Supplementary Information

Exact ensemble bifurcation analysis reveals hidden bistability and makes signalling dynamics predictable from kinetic parameters

Guturu L Harika[1] and K. Sriram[2,*]

[1]Department of Computational Biology, IIIT-Delhi, Okhla Phase III, New Delhi 110020, India
[2]Department of Computational Biology, Center for Computational Biology, and Centre for Artificial Intelligence, IIIT-Delhi, Okhla Phase III, New Delhi 110020, India
[*]Corresponding author: sriramk@iiitd.ac.in

This Supplementary Information accompanies the main text. It reports, for the ten-class ensemble of 40,000 parameter sets (four sampling iterations of 10,000 sets each, exact labels from the inverse method), every analysis that is summarised but not shown in full in the main text, together with the comparison with single-start continuation on which main-text Fig 5 is based. Figure and table numbers without the prefix "Supplementary" refer to the main text. All numbers are taken from the output files of the codes listed in Supplementary Table 17 (Supplementary Note 16).

Contents

Supplementary Note 1. Structural analysis of the reaction network **4**

Supplementary Note 2. Exact labelling of the four sampling iterations **6**

Supplementary Note 3. Validation of the exact labels against MATCONT **7**

Supplementary Note 4. Normal-form coefficients in each sampling iteration **10**

Supplementary Note 5. Comparison with single-start continuation **11**

Supplementary Note 6. The biphasic criterion **13**

Supplementary Note 7. The sampling design and its imprint on the features **13**

Supplementary Note 8. Machine-learning data and splits **14**

Supplementary Note 9. Feature engineering and feature selection **14**

Supplementary Note 10. Model comparison, ROC curves and confusion matrices **20**

Supplementary Note 11. Convergence of the classifiers **24**

Supplementary Note 12. Embedding audit **26**

Supplementary Note 13. Decisive two-class tests **27**

Supplementary Note 14. Misclassified parameter sets **29**

Supplementary Note 15. Computational cost **29**

Supplementary Note 16. Supplementary files and code **30**

Supplementary Note 17. Per-class feature contributions (TreeSHAP) **33**

Supplementary Note 18. Transitions between classes **35**

Supplementary Note 19. Detached branches beyond the labelling window **40**

Supplementary Note 20. Figures and tables referred to from the main text **43**

Supplementary Note 21. Experimentally accessible quantities that set the class **50**

Supplementary Figures

1 MATCONT with and without the inverse method (ten parameter sets)
2 Exact labels against MATCONT seeded on every branch (4,000 parameter sets)
3 Singularities found by continuation agree with the exact computation
4 Limits of single-start continuation
5 The biphasic criterion is insensitive to the prominence threshold
6 JSD and broken-stick selection for all 45 class pairs (pooled)
7 JSD and broken-stick selection within each sampling iteration
8 Summary of the divergence-based selection
9 Two-parameter views of two class pairs
10 Pooled feature importance of all models
11 Confusion matrices within each sampling iteration
12 Pooled confusion matrices of all models
13 Per-class convergence of the pooled classifiers
14 Convergence within each sampling iteration
15 A neighbourhood-preserving embedding encodes the sampling design
16 Decisive two-class tests (Classes 1/2, 3/4 and 7/8)
17 Per-class feature contributions within each sampling iteration
18 Per-class attribution to the 14 parameters
19 Transition map of the ten classes
20 Two-class switches
21 Breaking the symmetry removes the branch point
22 Open circles: detached branches beyond the labelling window
23 Misclassified test sets whose detached branch lies beyond the window
24 Normal-form coefficients at every branch point and fold
25 ROC curves and convergence of the pooled ten-class classifiers
26 Representative misclassified parameter sets

Supplementary Tables

1 Structural analysis with the Chemical Reaction Network Toolbox
2 Exact labelling: steady-state structure of each sampling iteration
3 Validation of the exact labels against MATCONT
4 Normal-form coefficients in each sampling iteration
5 Training, validation and test sets
6 Feature sets
7 Features kept by JSD and the broken stick within each iteration
8 Model comparison: cross-validated and held-out accuracy
9 Area under the ROC curve of every model
10 Per-class performance of the pooled LightGBM classifier
11 Convergence of the boosted models and the random forest
12 Embedding audit
13 Decisive two-class tests
14 All confusions of the pooled LightGBM classifier
15 Computational cost of the labelling
16 Supplementary Files
17 Code used for each figure and table
18 Five largest per-class contributions in each sampling iteration
19 Edges of the transition map
20 Two-class sweeps
21 Classifier errors from the labelling window
22 Dependence of the multistationary fraction on the A_{TOT} window
23 EMBED and the present analysis
24 Single-start MATCONT, the inverse method and MATCONT seeded on every branch
25 Normal-form coefficients at every singularity of the exact branches
26 Design-constrained feature ratios
27 Number of input features and LightGBM accuracy for each feature set
28 MATCONT continuation settings
29 Software dependencies
30 Experimentally accessible quantities that set the class

Supplementary Note 1. Structural analysis of the reaction network

The network of main-text Fig 1 [I] (eight reactions, nine species) was entered in the Chemical Reaction Network Toolbox, version 2.35, as the file `MAPKFULL.NET`; the Basic Report and the Higher Deficiency Report are given as Supplementary File 1. Supplementary Table 1 summarises both reports. The network has ten complexes in two linkage classes, one for the kinase and one for the phosphatase reactions; its stoichiometric subspace has dimension six, so there are three conservation laws (the totals of substrate, kinase and phosphatase, main-text Eqs (3)–(5)). Its deficiency is two, while each linkage class has deficiency zero. The conservation laws and the deficiency were verified independently in Wolfram Mathematica 15 (Supplementary File 6, `Analytical_Derivations_StepByStep_final.nb`, Section 2). Because the network is not weakly reversible, neither the Deficiency Zero nor the Deficiency One Theorem applies, and the question of multistationarity was settled with higher deficiency theory. The Higher Deficiency Report returns a set of mass-action rate constants with two positive steady states in the same stoichiometric compatibility class, one asymptotically stable and one unstable (one positive eigenvalue). The totals of these two states coincide ($A_{\mathrm{TOT}} = 18.314$, $K_{\mathrm{TOT}} = 5.283$, $P_{\mathrm{TOT}} = 9.140$), and the parametrisation of main-text Eqs (7)–(9) (derived in Mathematica, Supplementary File 6, `Analytical_Derivations_StepByStep_final.nb`, Sections 5 and 6), applied to the same rate constants, recovers both states together with a third, asymptotically stable state ($APP = 1.332$), so that the example is bistable (main-text Methods). The eigenvalues printed by the Toolbox differ from those of our Jacobian by a common factor of about 8.4, which corresponds to a rescaling of time and leaves the signs, and hence the stability, unchanged. The structural analysis establishes only the capacity for multistationarity; the exact steady-state analysis of the ensemble (Supplementary Notes 2 and 3) shows in which regions of parameter space it is realised.

Supplementary Table 1. Structural analysis with the Chemical Reaction Network Toolbox (version 2.35). Summary of the Basic Report and the Higher Deficiency Report (Supplementary File 1). Code: CRNToolbox reports (Supplementary File 1).

Property	Result
Species, reactions	9 species; 8 reactions (4 reversible binding steps and 4 irreversible catalytic steps; 12 reaction directions)
Complexes	10
Linkage classes	2: {A+K, AK, AP+K, APK, APP+K} and {APP+P, APPC, AP+P, APC, A+P}
Strong linkage classes	6 (2 terminal: {APP+K}, {A+P})
Reversibility	neither reversible nor weakly reversible
Rank of the network	6 (three conservation laws)
Deficiency	2 (0 for each linkage class)
Positive steady states	possible (positively dependent reaction vectors)
Conservative	yes
Capacity for multiple positive steady states	yes (mass-action kinetics)
Degenerate steady state (zero eigenvalue in the stoichiometric subspace)	possible
Example rate constants	$k_1 = 12.463$, $k_{11} = k_2 = 30.061$, $k_3 = 25.087$, $k_{33} = k_4 = 1.910$, $k_5 = 898.91$, $k_{55} = k_6 = 1$, $k_7 = 782.52$, $k_{77} = k_8 = 12.438$
Steady states of the example	$APP = 1.007$ (unstable; one eigenvalue $+2.9 \times 10^{-3}$) and $APP = 6.8 \times 10^{-3}$ (asymptotically stable); parametrisation: a third, stable state with $APP = 1.332$

Supplementary Note 2. Exact labelling of the four sampling iterations

Each sampling iteration contains 10,000 parameter sets drawn by log-uniform Latin hypercube sampling (seed 20260930) within the bounds and equality constraints of main-text Table 1, and every set was labelled from all of its steady states for $A_{\mathrm{TOT}} \in [0, 10^6]$ with the inverse method (main-text Fig 3 and Methods). Supplementary Table 2 lists, for each iteration, the quantities on which the ten classes are based.

Three features of the table are used in the main text. First, every multistationary parameter set has exactly three steady states at some A_{TOT}, two of which are stable (the bistable and multistationary fractions are identical in every iteration); no set has five, in agreement with the bound of $2n - 1 = 3$ for the ordered two-site cycle. Second, in the asymmetric iterations most multistationary sets carry a fold on a detached branch (51.8% and 83.1% of all sets), and the A_{TOT} of these folds spans five orders of magnitude, with medians of a few hundred; this is why single-start continuation, which stops near $A_{\mathrm{TOT}} \approx 600$ and never leaves the main branch, undercounts multistationarity (Supplementary Note 5). Third, in the symmetric iterations all 5,002 multistationary sets have a branch point, and the fraction of multistationary sets (50.0%) equals the fraction of sampled sets with $k_4 > k_2$ (main text, branch-point criterion). In Iteration 3 the branch point is classified as a pitchfork for 4,997 of the 5,002 sets; for the remaining five all branching coefficients are below 10^{-8}, so that the numerical test is not decisive, and they are labelled BP only like the others. In 31 sets of Iteration 3 and one set of Iteration 4 a fold on the crossing branch coincides with the branch point; these sets are flagged. The 32 flagged sets are kept in all tables and excluded from machine learning.

Supplementary Table 2. Exact labelling: steady-state structure of each sampling iteration ($A_{\mathrm{TOT}} \leq 10^6$). Multistationary: three steady states for some A_{TOT} in the window (Classes 1–8); 95% Wilson confidence intervals in brackets. Fold A_{TOT}: median and interquartile range of the fold on the detached branch. Crossing branch: the branch $A = 1/(\alpha u)$ that crosses the $u = 1$ branch at the branch point. Flagged: a fold coincides with the branch point to within 10^{-9}. Cost: mean (median) time per parameter set for the exact labelling including normal forms, one core. Iterations 1–4: kinase-dominant, phosphatase-dominant, full symmetry, first-binding asymmetry (main-text Table 1). Code: `Ensemble_LHS_v3.m` (folder A).

