpha_i^{(t)} \odot r_i^{(t)},$$

The acceptance gate $\alpha_i^{(t)} \in (0, 1)^H$ contains one value between zero and one for each channel. To calculate it, square brackets join, or concatenate, four vectors: the current atom state $h_i^{(t)}$, mean incoming message $a_i^{(t)}$, fixed chemical features x_i and CYP context c. W_α transforms this combined vector into the H gate values. The positive parameter s is the update scale. We capped $s\alpha_i^{(t)}$ at one in each channel; the 1 in the equation is therefore an H-component vector of ones. The symbol $\odot$ again denotes elementwise multiplication. The second expression then forms each new channel value as a weighted mixture of its previous value and the corresponding value in $r_i^{(t)}$.

Inertial reaction–diffusion. The numerical reaction and neighbour-exchange terms [4] combine with restoration and a velocity-like memory variable [20] (Figure 2B):

$$f_i^{(t)} = r_i^{(t)} + D \odot (\bar{h}_i^{(t)} - h_i^{(t)}) - R \odot h_i^{(t)},$$

$$v_i^{(t+1)} = \eta\gamma \odot v_i^{(t)} + \delta \odot f_i^{(t)}, \qquad h_i^{(t+1)} = \tanh\left(h_i^{(t)} + \delta \odot v_i^{(t+1)} \right).$$

The vector $f_i^{(t)} \in \mathbb{R}^H$ determines the change in atom i's velocity-like variable. It has three parts. The vector $r_i^{(t)}$ supplies the learned contribution from the atom's local inputs. The difference $\bar{h}_i^{(t)} - h_i^{(t)}$ compares the mean neighbour state with the atom's own state and therefore provides the numerical difference used for diffusion across bonds. The non-negative H-channel vectors D and R set the strengths of diffusion and restoration, respectively; the restoring term draws the state back towards zero.

The auxiliary velocity $v_i^{(t)} \in \mathbb{R}^H$ carries change between generations and was set to zero at the start of each trajectory. The H-channel vector γ controls how strongly the previous velocity is damped, while δ gives the step size for each channel. The scalar η sets the overall strength of inertia. Sigmoid or softplus transformations kept these quantities within stable ranges; the softplus transformation was used where a parameter had to remain positive. Finally, tanh bounded the updated state.

FitzHugh–Nagumo. Each atom's H channels divide into excitation u and recovery v [21, 22], updated with local inputs and neighbour exchange (Figure 2C):

$$\Delta u_i^{(t)} = u_i^{(t)} - \frac{(u_i^{(t)})^3}{3} - v_i^{(t)} + \kappa_s \tanh(W_u r_i^{(t)}) + D_u(\bar{u}_i^{(t)} - u_i^{(t)}),$$

$$\Delta v_i^{(t)} = \epsilon\left[u_i^{(t)} + q - v_i^{(t)} + 0.1\tanh(W_v r_i^{(t)}) \right] + D_v(\bar{v}_i^{(t)} - v_i^{(t)}),$$

We then updated the two channel groups together:

$$h_i^{(t+1)} = \tanh\Big([u_i^{(t)} + s\Delta u_i^{(t)},\ v_i^{(t)} + s\Delta v_i^{(t)}] \Big).$$

The vectors $u_i^{(t)}$ and $v_i^{(t)}$ each contain $H / 2$ values and together form the complete state $h_i^{(t)}$. The neighbour means $\bar{u}_i^{(t)}$ and $\bar{v}_i^{(t)}$ average the corresponding half-states over the atoms in $N(i)$. The quantities $\Delta u_i^{(t)}$ and $\Delta v_i^{(t)}$ are the proposed changes to excitation and recovery. W_u and W_v transform $r_i^{(t)}$ from H values into separate inputs of $H / 2$ values for the excitation and recovery channels.

The scalar κ_s controls the strength of chemical stimulation. The non-negative parameters D_u and D_v control graph diffusion for excitation and recovery. The positive parameter ε sets the recovery timescale, q shifts the recovery response, and 0.1 is the fixed coefficient applied to $\tanh(W_v r_i^{(t)})$ in the recovery calculation. The positive parameter s is the numerical integration step that advances both halves. Square brackets concatenate the updated excitation and recovery vectors to reconstruct the H-channel state, and tanh bounds its values.

Kuramoto–Sakaguchi. The phase formulation [23, 24] converts each channel using $\phi_i^{(t)} = \pi h_i^{(t)}$ and combines local phase-advance rates with neighbour coupling (Figure 2D):

$$q_i^{(t)} = \frac{1}{|N(i)|} \sum_{j \in N(i)} g_{ji} \odot \sin\Big(\phi_j^{(t)} - \phi_i^{(t)} - \psi \Big),$$

$$\phi_i^{(t+1)} = \phi_i^{(t)} + s\Big[\omega_i^{(t)} + K q_i^{(t)}\Big], \qquad \omega_i^{(t)} = A\tanh(W_\omega r_i^{(t)}).$$

The vector $\phi_i^{(t)} \in [-\pi, \pi)^H$ contains the H phases of atom i. For each bonded neighbour j, the sine term measures the phase difference between the two atoms after allowing for the preferred offset ψ, called the *phase lag*. The sine is calculated separately for each channel. The bond gate $g_{ji} \in (0, 1)^H$ controls how strongly that bond contributes in each channel, and $\odot$ applies this control element by element. Averaging over the neighbour set $N(i)$ produces the mean coupling signal $q_i^{(t)} \in \mathbb{R}^H$.

The parameter K sets the overall coupling strength and s is the integration step. The vector $\omega_i^{(t)} \in \mathbb{R}^H$ is the atom's chemically conditioned *natural frequency*: the rate at which its phases would tend to advance before neighbour coupling is added. The learned weights W_ω transform $r_i^{(t)}$ to calculate these rates, and A sets their scale. After each update, we wrapped every phase modulo 2π into the interval [−π,π), because angles differing by a complete turn are equivalent. Dividing by π converted the phase back to $h_i^{(t+1)} \in [-1, 1]$. A hyperparameter search conducted only on the training data selected the phase lag, coupling strength and frequency scale.

Delayed memory. Following the delayed-feedback construction [25], the rule combines current local inputs with the state retained d generations earlier (Figure 2E):

$$\tilde{r}_i^{(t)} = \tanh\Big(W_d [r_i^{(t)}, h_i^{(t-d)}] \Big),$$

$$h_i^{(t+1)} = \tanh\Big((1 - \zeta s) h_i^{(t)} + s\Big[(1-\mu) r_i^{(t)} + \mu \tilde{r}_i^{(t)} + \kappa_d (h_i^{(t-d)} - h_i^{(t)})\Big] \Big).$$

The vector $h_i^{(t-d)} \in \mathbb{R}^H$ is atom i's state d generations earlier. When fewer than d generations were available, the rule used the earliest retained state. To calculate $\tilde{r}_i^{(t)} \in \mathbb{R}^H$, square brackets join the current vector $r_i^{(t)}$ and the earlier state $h_i^{(t-d)}$; the learned weights W_d then transform this combined vector into H values. The hyperbolic tangent bounds both $\tilde{r}_i^{(t)}$ and the final updated state.

The mixing coefficient $\mu \in [0, 1]$ controls the balance between $r_i^{(t)}$ and $\tilde{r}_i^{(t)}$, which also includes information from the earlier state. The 1 in $(1 - \mu)$ is the scalar multiplicative identity. The parameter κ_d controls feedback from the difference $h_i^{(t-d)} - h_i^{(t)}$, which measures how far the earlier state is from the present state. The parameter ζ damps the

present state, and the positive parameter s sets the update scale. The rule-specific search selected d and these coefficients from their predefined search ranges.

Gray–Scott implementation. The rule splits the H channels into equal groups u and v and rescales their values from [−1,1] to [0,1]. The product uv^2 decreases u and increases v. Separate neighbour-difference terms exchange each quantity across bonds. A replenishment term increases u towards one, while a removal term reduces v. Learned transformations of $r_i^{(t)}$ contribute to both groups with coefficient 0.02. After each update, the rule clips both groups to [0,1] and converts them back to [−1,1].

Coupled-map implementation. The rule rescales each state value to $q = (h + 1) / 2$ and applies the logistic function $aq(1 - q)$, where a is a selected parameter between 2.5 and 4.0. It combines this result with a sigmoid transformation of the learned local-input vector $r_i^{(t)}$. It separately applies the same logistic function to the mean neighbour value, then mixes the two results using a coupling weight capped at 0.95. The rule clips the result to [0,1] and converts it back to [−1,1]. Applying the function to the neighbour mean is the specific averaging order used in this implementation.

Activator–inhibitor implementation. The rule divides the H channels equally into u and v. Its change in u contains $u - u^3 / 3 - v$, a learned transformation of $r_i^{(t)}$ and a neighbour-difference term. Its change in v combines a scaled response to u, an adjustable offset, a term that reduces v, neighbour exchange and a second learned transformation of $r_i^{(t)}$. Both groups advance using the selected step size and are bounded by the hyperbolic tangent. The FitzHugh–Nagumo version above uses a related structure with its own parameter ranges and fixed relationships between coefficients.

Damped-symplectic implementation. The rule divides the H channels equally into position-like values q and momentum-like values p. A learned transformation of $r_i^{(t)}$, the difference between the neighbours' mean q and the atom's own q, and a restoring term towards zero determine the momentum update. The rule multiplies the previous p by a damping factor between 0.8 and 0.999 before adding the scaled change. It then advances q using this new momentum. Hyperbolic tangent bounds both stored channel groups. The name describes the momentum-first order of calculation; the implemented damping and bounding operations modify the corresponding ideal mechanical scheme.

Conservative graph-flux implementation. For each directed bond from j to i, the rule applies a learned linear transformation without a bias to $h_j - h_i$, takes its hyperbolic tangent and multiplies by the bond gate. Reversing the bond changes the sign of this transfer because the two directions share bond descriptors and the transformation is odd. Summing incoming transfers gives each atom's update. Every atom uses the same divisor, the maximum number of bonded neighbours in the molecular graph, with a minimum divisor of one. The rule then adds the scaled transfers directly to the current state. Equal and opposite transfers preserve the sum across atoms in each channel, up to numerical rounding; this rule applies no final hyperbolic tangent to the updated state.

Pooling converts trajectories of differing atom counts and generation lengths into the fixed-length fingerprints introduced in Section 2.3. The following summaries define those fingerprints.

The basic fingerprint contained five types of summary. The final atom-state mean described the average channel values at the last generation, while the final atom-state variance described how much atoms differed from one another at that point. The time-averaged atom state summarised channel values across the trajectory. The temporal variance of the molecular mean state measured how much the molecule-wide average changed over generations. The mean state-change energy summarised the magnitude of changes between successive states. A multiscale version also retained the molecular mean states at 12.5%, 25%, 50%, 75% and 100% of the trajectory, allowing the prediction step to distinguish early, intermediate and final behaviour.

The model predicted four different CYP isoforms. The one-hot CYP context vector therefore selected the copy of the fingerprint belonging to the target isoform: the relevant block was active and the other three blocks were zero. This produced CYP-specific readout features and allowed ridge regression to associate the same types of trajectory measurement with different isoforms.

The genetic search also tested optional summaries of trajectory change. The *dynamic observables* option added the difference between the initial and final molecular mean states and a normalised correlation between the molecular mean states at successive generations. The correlation measured how strongly one generation resembled the next. *Summaries of squared changes between sampled molecular mean states* added the squared changes between the molecular mean states at consecutive sampling points listed above. The search also tested correlations separated by several generation lags, smooth approximations to the largest and smallest molecular mean states, and separate summaries of increases and decreases in state values. Together, these options allowed the readout to use persistence, variation, direction and scale of change rather than relying only on the final state.

Appendix B. Evolutionary-Strategy Configuration

The evolutionary-strategy experiment changed the procedure used to optimise numerical weights while holding the CYP3A4 Graph-CA design fixed. This allowed a direct comparison with backpropagation. Table B1 lists the fixed model settings that were removed from the main-text summary.

Fixed setting	Value and meaning
Transition rule and prediction target	A damped-symplectic Graph-CA predicted CYP3A4 pIC50. Section 2.3 describes this transition rule.
State width and trajectory depth	Each atom carried 16 numerical channels, and the transition rule was repeated for 32 generations.
Atom features	The baseline atom descriptors were supplemented by the Periodic and Electronic feature groups in Table 2.
Bond messages	Information passed between bonded atoms was conditioned on the recorded bond descriptors, including whether a bond was single, double, triple or aromatic.
Trajectory pooling	Multiscale pooling summarised the trajectory at several generations, allowing the fingerprint to retain early, intermediate and final behaviour as described in Appendix A.
Update scale	A value of 0.25 controlled the magnitude of the change applied to the atomic states at each generation.
Support fraction	A value of 0.6 assigned 60% of the eligible molecules in each training batch to the support subset used to fit the ridge readout. The remaining molecules formed the query subset.
Ridge penalty	A value of 0.1 discouraged excessively large fingerprint coefficients in the ridge-regression readout.

Table B1. Fixed Graph-CA settings used throughout the evolutionary-strategy experiment.

An evolutionary configuration was one combination of the settings in Table B2. We evaluated 32 such combinations. For each evolutionary update, random changes were generated in opposite pairs around the current weights, as described in Section 2.4.3.

Evolutionary setting	Meaning	Values tested
Population size	The total number of candidate weight sets evaluated during one update, including both members of every opposite pair.	32, 64, 96 or 128 candidates.
Perturbation standard deviation	The typical size of the random changes applied to the current weights. The changes were drawn from a Gaussian, or bell-shaped, distribution.	0.005, 0.01, 0.02, 0.04 or 0.08.
Evolutionary learning rate	The scale of the adjustment made to the current weights after the candidate pairs had been compared.	0.0005, 0.001, 0.003, 0.006 or 0.01.
Training-batch size	The number of molecules selected for one evolutionary update.	256, 512, 1,024 or 1,600 molecules.
Batch composition	Random composition selected molecules without using their measured activity to arrange the support and query subsets. Activity-stratified composition distributed query molecules across the observed pIC50 range, with the remaining molecules assigned to the support subset.	Random or activity-stratified.
Stopping policy	The rule specifying the maximum number of training epochs and how long training continued without sufficient improvement in validation MA-ST-RAE.	The four combinations of maximum training length and stopping thresholds defined in Table B3.

Table B2. Evolutionary-strategy settings and tested values.

Each epoch comprised one evolutionary update. Training stopped at the maximum number of epochs or when validation MA-ST-RAE failed to improve by at least the minimum improvement for the stated number of consecutive epochs, called the patience period.

Stopping policy	Maximum epochs	Patience period	Minimum improvement in MA-ST-RAE
Up to 40 updates	40	8	0.003
Up to 60 updates	60	14	0.002
Up to 80 updates	80	20	0.002
Up to 100 updates	100	25	0.001

Table B3. Stopping policies tested during evolutionary-strategy optimisation.