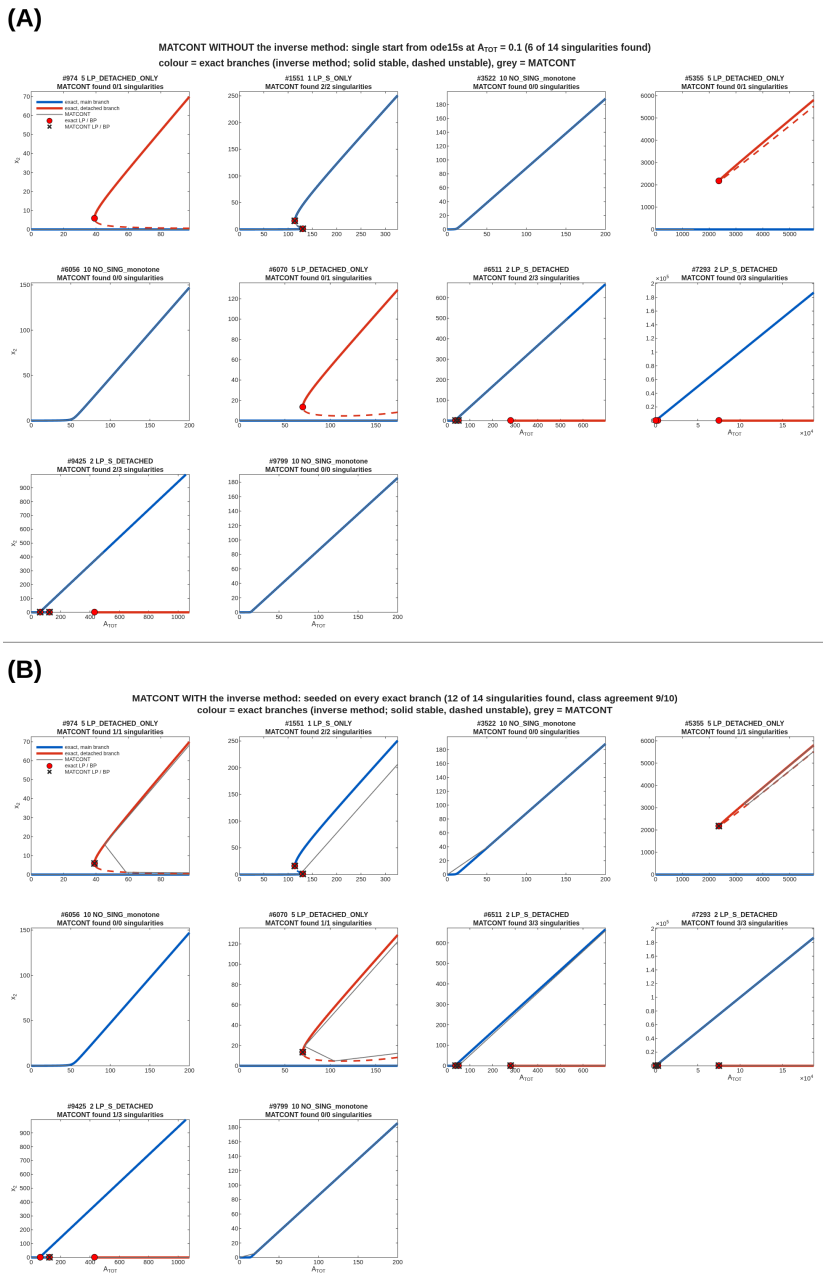
	Iter. 1	Iter. 2	Iter. 3	Iter. 4
Parameter sets	10,000	10,000	10,000	10,000
One steady state for all A_{TOT}	2,790	1,217	4,998	4,998
Up to three steady states	7,210	8,783	5,002	5,002
Multistationary (%)	72.1 (71.2–73.0)	87.8 (87.2–88.5)	50.0 (49.0–51.0)	50.0 (49.0–51.0)
Detached branch with fold in window (%)	51.8	83.1	0	0
Fold A_{TOT} on detached branch	253 (84–2,330)	504 (103–8,180)	–	–
Folds on the main branch	6,730	4,768	0	0
Folds on a detached branch	5,178	8,305	0	0
Folds on the crossing branch	0	0	31	5,002
Branch points	0	0	5,002	5,002
of which pitchforks	–	–	4,997	0
Flagged sets (LP = BP)	0	0	31	1
Used for machine learning	10,000	10,000	9,969	9,999
Cost per set, ms	14.3 (11.0)	12.2 (11.5)	9.7 (9.6)	9.2 (9.9)

Supplementary Note 3. Validation of the exact labels against MATCONT

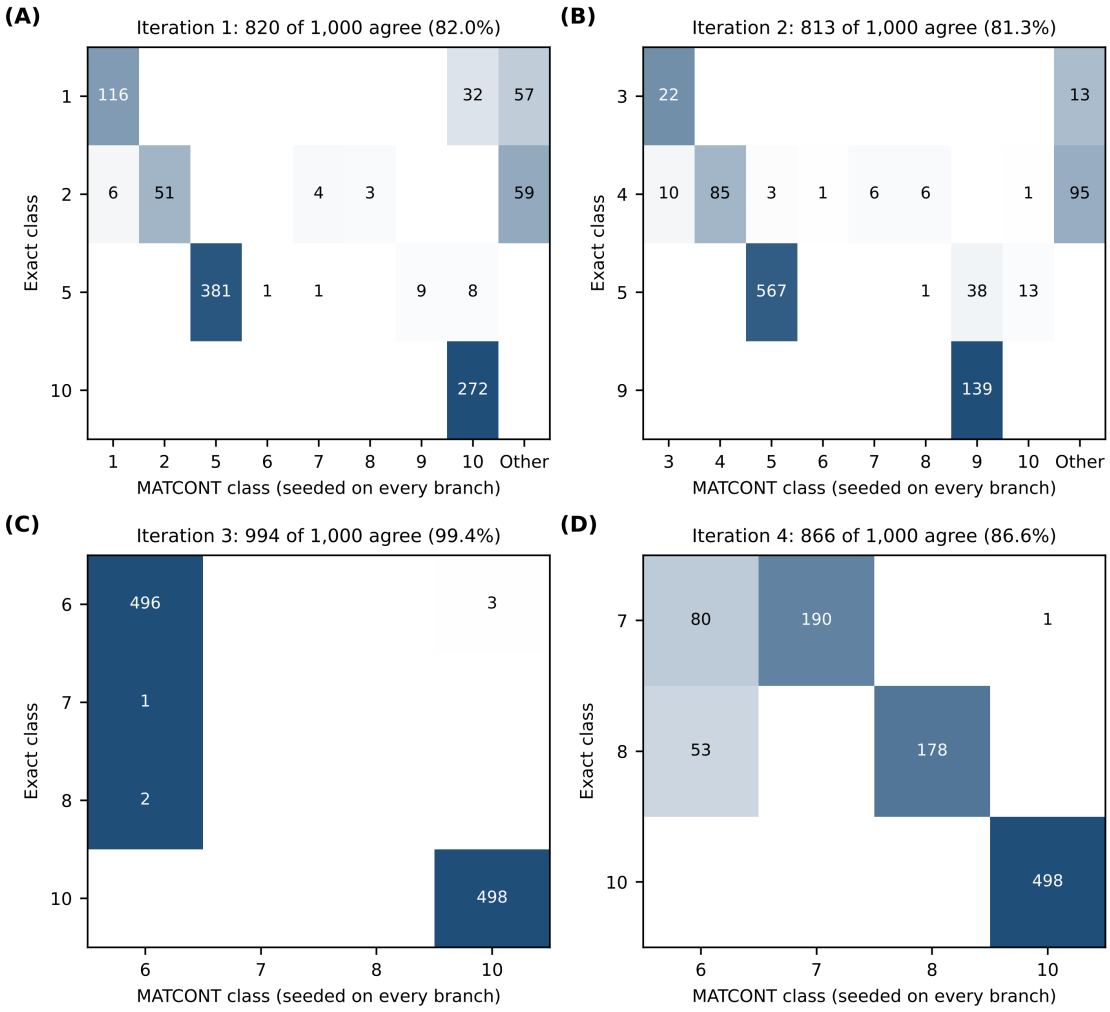
The exact labels were validated in two ways (Supplementary Table 3). In every iteration, a random subsample of 1,000 parameter sets was continued with MATCONT 7.6 seeded on a stable steady state of every branch found by the inverse method (including detached and crossing branches), in both directions and trimmed to the same window (settings: Supplementary Table 28); the MATCONT singularities were classified with the same rules as the exact ones. In addition, for ten parameter sets per iteration, MATCONT was also run without the inverse method, i.e. from a single start obtained by integrating the ODEs at $A_{\mathrm{TOT}} = 0.1$ (Supplementary Fig. 1).

Agreement. Over the 4,000 validated sets MATCONT gives the same class as the exact labels for 3,493 sets (87.3%; 95% Wilson CI 86.3–88.3%; Supplementary Fig. 2). MATCONT never reports a singularity in a set that has none (specificity 100% in every iteration), and it finds at least one singularity in 93.3–99.8% of the multistationary sets. Every disagreement is a singularity that MATCONT misses or does not resolve, never one that the exact computation misses: (i) in Iterations 1 and 2, main-branch hysteresis that MATCONT assigns to its residual class ("other", 116 and 108 sets), S-shaped hysteresis that MATCONT labels monotone (32 sets) and detached folds that it does not reach (Class 5 labelled biphasic or monotone, 17 and 51 sets); (ii) in Iteration 4, the fold on the crossing branch when it lies close to the branch point (133 sets labelled BP only by MATCONT); (iii) in Iteration 3, three flagged LP+BP sets labelled BP only and three BP-only sets labelled monotone. At the 2,422 folds of the validation subsample at which MATCONT reports its normal-form coefficient, our coefficient a_{LP} agrees with MATCONT's in absolute value to a median relative difference of 1.4×10^{-8} (95th percentile 1.7×10^{-7}); the sign depends on the orientation of the null vectors, which differs between the two computations.

Single start. For the 40 parameter sets of the ten-set checks, MATCONT seeded on every branch finds 38 of the 43 exact singularities and agrees with the exact class for 36 sets, whereas a single start at $A_{\mathrm{TOT}} = 0.1$ finds 26, none of them on a detached branch (13 detached folds in Iterations 1 and 2), because a run that starts on the main branch cannot reach a branch that is not connected to it.