Appendix C. Hybrid Genetic Graph-CA Structure Search

The hybrid genetic search changed discrete choices in the CYP3A4 Graph-CA design while backpropagation trained the numerical weights of every candidate. Each candidate was defined by a complete list of model-design choices. The numerical optimiser settings associated with each transition-rule family remained fixed, allowing the genetic algorithm to compare model structures without simultaneously searching a second set of continuous training parameters.

Design choice	Meaning	Options tested
Transition-rule family	The mathematical rule used to update every atomic cell.	Damped symplectic, FitzHugh-Nagumo or delayed memory.
Recurrent generations	The number of times the transition rule was applied.	16, 32, 64 or 128.
State width	The number of numerical channels carried by each atom.	8, 16 or 24.
Atom-feature profile	The baseline descriptors supplemented by selected feature groups from Table 2.	Periodic and Electronic; Electronic and Local environment; Periodic and Valence; or all five additional groups.
Trajectory pooling	The method used to convert the variable-length trajectory into a fixed-length fingerprint. Basic pooling used the five core summaries in Appendix A; multiscale pooling also retained five intermediate states; temporal-attention pooling learned how strongly to weight those five states.	Basic, multiscale or temporal attention.
Degree normalisation	The strength with which the number of bonded neighbours reduced the combined incoming message.	Exponent 0.5 or 1.0.
Chemical-feature gating	Whether a learned gate adjusted the contribution of each chemical descriptor.	Disabled or enabled.
Initial molecular mean and variance	Whether the initial molecular mean and variance were retained in the final fingerprint.	Disabled or enabled.
Dynamic observables	Whether the fingerprint included change from the initial to final molecular mean and correlation between successive generations.	Disabled or enabled.
Squared changes between sampled molecular mean states	Whether the fingerprint included squared changes between the five multiscale sampling points.	Disabled or enabled.
Channel-specific timescales	Whether each channel learned its own rate of movement towards the proposed next state.	Disabled or enabled.
State correlations across generation intervals	Whether the fingerprint included correlations between states separated by several generation intervals.	Disabled or enabled.
Temporal extrema	Whether the fingerprint included smooth estimates of the largest and smallest molecular mean states reached during the trajectory.	Disabled or enabled.
Separate summaries of state increases and decreases	Whether increases and decreases in state values were summarised separately.	Disabled or enabled.

Table C1. Model-design choices varied by the hybrid genetic search.

The search used four genetic generations containing 24 unique candidate structures each. Tournament selection chose a parent by comparing three candidates. Uniform crossover constructed a child by choosing each design setting from either parent, and mutation independently changed each setting with probability 0.25.

Every candidate was trained by backpropagation on two scaffold-separated folds for at most 35 epochs. The 96 candidates therefore produced 192 screening runs. The four strongest candidates and the fixed damped-symplectic reference then underwent confirmation using five scaffold folds, two seeds and at most 80 epochs, producing another 50 runs and 242 runs in total. MA-ST-RAE on the internal validation folds guided selection of candidate model configurations, with lower prediction error favouring selection. RMSE provided a complementary measure of prediction error. The sealed holdout and blind labels were excluded from screening, selection and confirmation.

Component	Selected setting
Transition rule	FitzHugh-Nagumo.
Recurrent generations	128.
State width	24 channels.
Atom-feature profile	Comprehensive, comprising the baseline descriptors and all five additional feature groups in Table 2.
Trajectory pooling	Multiscale.
Degree normalisation	Exponent 1.0.
Dynamic observables	Enabled.
Squared changes between sampled molecular mean states	Enabled.
Chemical-feature gating	Disabled.
Initial molecular mean and variance	Disabled.
Channel-specific timescales	Disabled.
State correlations across generation intervals	Disabled.
Temporal extrema	Disabled.
Separate summaries of state increases and decreases	Disabled.

Table C2. Configuration selected by the hybrid genetic search.

Appendix D. Genetic Programming of the Transition Rule

The genetic-programming experiment retained the hybrid search's selected CYP3A4 configuration and changed only the mathematical transition rule. Each candidate rule was represented as an expression tree. Mathematical operations formed the branches, while the available input vectors formed the terminal leaves. A candidate rule remained fixed throughout its training and validation trajectories, although backpropagation continued to train its numerical weights.

Candidate input	Meaning
Local input, $r_i^{(t)}$	The vector calculated from the atom's current state, bonded-neighbour information, chemical descriptors and CYP context.
Current state, $h_i^{(t)}$	The atom's H-channel state at the current generation.
Neighbour difference, $\bar{h}_i^{(t)} - h_i^{(t)}$	The mean state of the atom's bonded neighbours minus the atom's own state.
Delayed-state difference, $h_i^{(t-d)} - h_i^{(t)}$	A retained earlier state of the atom minus its current state.

Table D1. Input vectors available to each candidate transition rule.

Operation group	Permitted operations
Two-input operations	Addition, subtraction, multiplication and the mean of two inputs.
One-input operations	Negation, hyperbolic tangent, sine and multiplication by a fixed scale.
Fixed scales	−1.0, −0.5, 0.25, 0.5 or 1.0.

Table D2. Mathematical operations available to genetic programming.

Expression trees were limited to a maximum depth of four. The search evaluated four successive populations, each containing 24 distinct candidate cellular automaton transition rules. Tournament selection compared groups of three candidates when choosing parents. Subtree crossover exchanged a complete part of one parent expression with a part of another. With probability 0.35, mutation replaced a randomly selected part of the resulting expression with a newly generated subtree. A penalty of 0.0005 per expression-tree node discouraged additional complexity unless it improved predictive performance.

The model using each candidate cellular automaton transition rule was trained using two scaffold folds and a maximum of 35 epochs. Five finalists then underwent five-fold confirmation with two independent seeds and a maximum of 80 epochs. This produced 242 training runs. Backpropagation trained the weights used for bond messages, local-input construction and numerical transformations through the ridge fit in Section 2.4.2. MA-ST-RAE on the internal scaffold-separated validation folds measured the prediction error of each candidate cellular automaton transition rule. This error and the expression-size penalty described above guided selection, favouring accurate rules with fewer mathematical operations. RMSE provided a complementary measure of prediction error. The sealed holdout and blind labels were excluded throughout the search and confirmation stages.

The selected cellular automaton transition rule averaged the local-input vector and bonded-neighbour difference. Its expression tree contained three nodes: the two inputs and the averaging operation joining them. The numerical output of this expression is given in Section 2.4.5.

Appendix E. A Visual Gallery of Molecular Space-Time

This gallery illustrates the aesthetic beauty and visual variety of molecular space-time: numerical information passes between bonded atomic cells and forms a trajectory as the cellular automaton repeatedly updates their states. The ten pages each present one transition-rule family, providing a visual companion to the quantitative analyses in the main text. We selected three archived molecule-and-CYP-context trajectories for each family, giving 30 paired examples across the gallery.

Each row presents the same recorded trajectory in two ways. The left panel places atoms in their stored graph order along the horizontal axis and successive generations down the vertical axis. Colour represents the first numerical state channel of each atom. Atom order serves as a display arrangement; the chemical bonds define which atoms exchange information. The right panel places those same colours on atom traces derived from a three-dimensional molecular conformer and rotates the conformer as successive generations descend. This rotation provides different views of the structure. The conformer and rotation are visualisation choices, while the colours come from the recorded cellular automaton states.

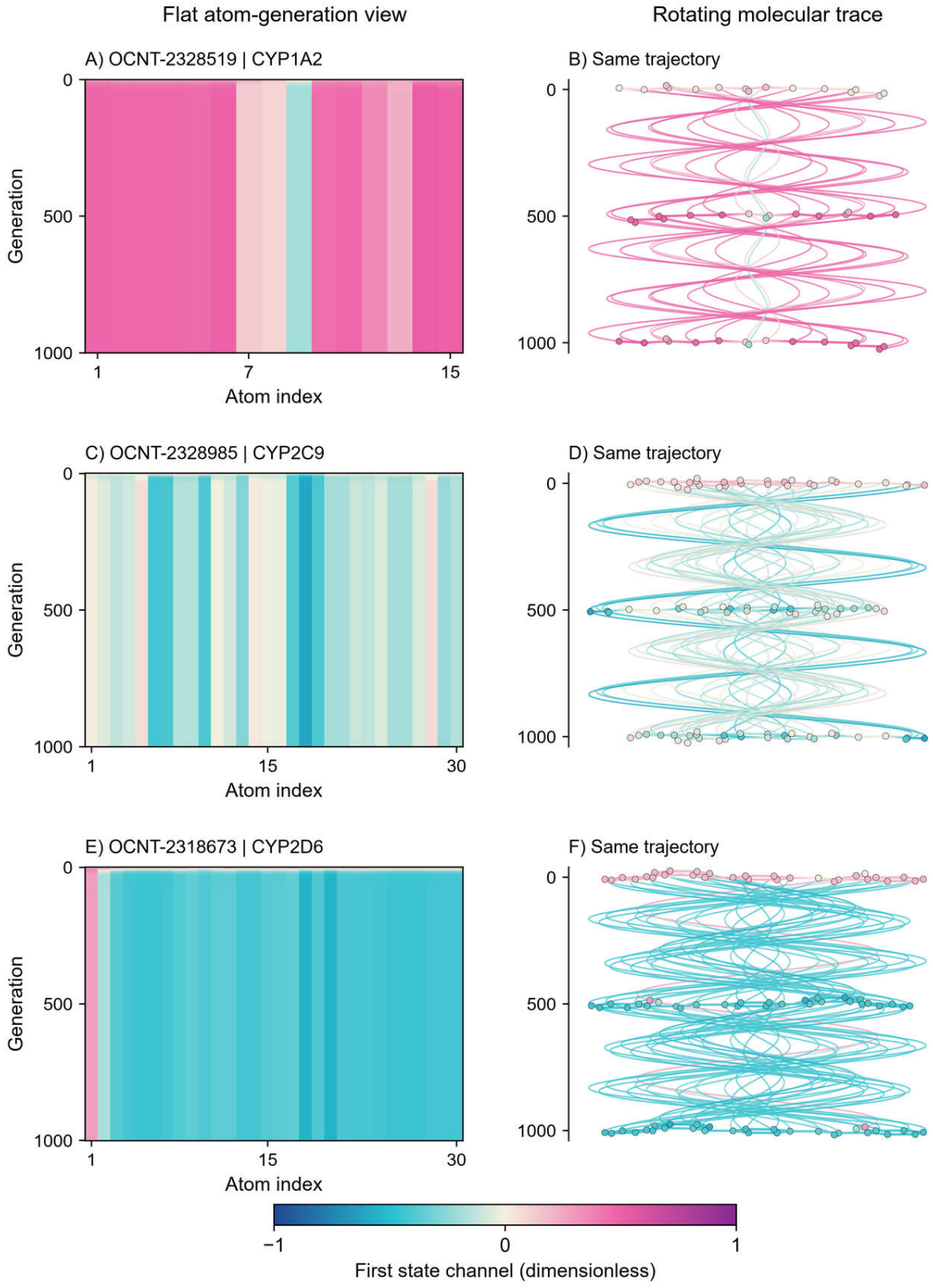
The coloured curves on the right connect successive displayed positions of each atom. Chemical bonds appear at the beginning, midpoint and end of the displayed trajectory, with colours interpolated between their endpoint atoms. Small vertical offsets retain the molecular shape around each generation's position. Both panels in a pair use the same state values and colour scale. Kuramoto-Sakaguchi colours represent the first wrapped phase divided by π on a cyclic scale; the other pages show the first state channel on a linear, dimensionless scale. The same scale applies to all three examples on a page, and the conservative graph flux page uses a wider range to contain its recorded values.

Training provenance. Every example in this gallery comes from a fixed-rule Graph-CA trained by backpropagation, labelled BP-Graph-CA on each page. The rule family was specified before training, and the learned numerical weights remained fixed during the recorded trajectory. Genetic-algorithm structure search, genetic programming and evolutionary-strategy weight training are separate experiments described in the main text; this gallery contains trajectories from the fixed-rule backpropagation archives.

Reading the gallery. Figures E1-E10 cover gated residual, inertial reaction-diffusion, FitzHugh-Nagumo, Kuramoto-Sakaguchi, delayed memory, Gray-Scott, coupled map, activator-inhibitor, damped symplectic and conservative graph flux, respectively. Panels A/B, C/D and E/F pair three examples on each page; molecule identifiers and CYP contexts appear above the left panels. The first nine pages show generations 0-1000, while E10 shows the available recorded interval, generations 0-256. Examples may share a molecule under different CYP contexts. Independently trained models and different molecules contribute to the visual differences, so the gallery supports aesthetic exploration and qualitative inspection; quantitative claims about chaos or attractor geometry rely on the analyses in the main text.