Supplementary Fig. 1. MATCONT with and without the inverse method. Ten random parameter sets of Iteration 1. Coloured: exact branches (blue, main branch; red, detached branch; solid, stable; dashed, unstable; red circles, exact folds); grey: MATCONT; crosses, singularities found by MATCONT. Each panel gives the sample number, the exact class and the number of exact singularities found by MATCONT. (A) Single start: integration to a steady state at $A_{\mathrm{TOT}} = 0.1$ and one continuation in A_{TOT} (6 of 14 singularities found; no detached branch reached). (B) MATCONT seeded on every exact branch (12 of 14 singularities found; the same class for 9 of the 10 sets). Code: `Ensemble_LHS_v3.m` (MATCONT check; folder A).



Supplementary Fig. 2. Exact labels against MATCONT seeded on every branch. Confusion matrices of the exact class (rows) and the MATCONT class (columns) for the 1,000 validated parameter sets of each iteration; numbers are counts, shading the percentage of the row. Only classes that occur in an iteration are shown; "Other": MATCONT's residual class. (A) Iteration 1. (B) Iteration 2. (C) Iteration 3. (D) Iteration 4. Code: `Ensemble_LHS_v3.m` (folder A).

Supplementary Table 3. Validation of the exact labels against MATCONT. Upper part: 1,000 random parameter sets per iteration, MATCONT seeded on every branch. Sensitivity: multistationary sets in which MATCONT finds at least one singularity; specificity: sets without singularity in which MATCONT finds none; 95% Wilson confidence intervals in brackets. "Other": MATCONT's residual class. Lower part: ten parameter sets per iteration continued both ways. Code: `Ensemble_LHS_v3.m` (folder A).

	Iter. 1	Iter. 2	Iter. 3	Iter. 4
Validated sets	1,000	1,000	1,000	1,000
Same class (%)	82.0 (79.5–84.3)	81.3 (78.8–83.6)	99.4 (98.7–99.7)	86.6 (84.3–88.6)
Sensitivity (%)	93.3 (679/728)	94.0 (809/861)	99.4 (499/502)	99.8 (501/502)
Specificity (%)	100 (272/272)	100 (139/139)	100 (498/498)	100 (498/498)
MATCONT "other"	116	108	0	0
Folds with a MATCONT coefficient	953	1,101	0	368
Median rel. difference of $\lvert a_{LP}\rvert$	1.4×10^{-8}	1.1×10^{-8}	–	4.4×10^{-8}
Exact singularities (10 sets)	14	11	6	12
Found, MATCONT seeded on every branch	12	10	6	10
Found, MATCONT from a single start	6	4	6	10
Same class, seeded on every branch	9/10	9/10	10/10	8/10
A_{TOT} reached by single start (median)	1,062	1,298	1,447	1,420

Supplementary Note 4. Normal-form coefficients in each sampling iteration

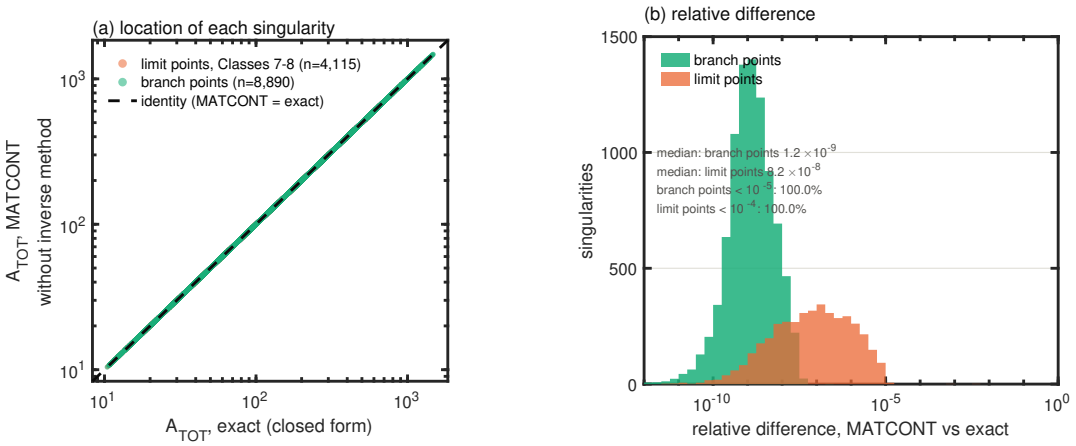
At every fold and branch point inside the window the normal-form coefficients were computed from the analytic first, second and third derivatives of the right-hand side, in MATCONT's normalisation (main-text Methods); the analytic Jacobian was derived in Mathematica and the symbolic definitions of the coefficients are given there (Supplementary File 6, `Analytical_Derivations_StepByStep_final.nb`, Sections 3 and 9). Supplementary Table 25 gives the ranges over the whole ensemble and Supplementary Fig. 24 their distributions; Supplementary Table 4 separates the four iterations. All 30,014 folds are non-degenerate ($a_{LP} \neq 0$; the smallest $|a_{LP}|$ of the asymmetric iterations is 3.9×10^{-9}). The 31 crossing-branch folds of Iteration 3 have $|a_{LP}| \leq 10^{-12}$, i.e. they are numerically indistinguishable from the pitchfork branch point with which they coincide; these are the flagged sets. The branch points of Iteration 3 have $|a_{BP}| \leq 3 \times 10^{-14}$ (pitchfork, by symmetry), those of Iteration 4 a non-zero quadratic coefficient (median 5.2×10^{-5}, up to 0.12; transcritical); in both iterations the discriminant of the branching equation is positive at every branch point, i.e. two branches cross transversally.

Supplementary Table 4. Normal-form coefficients in each sampling iteration. a_{LP}: quadratic coefficient of the fold normal form; β: derivative with respect to A_{TOT}; a_{BP}: quadratic coefficient of the branching equation. Absolute values; median and range. Code: `Ensemble_LHS_v3.m` (`support/normal_forms.m`), `Manuscript_numbers_v3.m` (folder A).

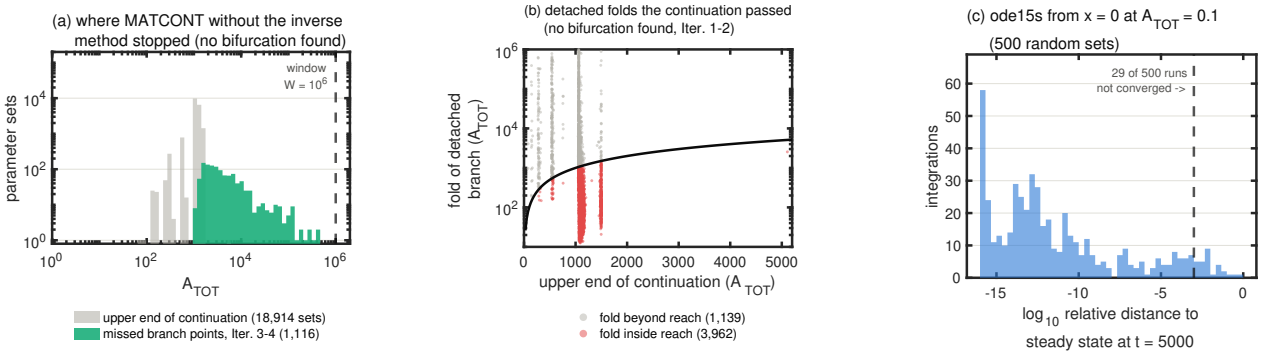
	Iter. 1	Iter. 2	Iter. 3	Iter. 4
Folds	11,908	13,073	31	5,002
$\lvert a_{LP}\rvert$, median	3.0×10^{-4}	3.3×10^{-4}	1.1×10^{-14}	2.2×10^{-5}
$\lvert a_{LP}\rvert$, range	3.9×10^{-9}–0.091	4.6×10^{-9}–0.074	3×10^{-16}–9×10^{-13}	3.1×10^{-14}–0.057
$\lvert\beta\rvert$, median	1.6×10^{-3}	9.0×10^{-4}	2.7×10^{-9}	2.5×10^{-3}
Branch points	0	0	5,002	5,002
$\lvert a_{BP}\rvert$, median (max.)	–	–	1×10^{-17} (3×10^{-14})	5.2×10^{-5} (0.12)
Positive discriminant	–	–	5,002	5,002

Supplementary Note 5. Comparison with single-start continuation

Main-text Fig 5 and Supplementary Tables 23 and 24 compare the exact structure with labels obtained by single-start continuation, the workflow of our earlier EMBED framework applied to the full mass-action model: integration of the ODEs to a steady state at $A_{\mathrm{TOT}} = 0.1$, one continuation in A_{TOT} with branch switching at every branch point, and a biphasic test on the computed branch (main-text Methods, "Single-start MATCONT labels"). Single-start continuation was run on the same 40,000 parameter sets that carry the exact labels (`SingleStart_v3.m`, Supplementary File 2). For 362 sets (119, 241, 2 and 0 in Iterations 1–4) the run ended on negative concentrations or failed. Main-text Fig 5 shows all 10,000 sets of each iteration, with these runs as a separate bar, against the exact labels of main-text Fig 2; main-branch hysteresis is merged to the resolution of single-start continuation (Classes 1+2 and 3+4). Counting a failed run as no singularity found, single-start continuation finds a fold or branch point in 27.4%, 19.1%, 44.8% and 44.1% of the sets of Iterations 1–4, the exact labels in 72.1%, 87.8%, 50.0% and 50.0%. The detailed analyses below (where the runs stopped, which detached folds they passed) use the 39,638 sets for which the run completed (9,881, 9,759, 9,998 and 10,000), labelled exactly with an independent implementation of the parametrisation (`make_comparison_figures.m`, Supplementary File 4). Single-start continuation never reports a singularity in a set whose exact label has none; all differences are omissions. Main-text Fig 3A and Supplementary Figs 3 and 4 show the three findings on which the main-text conclusion rests: single-start continuation misses detached branches by construction (main-text Fig 3A); where it finds a singularity, it locates it accurately (Supplementary Fig. 3); and its omissions follow from where the runs start and where they stop (Supplementary Fig. 4). The ten-set checks of Supplementary Note 3 reproduce the same behaviour with the ten-class pipeline itself (Supplementary Fig. 1A).



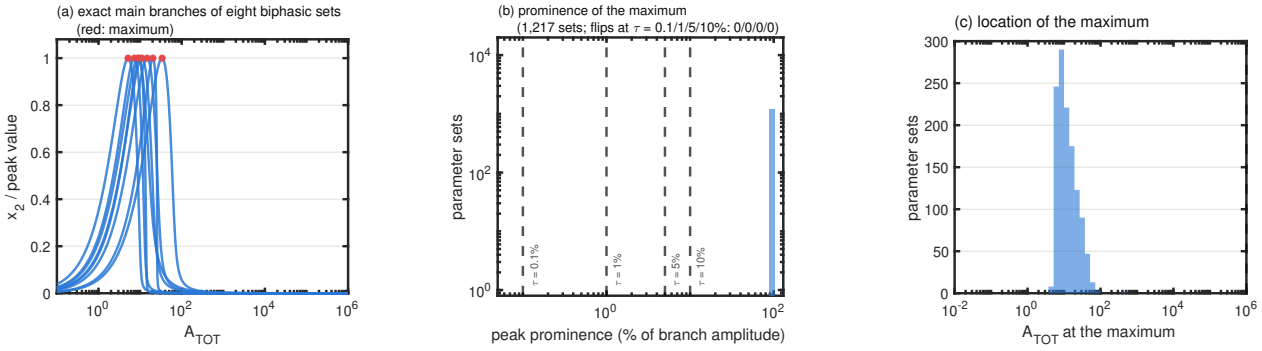
Supplementary Fig. 3. Singularities found by continuation agree with the exact computation. (a) A_{TOT} of every branch point (green; $n = 8{,}890$) and of every fold of coexisting LP and BP (orange; $n = 4{,}115$) located by single-start MATCONT, against the exact value; dashed, identity. (b) Distribution of the relative difference $|A_{\mathrm{TOT}}^{\mathrm{MATCONT}}/A_{\mathrm{TOT}}^{\mathrm{exact}} - 1|$. All branch points agree to better than 10^{-5} (median 1.2×10^{-9}), and 99.95% of the limit points to better than 10^{-4} (median 8.2×10^{-8}). Code: `make_comparison_figures.m` (folder B).



Supplementary Fig. 4. Limits of single-start continuation. (a) Upper end of the A_{TOT} range reached for the 18,914 parameter sets labelled "no bifurcation" by single-start continuation (grey; median 1,129), and the location of the 1,116 branch points in Iterations 3 and 4 that these runs missed (green); dashed, the fixed window $W = 10^6$ of the exact labels. (b) For single-start "no bifurcation" sets of Iterations 1 and 2 that have a detached branch: A_{TOT} of the fold of the detached branch against the upper end of the continuation. Red points lie below the line fold = reach (3,962 of 5,101 sets), i.e. the continuation passed through a bistable region without detecting it; grey points (1,139) have the fold beyond the reach. (c) Relative distance between the state reached by integrating from $x = 0$ at $A_{\mathrm{TOT}} = 0.1$ up to $t = 5000$ (`ode15s`, relative tolerance 10^{-8}) and the exact stable steady state, for 500 random parameter sets; 29 runs had not converged (relative distance $> 10^{-3}$), and only 7 of these have $|\dot{x}|_\infty > 10^{-6}$ at $t = 5000$. Code: `make_comparison_figures.m` (folder B).

Supplementary Note 6. The biphasic criterion

A parameter set without fold or branch point in the window is labelled biphasic (Class 9) when x_2 has exactly one interior maximum along the main branch (main-text Methods). To test whether this label depends on how small a maximum is accepted, a minimum peak prominence τ was imposed, as a fraction of the amplitude of x_2 along the branch. None of the 1,217 biphasic sets changes label for τ between 0.1% and 10%: the smallest prominence is 99.99% of the amplitude, i.e. the maximum is always the global maximum and the response returns almost to zero. The maxima lie far inside the window (median $A_{\mathrm{TOT}} \approx 11$; 5–95% range 6–38; Supplementary Fig. 5).



Supplementary Fig. 5. The biphasic criterion is insensitive to the prominence threshold. (a) Exact main branches of eight randomly chosen biphasic parameter sets, with x_2 scaled to its maximum (red). (b) Peak prominence, as a percentage of the amplitude of x_2 along the branch, for all 1,217 biphasic sets; dashed lines mark thresholds τ of 0.1, 1, 5 and 10%; no set changes label at any of them. (c) A_{TOT} at the maximum; all maxima lie far inside the window. Code: `make_comparison_figures.m` (folder B).

Supplementary Note 7. The sampling design and its imprint on the features

The four iterations were designed to reach the four regimes of the cycle (main-text Table 1): two unconstrained regimes, dominated by the kinase or by the phosphatase, and two symmetric regimes in which seven ratios of corresponding kinase and phosphatase parameters are fixed to one (all seven in Iteration 3; five in Iteration 4, where the first-binding constants are free). As a consequence the pattern of these seven ratios identifies the iteration of every parameter set: four distinct patterns occur, and they recover the iteration for 100% of the sets; 50.0% of all sets have at least four ratios exactly equal to one and 24.9% have all seven (Supplementary Table 26). Trained to predict the iteration instead of the class, LightGBM reaches 100.00% (baseline 25.02%). Because eight of the ten classes occur in a single iteration (main-text Fig 1[IV] and Table 2), a pooled classifier can partly predict the class by recognising the iteration. Three controls show that this does not explain the classifier's performance: removing the seven constrained ratios lowers pooled LightGBM accuracy by only 1.2 points (feature set B, Supplementary Table 27); within each iteration, where the constrained ratios are constant and are removed automatically, the classifier still reaches 84.5–100% (Supplementary Table 27); and the features ranked highest within each iteration are exactly those of the analytical criteria (main-text Fig 10A and main-text Table 5). Mechanistic conclusions are therefore drawn only from within-iteration analyses, and the pooled feature ranking (Supplementary Fig. 10) is not interpreted mechanistically.

Supplementary Note 8. Machine-learning data and splits

The 39,968 unflagged parameter sets were split, stratified jointly by iteration and class, into training (70%), validation (10%) and held-out test (20%) sets (Supplementary Table 5). Models are fitted on the training rows only; the validation rows are used only for the convergence curves (Supplementary Note 11) and every reported accuracy refers to the test rows. Training samples are weighted inversely to the frequency of their class. The classes differ in size by a factor of 27 (Class 3, 478 sets; Class 10, 12,786), so balanced accuracy and macro-F1 are reported next to the accuracy throughout. Feature sets, model settings and the evaluation protocol are described in the main-text Methods.

Supplementary Table 5. Training, validation and test sets (pooled ten-class data, first split, seed 43). Number of parameter sets per class. Code: `ML_v3_10class_parallel.m` (folder A).

Class	1	2	3	4	5	6	7	8	9	10	Total
Training	1,423	933	334	1,334	7,170	3,480	1,794	1,707	852	8,949	27,976
Validation	203	133	48	191	1,025	497	256	244	122	1,279	3,998
Test	406	267	96	381	2,049	994	512	488	243	2,558	7,994
Total	2,032	1,333	478	1,906	10,244	4,971	2,562	2,439	1,217	12,786	39,968

Supplementary Note 9. Feature engineering and feature selection

This note gives the details behind main-text Table 3: why the classifiers are given engineered features, how the six feature sets were built and selected, and what their comparison shows. The features and the selection are computed by `ML_v3_feature_engineering.m` (folder A) with the same functions as the classifiers (`support/ml_v3_core.m`).

Why features were engineered. The only inputs are the 14 sampled quantities (K_{TOT}, P_{TOT} and the twelve rate constants). The exact analysis shows that the class boundaries are conditions on products and ratios of a few of them: criterion (10) compares $k_8 P_{\mathrm{TOT}}$ with $k_2 K_{\mathrm{TOT}}$ and $k_4 K_{\mathrm{TOT}}$ with $k_6 P_{\mathrm{TOT}}$, under kinase–phosphatase symmetry a branch point exists if and only if $k_4 > k_2$, and the side of the transcritical crossing on which the fold lies is set by $K_{M5} - K_{M1}$. On a logarithmic scale such conditions become linear: $\log_{10}(k_8 P_{\mathrm{TOT}}/k_2 K_{\mathrm{TOT}}) = \log_{10} k_8 + \log_{10} P_{\mathrm{TOT}} - \log_{10} k_2 - \log_{10} K_{\mathrm{TOT}}$. A decision tree splits on one feature at a time, so a boundary that is oblique to the parameter axes can only be approximated by a staircase of many splits, whereas a log-ratio of the two parameters turns a two-parameter boundary into a single threshold. The features were therefore engineered with a specific aim: to give the tree ensembles coordinates in which boundaries of the kind predicted by the mechanism are aligned with the axes, without telling them which ratios matter. All features are computed from $\log_{10}$ of the parameters: the 14 log-parameters and all $\binom{14}{2} = 91$ pairwise log-ratios, $\log_{10}(p_i/p_j) = \log_{10} p_i - \log_{10} p_j$ (feature set A, 105 features; main-text Fig 8, panel 2). Ratios of three or more parameters, such as the four-parameter ratios of criterion (10), were deliberately not included: the classifier has to combine pairwise ratios to reach them, so that a feature ranking that points to these ratios is evidence rather than input. No quantity derived from the labels (bistability, presence of a detached branch, singularity locations) is ever used as a feature.

Six feature sets, each answering one question. After the reviewers' comments, the single feature set of the submitted version (JSD selection on parameters and ratios) was replaced by six feature sets that test the contribution of each step of the pipeline (Supplementary Table 6). Set B removes the seven ratios that the sampling design fixes to one in Iterations 3 and 4 ($K_{\mathrm{TOT}}/P_{\mathrm{TOT}}$, k_1/k_5, k_{11}/k_{55}, k_2/k_6, k_3/k_7, k_{33}/k_{77}, k_4/k_8; Supplementary Table 26): it asks whether the classifier merely