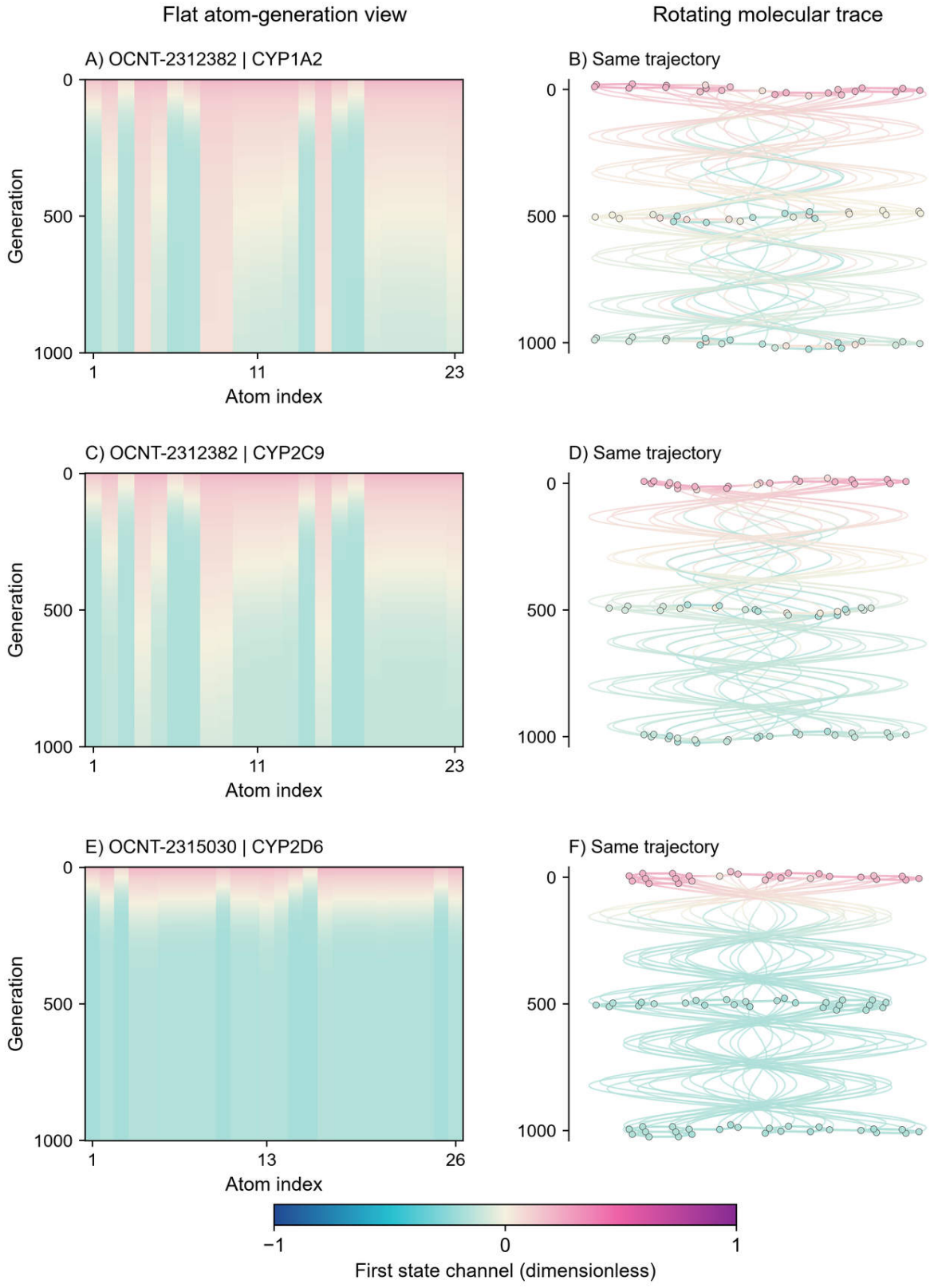
E1. Gated residual

Training: BP-Graph-CA (fixed rule; backpropagation)



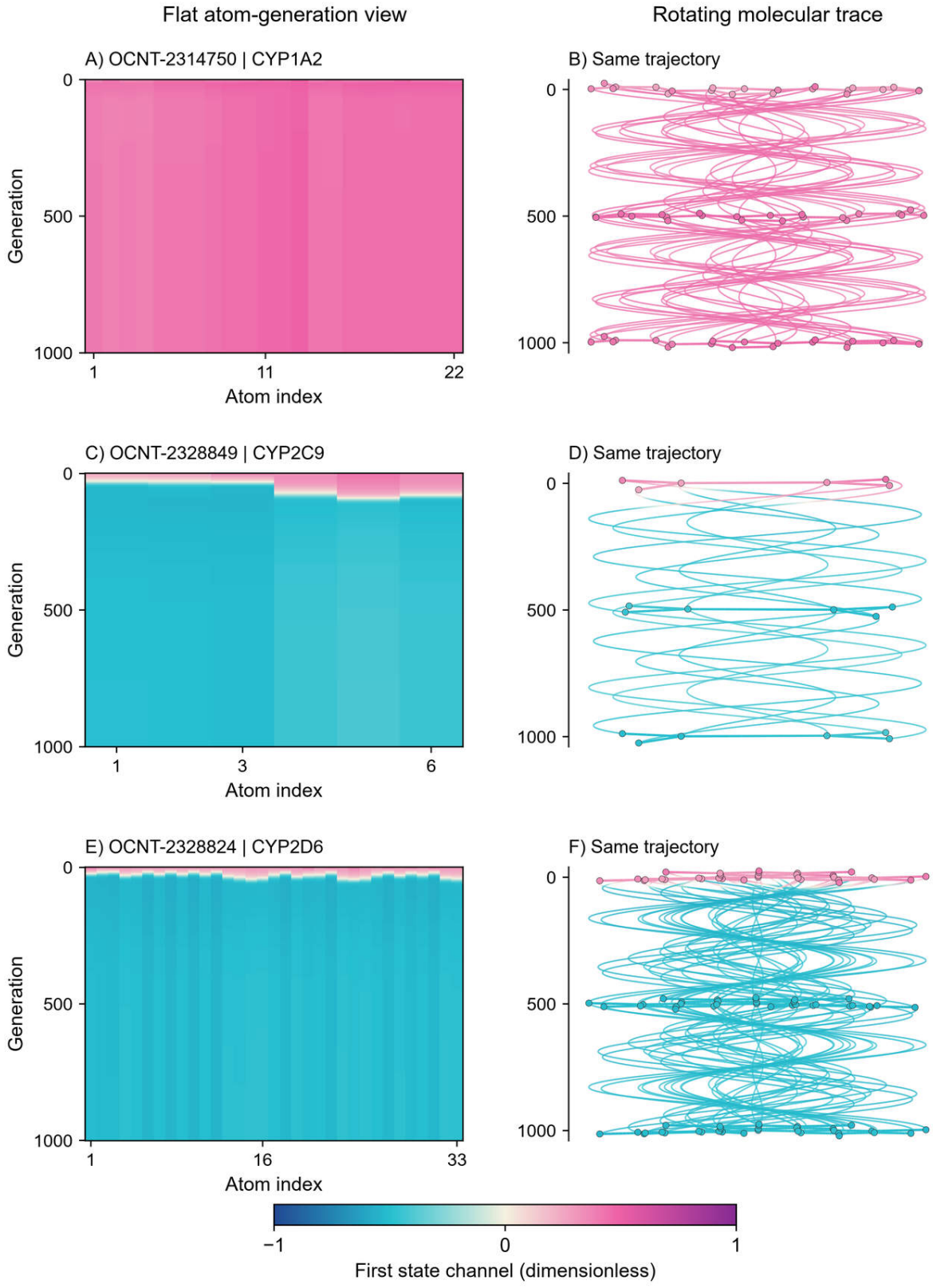
E2. Inertial reaction-diffusion

Training: BP-Graph-CA (fixed rule; backpropagation)