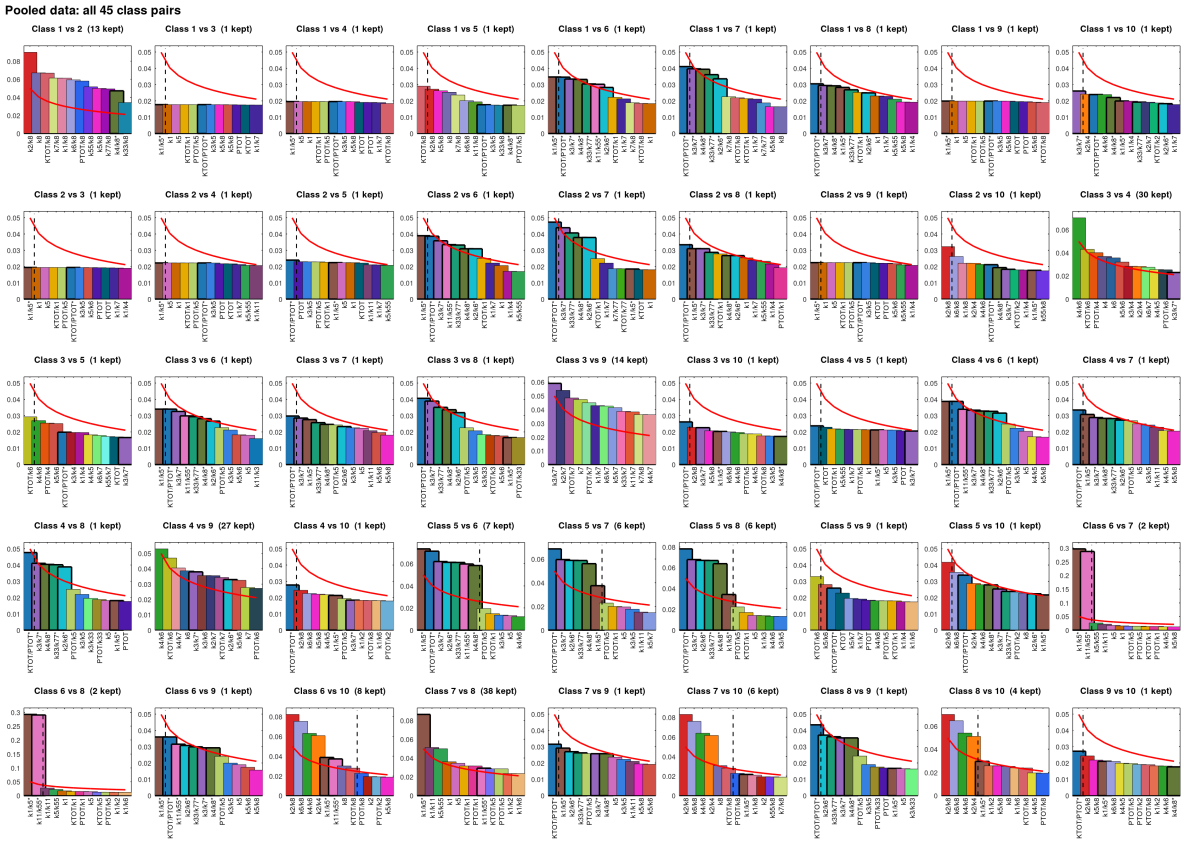
recognises the sampling iteration. Set C keeps only the 14 log-parameters: it asks whether feature engineering is needed at all. Set D applies the divergence-based selection of the submitted version to set A: it asks whether filtering the features helps. Sets E and F use centred log-ratios, a scale-free (compositional) representation of the parameters; F keeps $\log_{10}(K_{\mathrm{TOT}}/P_{\mathrm{TOT}})$ separate because the enzyme totals and the rate constants have different units. Before training, columns with zero variance are removed and exactly duplicated columns are collapsed to one; within Iterations 3 and 4 the symmetry constraints make many ratios constant or identical, which reduces set A from 105 to 43 and 59 features.

Supplementary Table 6. Feature sets. Number of features used for the pooled data and within Iterations 1–4 (after removal of constant and duplicated columns; for set D the mean number selected over the three training splits), the question that each set addresses, and the held-out LightGBM accuracy on the pooled ten-class data (main-text Fig 9A and Supplementary Table 27). Code: `ML_v3_feature_engineering.m` (feature sets), `ML_v3_10class_parallel.m` (accuracies) (folder A).

Set	Construction	Features: pooled; Iter. 1–4	Question	Acc. (%)
A	14 log-parameters and 91 pairwise log-ratios	105; 105, 105, 43, 59	reference: all engineered features	92.66
B	A without the 7 design-constrained ratios	98; 98, 98, 43, 57	does the classifier only recognise the sampling iteration?	91.43
C	14 log-parameters	14; 14, 14, 7, 9	is feature engineering needed?	89.13
D	A, selected by JSD and broken stick	64.3; 18.0, 36.3, 2.0, 5.0	does filter selection help (submitted pipeline)?	92.71
E	centred log-ratios of the 14 log-parameters	14; 14, 14, 7, 9	does a scale-free representation help?	88.79
F	centred log-ratios of the 12 rate constants and $\log_{10}(K_{\mathrm{TOT}}/P_{\mathrm{TOT}})$	13; 13, 13, 6, 8	as E, with the enzyme totals kept apart	89.77

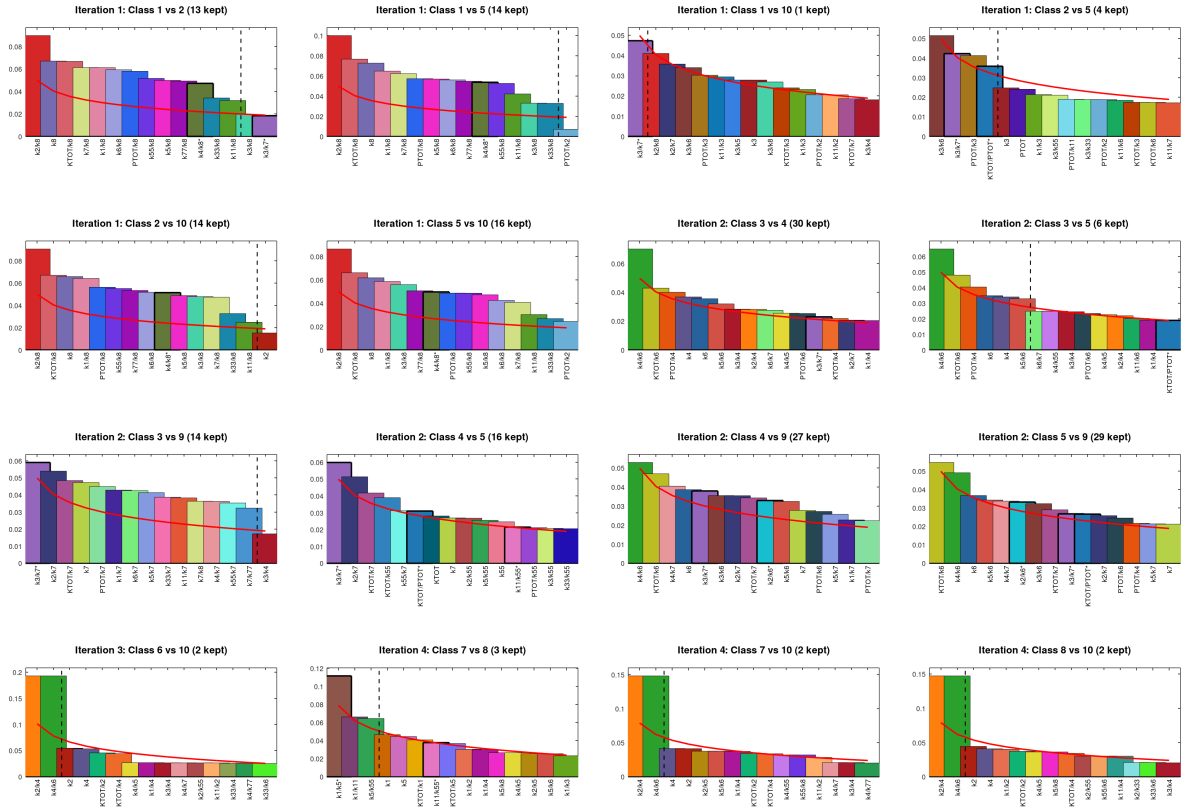
Feature selection by Jensen–Shannon divergence and the broken stick. For every pair of classes, every feature is scored by the Jensen–Shannon divergence (JSD, natural logarithm) between its two class-conditional distributions, estimated from 30-bin histograms over the common range of the two classes. The JSD is zero for identical distributions and reaches its maximum, $\ln 2 \approx 0.693$, when the two classes do not overlap. The features are ranked by JSD, their scores are normalised to sum to one, and the ranked scores are compared with the broken-stick expectation $b_k = n^{-1} \sum_{i=k}^{n} 1/i$ for n features; the features ranked before the first rank at which the normalised score falls below b_k are kept. This cut-off adapts to the shape of the JSD spectrum instead of using a fixed threshold. The union of the features kept over all class pairs is the selected set. The classifiers recompute the selection on the training rows of every split (main-text Methods); the figures below apply it to all 39,968 sets of each scope. Of the 105 features, it keeps 76 in the pooled data (8 parameters, 68 log-ratios), 20 in Iteration 1, 47 in Iteration 2, 2 of 43 in Iteration 3 and 5 of 59 in Iteration 4 (Supplementary Figs 6–8).

In the pooled data the selection is driven by the sampling design. Of the 45 pairs of classes, 29 never occur in the same iteration, and for these the two classes are separated, often perfectly (JSD at or close to $\ln 2$), by a ratio that the design fixes to one in one of the two iterations (Supplementary Fig. 6). When many features separate two classes equally well the JSD spectrum is flat, and the broken stick keeps a single, essentially arbitrary feature: 32 of the 45 pairs keep one feature. In



Supplementary Fig. 6. JSD and broken-stick selection for all 45 class pairs of the pooled data. For each pair of classes, the 12 features with the largest Jensen–Shannon divergence between the two classes, ranked; bars, normalised JSD (share of the total over all 105 features), coloured by feature (the same colours as in every importance plot); thick outline and asterisk, design-constrained ratio; red curve, broken-stick expectation; dashed line, elbow, i.e. the features to its left are kept. Panel titles give the number of features kept. Code: `ML_v3_feature_engineering.m` (folder A).

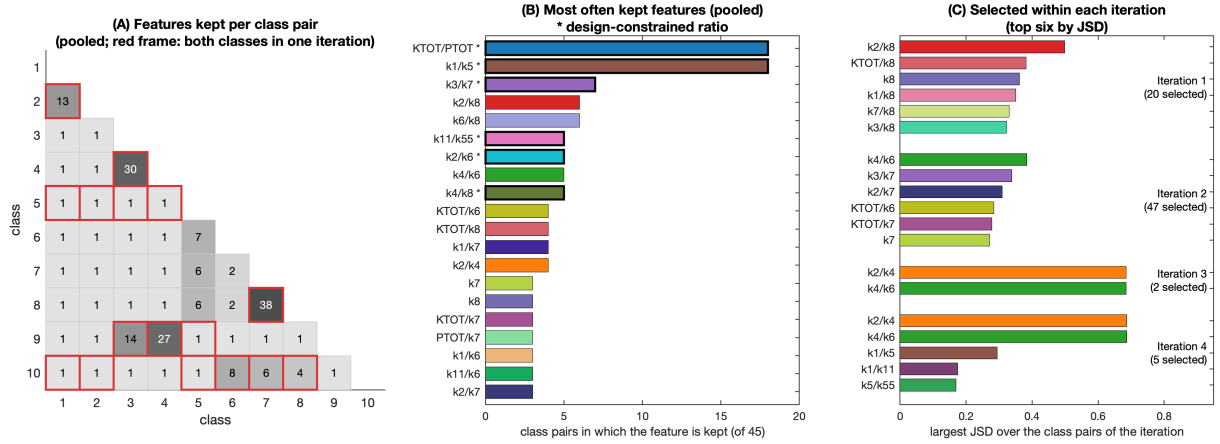
every one of the 29 pairs whose classes never meet in an iteration, every feature kept is a design-constrained ratio, and $K_{\mathrm{TOT}}/P_{\mathrm{TOT}}$ and k_1/k_5 are kept in 18 pairs each, more than any other feature (Supplementary Fig. 8A,B). Divergence-based selection therefore does not remove the imprint of the sampling design; it concentrates on it. This is why the analysis was extended beyond set D of the submitted version: set B removes the constrained ratios, and all mechanistic interpretation is based on analyses within each iteration, where these ratios carry no information (Supplementary Note 7).