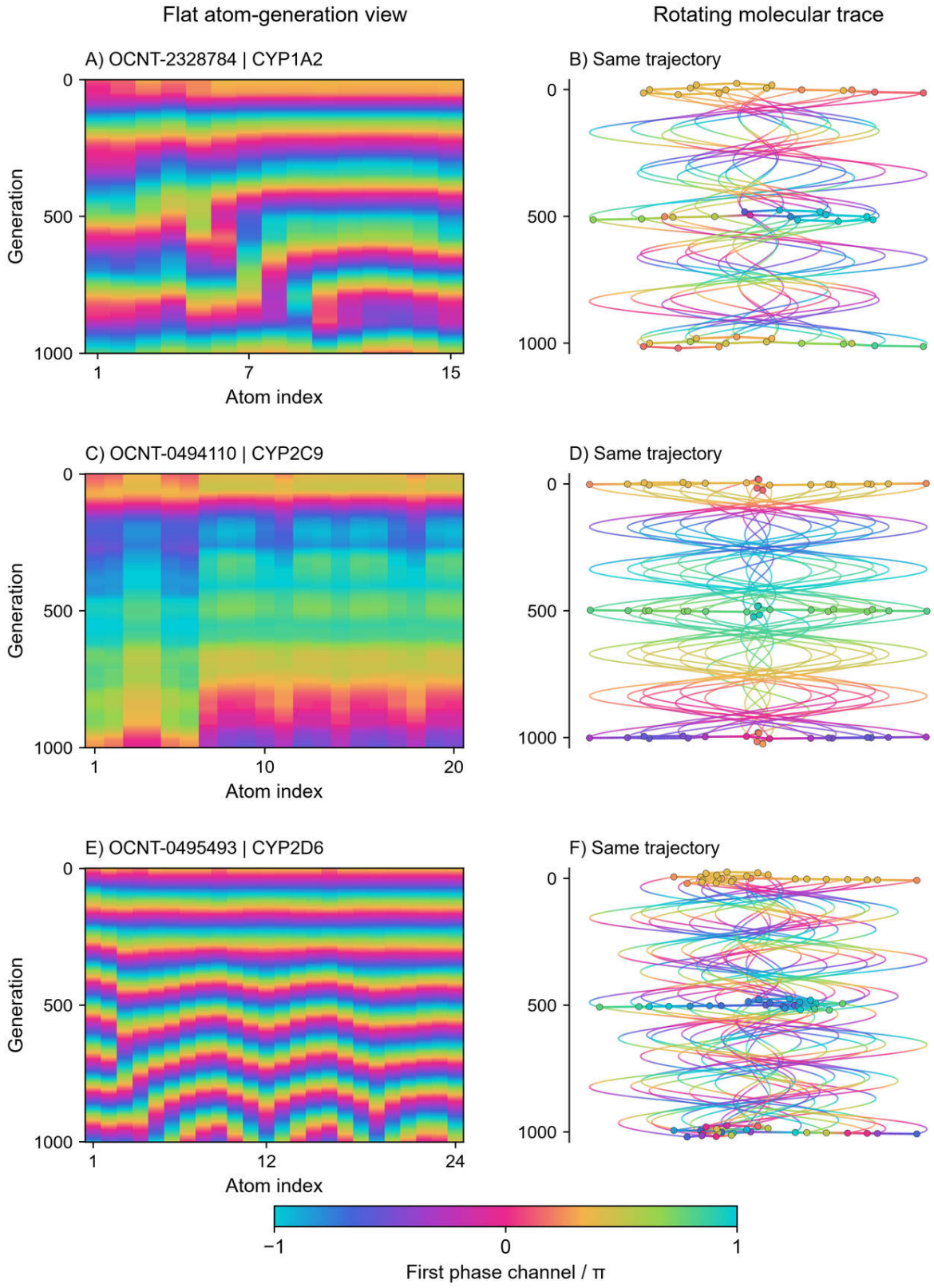
E3. FitzHugh-Nagumo

Training: BP-Graph-CA (fixed rule; backpropagation)



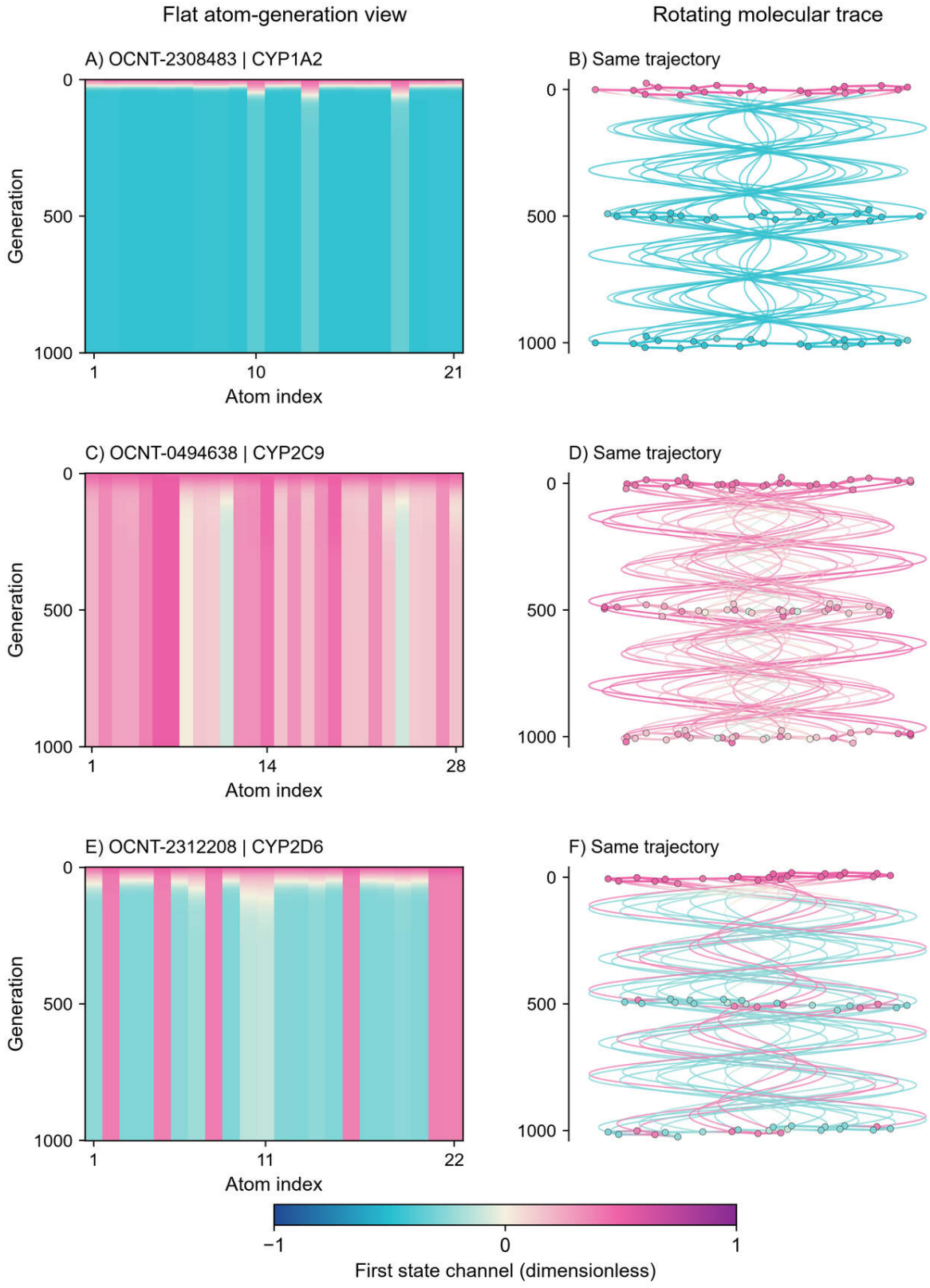
E4. Kuramoto-Sakaguchi

Training: BP-Graph-CA (fixed rule; backpropagation)



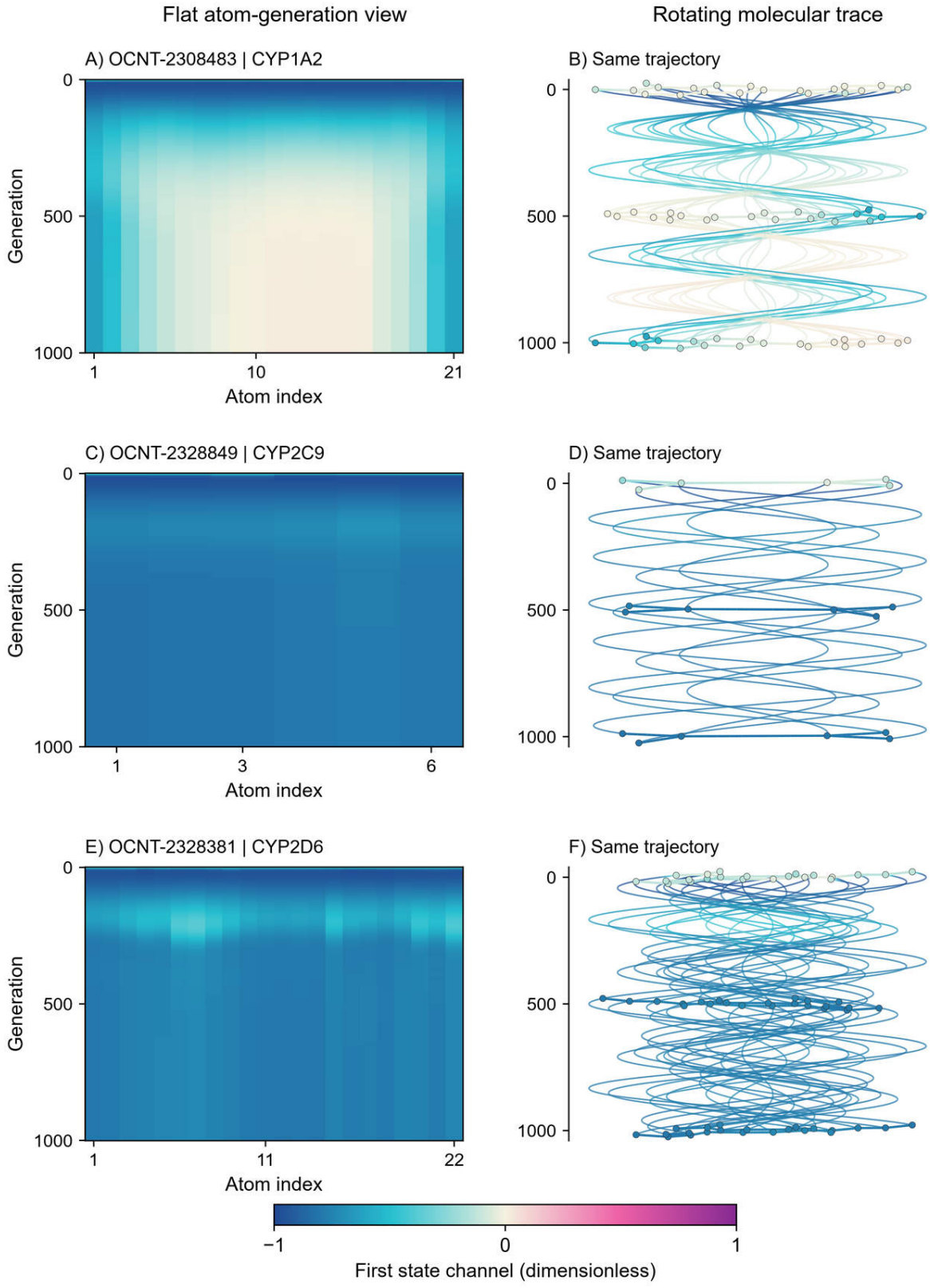
E5. Delayed memory

Training: BP-Graph-CA (fixed rule; backpropagation)



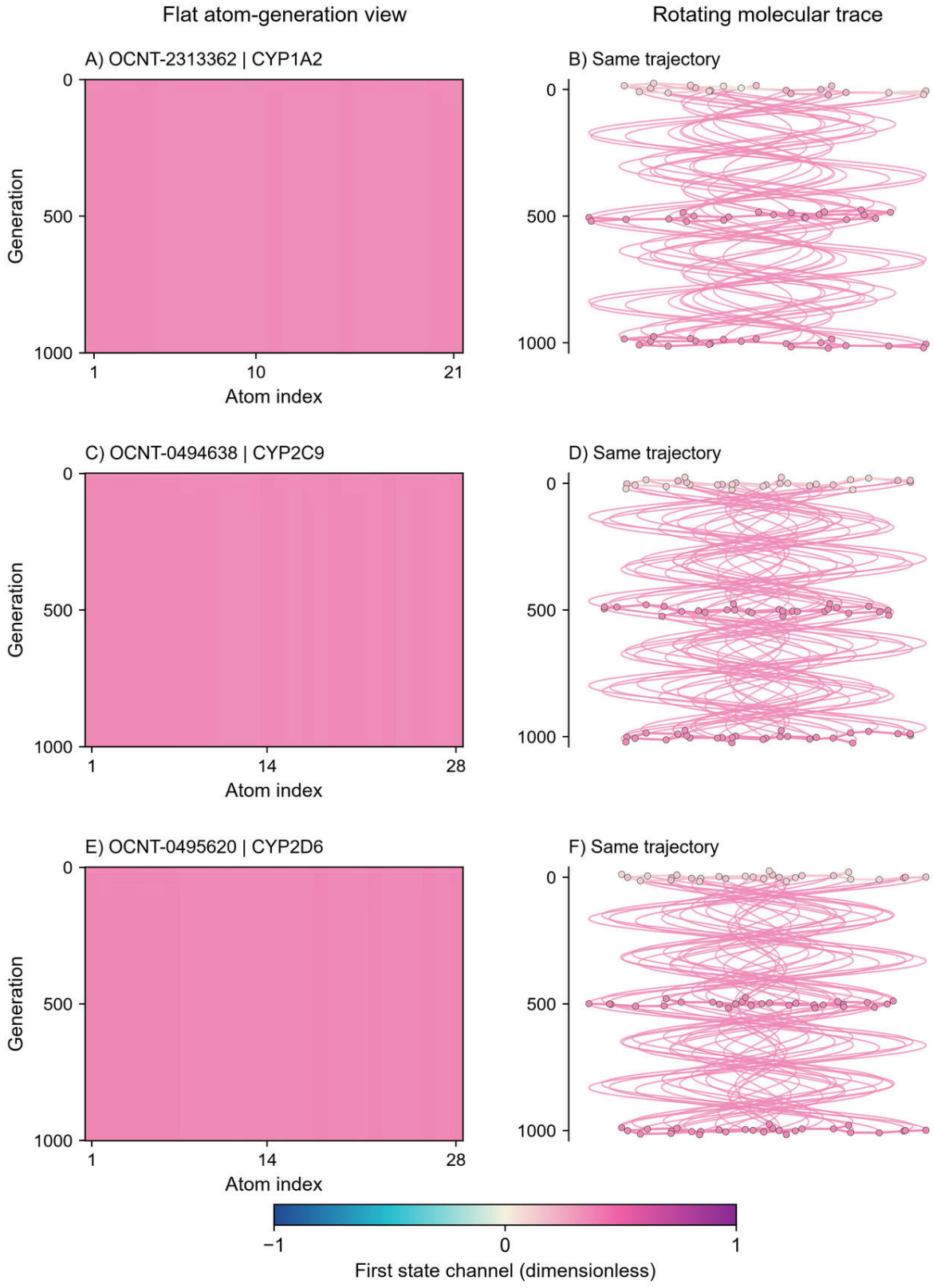
E6. Gray-Scott

Training: BP-Graph-CA (fixed rule; backpropagation)