Supplementary Fig. 7. JSD and broken-stick selection within each sampling iteration. The 16 pairs of classes that occur together in an iteration (Iteration 1: Classes 1, 2, 5, 10; Iteration 2: 3, 4, 5, 9; Iteration 3: 6, 10; Iteration 4: 7, 8, 10), each with the 15 features of largest JSD; columns that are constant or duplicated within the iteration are removed first. An asterisk marks a ratio that is fixed by the design in Iterations 3 or 4; within Iterations 1 and 2 these ratios are free. Bars, curves and lines as in Supplementary Fig. 6. Code: `ML_v3_feature_engineering.m` (folder A).

Within each iteration the selection recovers the analytical conditions (Supplementary Fig. 7 and Supplementary Table 7). In Iteration 3 only two features are kept, k_2/k_4 and k_4/k_6 (identical up to sign because $k_2 = k_6$; JSD 0.685), which express the branch-point condition $k_4 > k_2$. In Iteration 4 the same two ratios separate Classes 7 and 8 from Class 10, and Classes 7 and 8 are separated by k_1/k_5, k_1/k_{11} and k_5/k_{55}, the binding and unbinding constants that enter K_{M1} and K_{M5} (criterion (14)). In Iteration 1, k_2/k_8 has the largest JSD (0.49–0.50) in every pair that differs in the presence of a detached branch (1 against 2 and 5; 2 and 5 against 10): it is the site-1 part of criterion (10). In Iteration 2, k_4/k_6, the site-2 part, ranks first for Class 3 against Classes 4 and 5 and for Class 4 against Class 9, and k_3/k_7, the competition of kinase and phosphatase for the singly

phosphorylated substrate, for Class 3 against Class 9. The pairs with the smallest JSD are those separated only by the one boundary without a closed form, whether the main branch carries a fold: Class 1 against 10 (largest JSD 0.12), 2 against 5 (0.06) and 4 against 5 (0.05). No single feature separates these classes, and they are where the classifiers make most of their errors (Supplementary Note 14).



Supplementary Fig. 8. Summary of the divergence-based selection. (A) Number of features kept for each pair of classes in the pooled data; red frame, the two classes occur together in at least one iteration. (B) The 20 features kept most often over the 45 pairs (pooled); asterisk and thick outline, design-constrained ratio. (C) Within each iteration: the six selected features with the largest JSD (the largest value over the class pairs of the iteration) and the number of features selected. Feature colours as in every importance plot. Code: `ML_v3_feature_engineering.m` (folder A).

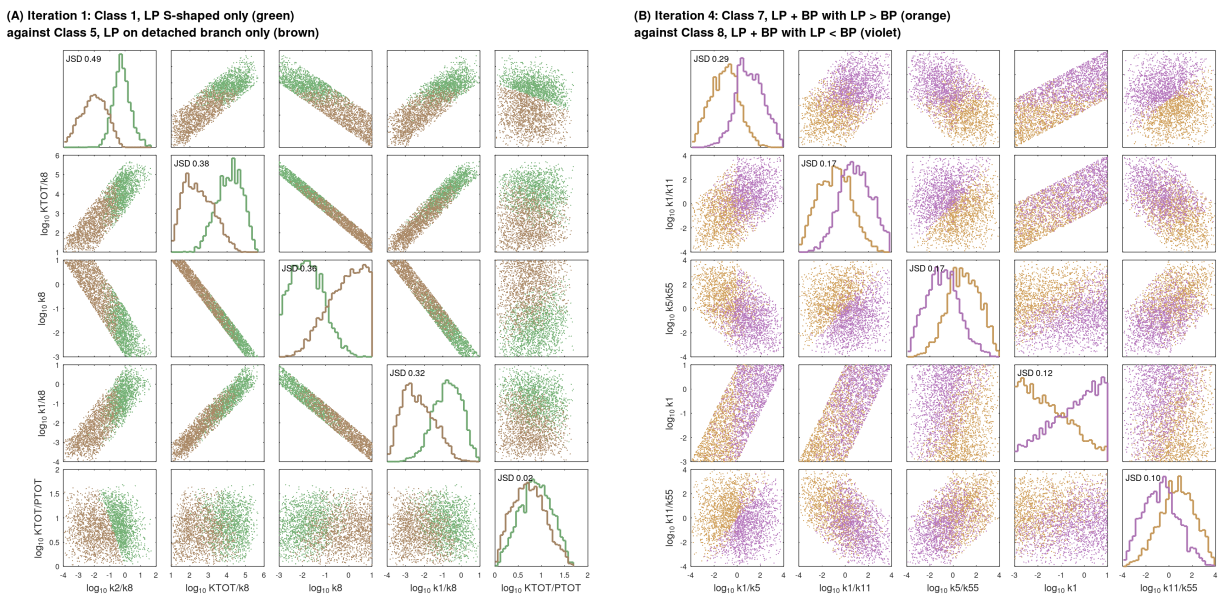
A score of one feature at a time misses boundaries in two features (Supplementary Fig. 9). The JSD scores each feature on its own. In Iteration 1, $K_{\mathrm{TOT}}/P_{\mathrm{TOT}}$ alone hardly separates Class 1 from Class 5 (JSD 0.02), yet in the plane of $\log_{10}(k_2/k_8)$ and $\log_{10}(K_{\mathrm{TOT}}/P_{\mathrm{TOT}})$ the two classes are divided by a straight line of slope -1, the boundary $k_8 P_{\mathrm{TOT}} = k_2 K_{\mathrm{TOT}}$ of criterion (10). Likewise, in Iteration 4 the two first-binding ratios k_1/k_5 and k_{11}/k_{55} overlap strongly one at a time, but together they place Classes 7 and 8 on either side of the boundary $K_{M5} = K_{M1}$. A filter that ranks single features can therefore discard a feature that is essential in combination, whereas a tree ensemble, which splits on several features in sequence, can use it. This is why set D, although it removes about 40% of the features, is not more accurate than set A, and why the main analyses use all 105 features.

What the six feature sets show (Supplementary Table 6; main-text Fig 9A and Supplementary Table 27). Pooled LightGBM accuracy lies between 88.8% and 92.7% for all six sets, so the information that determines the class is contained in the 14 parameters themselves; feature engineering changes how efficiently a tree ensemble can use it, not whether it is there. The gain is largest where the boundaries are oblique: within Iterations 1 and 2, set A is 4.8 and 4.6 points more accurate than set C, whereas within Iterations 3 and 4, where the boundary is the single comparison $k_4 > k_2$ or the first-binding asymmetry, every set reaches 96.8–100%. Removing the design-constrained ratios (set B) costs 1.2 points in the pooled data and nothing within iterations, where these ratios are constant, so the classifier's performance does not rest on recognising the sampling design. Divergence-based selection (set D) is as accurate as set A with about 40% fewer features in the pooled data and with

Supplementary Table 7. Features kept by JSD and the broken stick within each iteration. For each pair of classes that occur in the same iteration: the number of features kept and the three with the largest JSD (in brackets). Features selected in each iteration (union): Iteration 1, 20; Iteration 2, 47; Iteration 3, 2 (k_2/k_4, k_4/k_6); Iteration 4, 5 (k_1/k_{11}, k_1/k_5, k_2/k_4, k_4/k_6, k_5/k_{55}). Full lists: `JSD_selected_features.txt`. Code: `ML_v3_feature_engineering.m` (folder A).

Iteration	Classes	Kept	Largest JSD
1	1 vs 2	13	k_2/k_8 (0.485), k_8 (0.362), K_{TOT}/k_8 (0.361)
1	1 vs 5	14	k_2/k_8 (0.492), K_{TOT}/k_8 (0.377), k_8 (0.357)
1	1 vs 10	1	k_3/k_7 (0.121)
1	2 vs 5	4	k_3/k_6 (0.057), k_3/k_7 (0.047), P_{TOT}/k_3 (0.046)
1	2 vs 10	14	k_2/k_8 (0.495), K_{TOT}/k_8 (0.367), k_8 (0.360)
1	5 vs 10	16	k_2/k_8 (0.499), K_{TOT}/k_8 (0.382), k_8 (0.357)
2	3 vs 4	30	k_4/k_6 (0.368), K_{TOT}/k_6 (0.225), P_{TOT}/k_4 (0.210)
2	3 vs 5	6	k_4/k_6 (0.384), K_{TOT}/k_6 (0.285), P_{TOT}/k_4 (0.239)
2	3 vs 9	14	k_3/k_7 (0.339), k_2/k_7 (0.310), K_{TOT}/k_7 (0.278)
2	4 vs 5	16	k_3/k_7 (0.051), k_2/k_7 (0.044), K_{TOT}/k_7 (0.036)
2	4 vs 9	27	k_4/k_6 (0.204), K_{TOT}/k_6 (0.181), k_4/k_7 (0.156)
2	5 vs 9	29	K_{TOT}/k_6 (0.245), k_4/k_6 (0.220), k_6 (0.164)
3	6 vs 10	2	k_2/k_4 (0.685), k_4/k_6 (0.685)
4	7 vs 8	3	k_1/k_5 (0.294), k_1/k_{11} (0.175), k_5/k_{55} (0.170)
4	7 vs 10	2	k_2/k_4 (0.687), k_4/k_6 (0.687)
4	8 vs 10	2	k_2/k_4 (0.670), k_4/k_6 (0.670)

only 2 and 5 features in Iterations 3 and 4, so many of the 91 ratios are redundant; but, as shown above, in the pooled data it retains all seven design-constrained ratios and it ranks features one at a time. The centred log-ratio sets E and F perform like the raw parameters (88.8–89.8% pooled), so it is the explicit pairwise ratios, not a scale-free normalisation, that help. In summary, the engineered log-ratios matter less for accuracy than for interpretation: they let the importance of a tree ensemble fall on single, nameable ratios, and within each iteration these are the ratios of the analytical criteria (main-text Fig 10 and Table 5; Supplementary Note 17), which the models were never given.



Supplementary Fig. 9. Two-parameter views of two class pairs. Feature matrices of the four features with the largest JSD for the pair and the ratio that completes the analytical boundary. Diagonal: class-conditional distributions of each feature (30-bin histograms, normalised) with its JSD. Off-diagonal: the two features against each other (up to 1,500 random sets per class). (A) Iteration 1, Class 1 (green) against Class 5 (brown); the panel of $\log_{10}(k_2/k_8)$ against $\log_{10}(K_{\mathrm{TOT}}/P_{\mathrm{TOT}})$ (bottom left) shows the boundary $k_8 P_{\mathrm{TOT}} = k_2 K_{\mathrm{TOT}}$ of criterion (10), although $K_{\mathrm{TOT}}/P_{\mathrm{TOT}}$ alone has JSD 0.02. (B) Iteration 4, Class 7 (orange) against Class 8 (violet); the panel of $\log_{10}(k_1/k_5)$ against $\log_{10}(k_{11}/k_{55})$ shows the boundary $K_{M5} = K_{M1}$. Code: `ML_v3_feature_engineering.m` (folder A).

Supplementary Note 10. Model comparison, ROC curves and confusion matrices

Cross-validation. Every model was evaluated by five-fold stratified cross-validation inside the training rows and on the held-out test rows, for the pooled data and within each iteration (Supplementary Table 8). The cross-validated and held-out accuracies agree to within about two percentage points for every tree ensemble in every scope, so the held-out accuracies are not the result of a favourable split. LightGBM and XGBoost are the best models, or within one point of the best, in every scope; naive Bayes, which assumes independent features, is far behind wherever the features are strongly correlated, and in Iteration 2 it is below the majority baseline. With the features selected by Jensen–Shannon divergence (feature set D) the results are the same to within about one point (pooled LightGBM: held-out 92.61%, CV 92.18 ± 0.61%); within Iteration 2, set D has an about five times larger cross-validation spread (± 5.0 against ± 1.1 points for LightGBM), because the selected features change between folds.

ROC curves. The macro-averaged area under the one-vs-rest ROC curve is at least 0.992 for every tree ensemble on the pooled data, 0.962–0.979 in Iteration 1, 0.944–0.963 in Iteration 2 and 0.9995–1.000 in Iterations 3 and 4 (Supplementary Table 9). The smallest per-class areas belong to the classes that differ from Class 5 only by a fold on the main branch, Classes 2 and 4 (pooled LightGBM 0.988 and 0.984; within Iteration 2, Class 4: 0.920). The per-class precision, recall and

F1 (Supplementary Table 10) show the same: Classes 6, 7, 8 and 10 are recovered with F1 above 98%, Classes 2 and 4 with F1 of 65% and 63%.

Confusion matrices. Within each iteration the errors are confined to pairs of classes that differ by one singularity (Supplementary Fig. 11), and all tree ensembles make the same errors on the pooled data (Supplementary Fig. 12). The complete list of confusions of the pooled LightGBM classifier is given in Supplementary Note 14.

Supplementary Table 8. Model comparison: cross-validated and held-out accuracy. Feature set A. Each entry: five-fold cross-validated accuracy inside the training rows (mean ± SD, %) and, in brackets, the accuracy on the held-out test rows of the same split. Last row: majority-class baseline of the test rows. Code: `ML_v3_10class_parallel.m` (folder A).

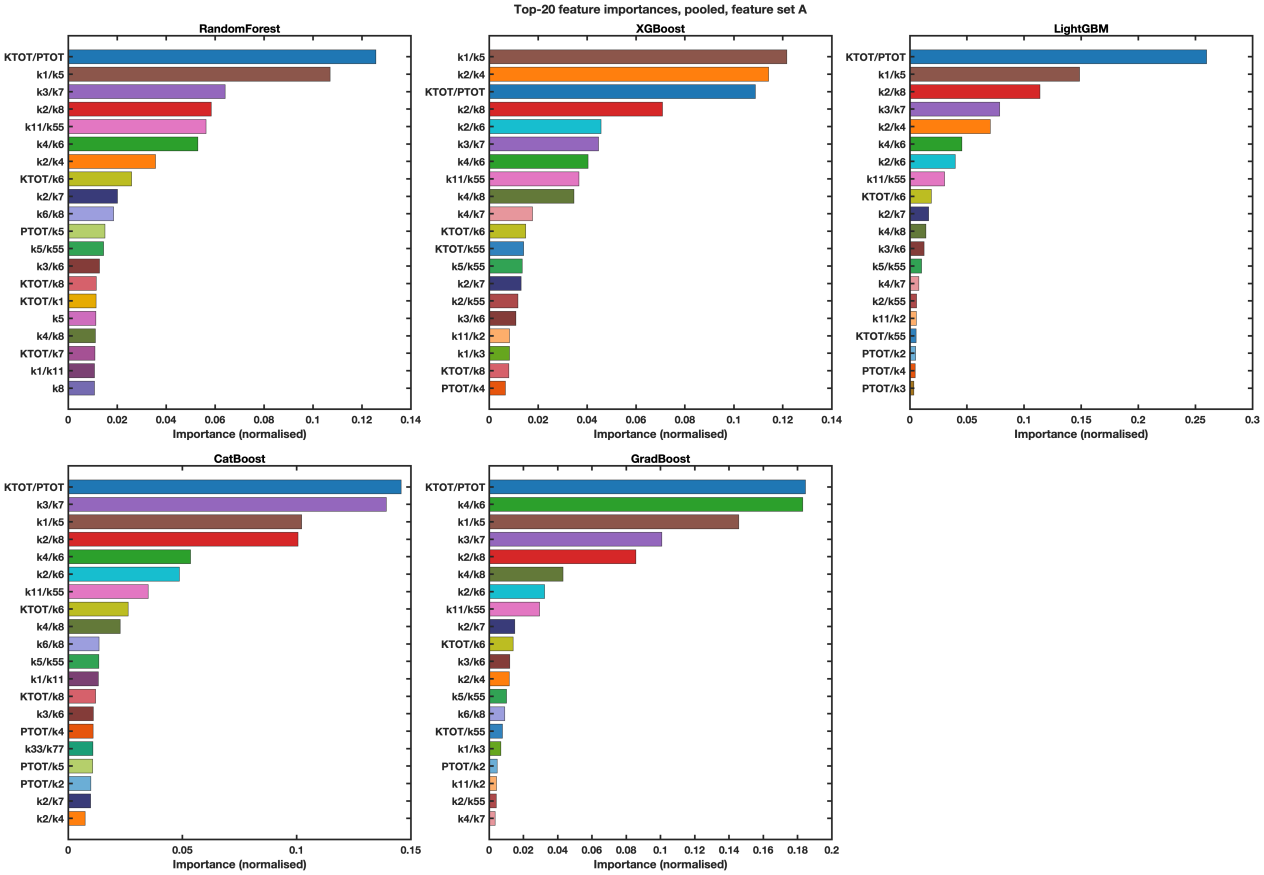
Model	Pooled	Iter. 1	Iter. 2	Iter. 3	Iter. 4
LightGBM	92.22 ± 0.36 (92.61)	85.46 ± 0.65 (87.60)	83.87 ± 1.06 (84.10)	99.97 ± 0.04 (100.00)	99.09 ± 0.31 (99.00)
XGBoost	91.48 ± 0.50 (91.87)	85.71 ± 0.38 (86.70)	84.23 ± 1.15 (83.20)	99.96 ± 0.06 (99.95)	99.06 ± 0.25 (99.15)
Grad. boosting	89.69 ± 0.42 (90.11)	84.60 ± 0.34 (85.70)	81.91 ± 1.38 (81.60)	99.96 ± 0.04 (99.95)	98.99 ± 0.13 (99.25)
Random forest	89.22 ± 0.29 (89.59)	80.50 ± 0.85 (81.70)	78.91 ± 0.59 (79.90)	99.97 ± 0.04 (99.95)	98.60 ± 0.42 (98.35)
CatBoost	86.07 ± 0.43 (86.69)	85.51 ± 0.60 (86.15)	82.00 ± 1.05 (80.10)	99.86 ± 0.07 (99.90)	98.97 ± 0.12 (99.20)
Naive Bayes	44.78 ± 0.31 (44.93)	63.90 ± 0.90 (65.60)	55.87 ± 0.26 (55.65)	98.78 ± 0.39 (98.80)	91.24 ± 0.40 (91.10)
Majority baseline	32.00	38.45	64.00	50.15	50.00

Supplementary Table 9. Area under the ROC curve of every model. Macro average of the one-vs-rest AUC on the held-out test rows, feature set A. In brackets: the smallest per-class AUC and its class. Code: `ML_v3_10class_parallel.m` (folder A).

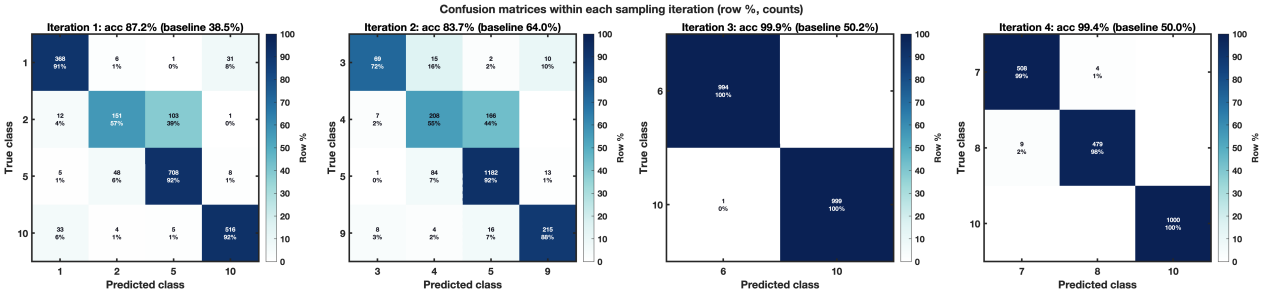
Model	Pooled	Iter. 1	Iter. 2	Iter. 3	Iter. 4
LightGBM	0.9955 (0.984, 4)	0.9775 (0.951, 2)	0.9615 (0.920, 4)	1.0000	0.9998
XGBoost	0.9952 (0.983, 4)	0.9776 (0.952, 2)	0.9598 (0.915, 4)	1.0000	0.9997
CatBoost	0.9943 (0.982, 4)	0.9792 (0.959, 2)	0.9626 (0.923, 4)	1.0000	0.9998
Gradient boosting	0.9937 (0.979, 4)	0.9740 (0.947, 2)	0.9499 (0.896, 4)	1.0000	0.9998
Random forest	0.9925 (0.979, 2, 4)	0.9622 (0.927, 2)	0.9442 (0.894, 4)	1.0000	0.9995
Naive Bayes	0.9562 (0.904, 5)	0.8981 (0.843, 2)	0.8435 (0.737, 4)	0.9997	0.9892

Supplementary Table 10. Per-class performance of the pooled LightGBM classifier (feature set A, first held-out test set, accuracy 92.61%, balanced accuracy 86.43%, macro-F1 87.14%). AUC: one-vs-rest area under the ROC curve. Code: `ML_v3_10class_parallel.m` (folder A).