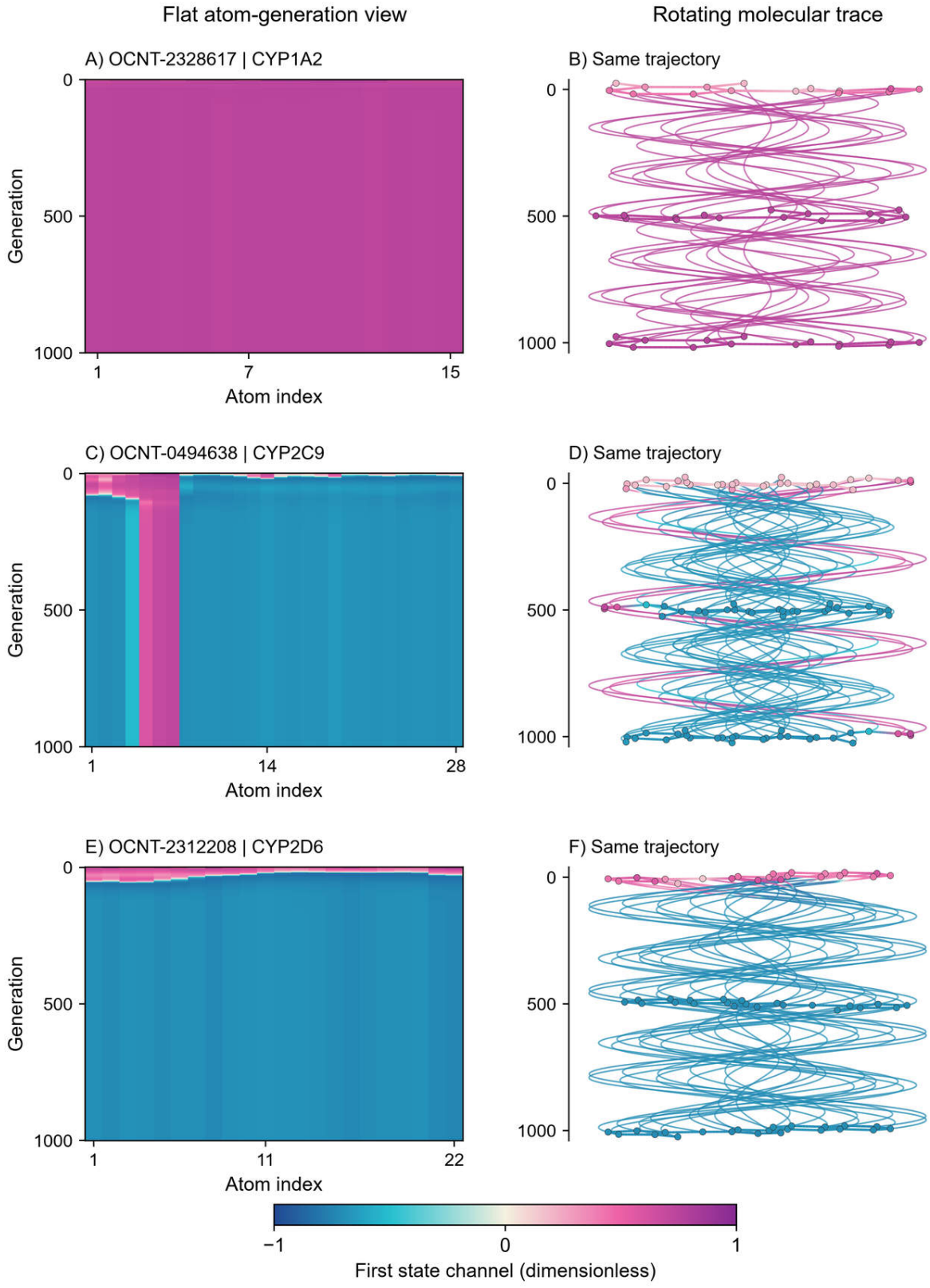
E7. Coupled map

Training: BP-Graph-CA (fixed rule; backpropagation)



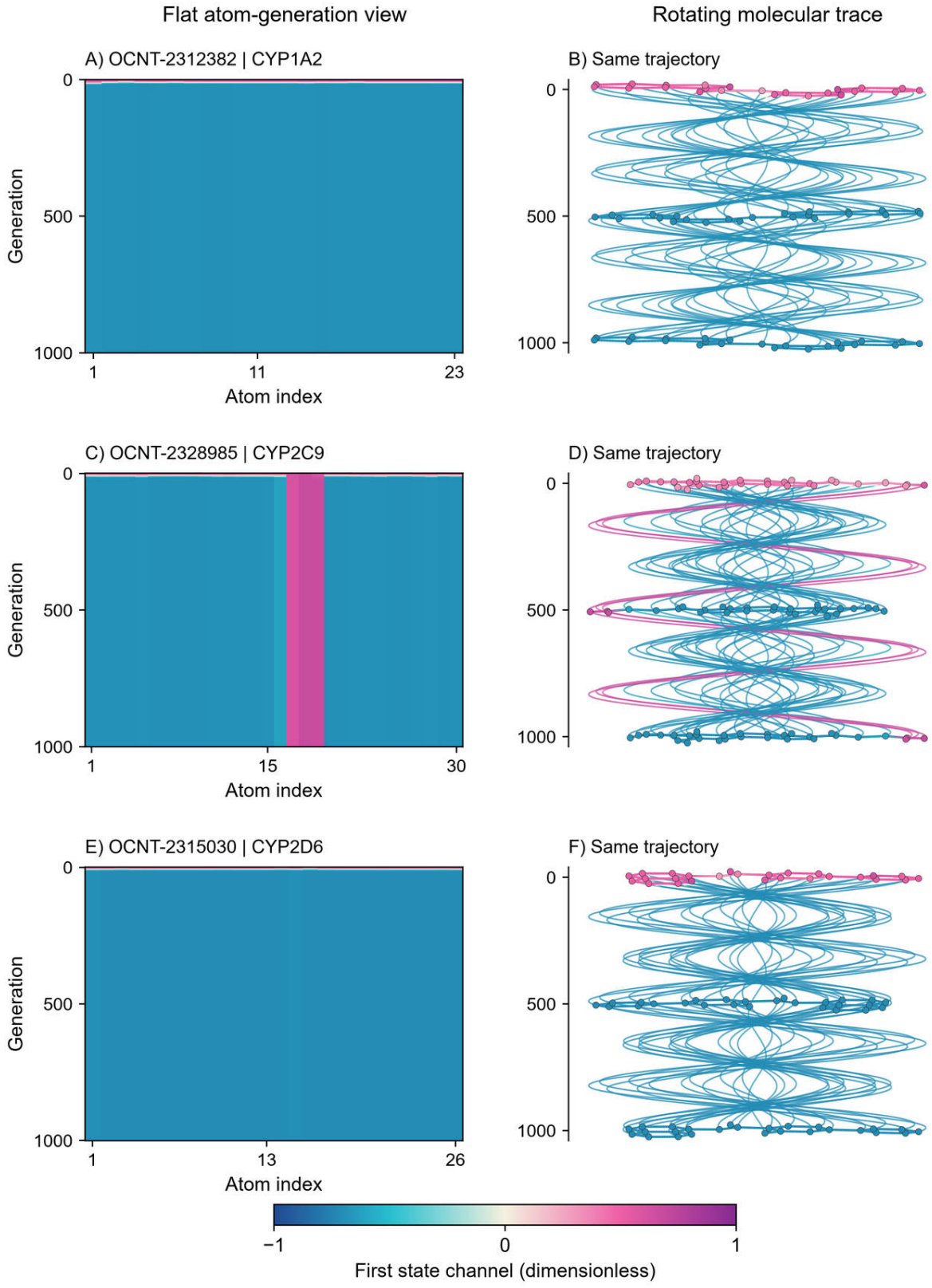
E8. Activator-inhibitor

Training: BP-Graph-CA (fixed rule; backpropagation)



E9. Damped symplectic

Training: BP-Graph-CA (fixed rule; backpropagation)



E10. Conservative graph flux

Training: BP-Graph-CA (fixed rule; backpropagation)