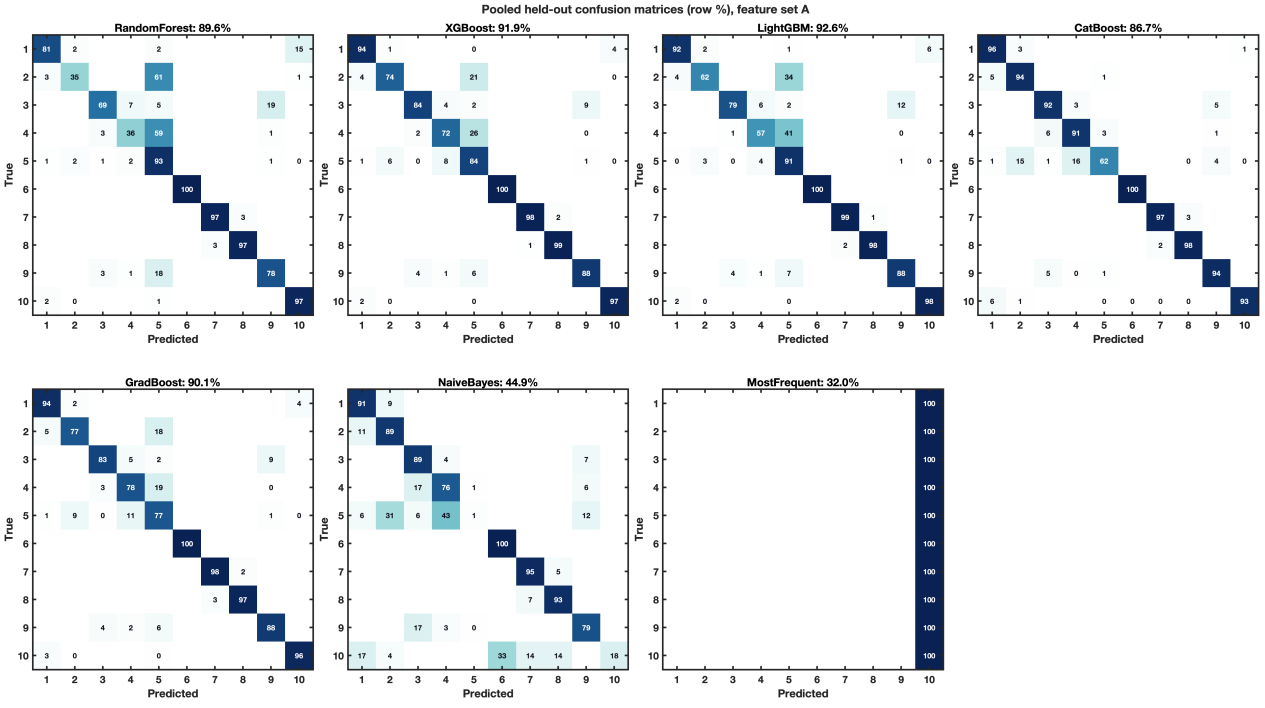
Class	Label	Test sets	Precision (%)	Recall (%)	F1 (%)	AUC
1	LP only, S-shaped	406	86.1	91.6	88.8	0.998
2	S-shaped LP + detached LP	267	68.5	61.8	65.0	0.988
3	LP only, Z-shaped	96	83.5	79.2	81.3	0.999
4	Z-shaped LP + detached LP	381	70.2	57.5	63.2	0.984
5	LP on detached branch only	2,049	87.0	91.4	89.1	0.988
6	BP only	994	100.0	100.0	100.0	1.000
7	LP + BP, $x_2^{LP} > x_2^{BP}$	512	98.2	98.6	98.4	1.000
8	LP + BP, $x_2^{LP} < x_2^{BP}$	488	98.6	98.2	98.4	1.000
9	no singularity, biphasic	243	89.5	88.1	88.8	0.999
10	no singularity, monotone	2,558	99.0	98.0	98.5	1.000



Supplementary Fig. 10. Pooled feature importance of all models. Normalised importance of the top features of each tree ensemble on the pooled data (feature set A). The design-constrained ratios $K_{\mathrm{TOT}}/P_{\mathrm{TOT}}$ and k_1/k_5 rank among the top three features of every model, together with the mechanistic ratios k_2/k_8, k_3/k_7 and k_4/k_6; because the constrained ratios identify the sampling iteration (Supplementary Note 7), the pooled ranking reflects the design and is not interpreted mechanistically (compare main-text Fig 10A). Each feature has one fixed colour, the same as in main-text Fig 10A and in the labels of Supplementary Figs 17 and 18. Code: `ML_v3_10class_parallel.m` (folder A).



Supplementary Fig. 11. Confusion matrices within each sampling iteration. LightGBM, feature set A, held-out test set of each iteration. Iteration 1: the errors are Class 2 predicted as Class 5 (39%) and Classes 1 and 10 exchanged (6–8%); Iteration 2: Class 4 predicted as Class 5 (44%) and Class 3 as Class 4 (16%); Iterations 3 and 4 are classified almost without error. Colour: row percentage (white to dark blue). Code: `ML_v3_10class_parallel.m` (folder A).



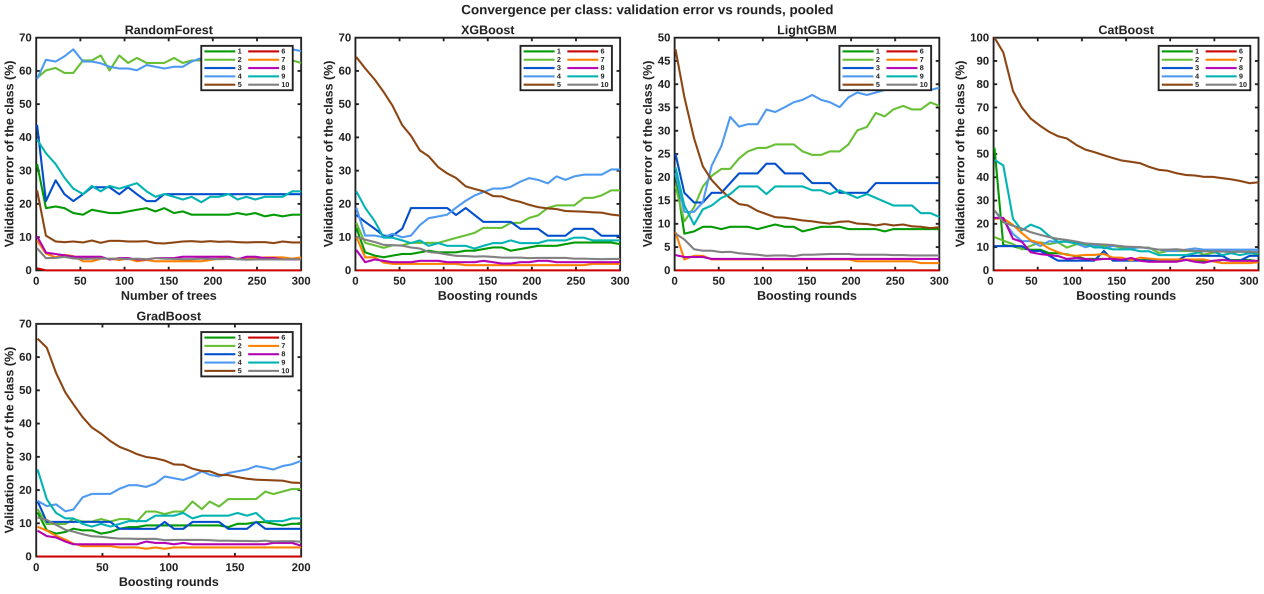
Supplementary Fig. 12. Pooled confusion matrices of all models. Feature set A, the same held-out test set for every model. All tree ensembles show the pattern of the LightGBM matrix: the errors concentrate in Classes 2 and 4 predicted as Class 5. Colour: row percentage (white to dark blue). Code: `ML_v3_10class_parallel.m` (folder A).

Supplementary Note 11. Convergence of the classifiers

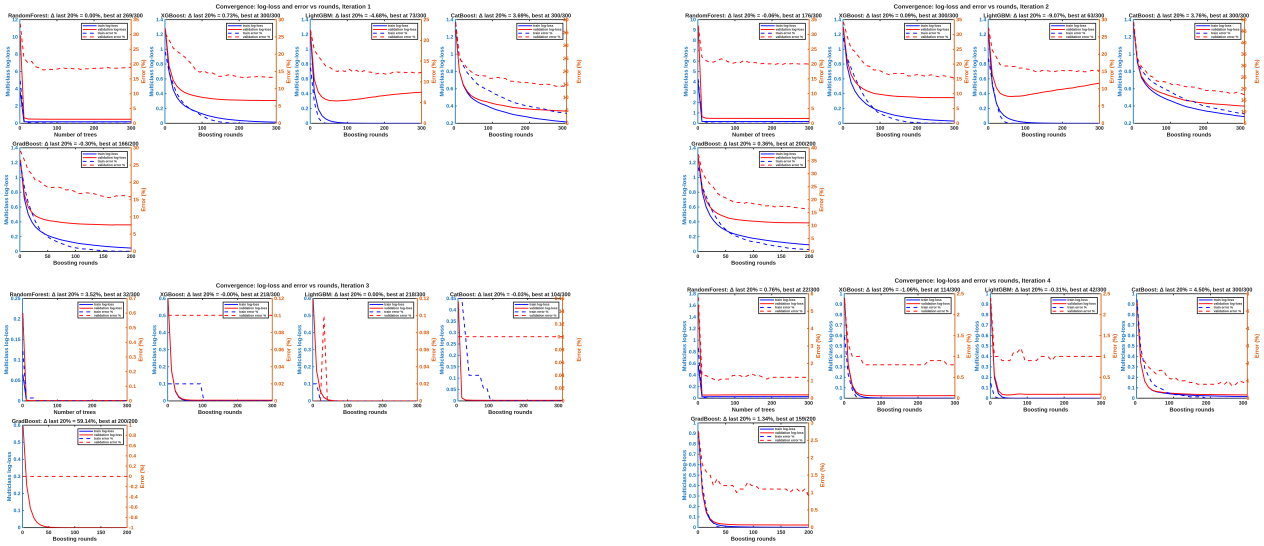
The multiclass log-loss and the error on the training rows (a random subsample of up to 5,000) and on the validation rows were recorded at 30 checkpoints along the boosting rounds, together with the validation error of every class (Supplementary Fig. 25B; Supplementary Figs 13 and 14; Supplementary Table 11). Three observations support the statement in the main text that 300 rounds suffice and that the remaining errors are not a matter of training length. (i) The random forest stabilises within a few tens of trees, and XGBoost and gradient boosting change their validation log-loss by at most 2.4% over the last 20% of rounds. (ii) LightGBM reaches its smallest validation log-loss early (round 63 of 300 on the pooled data) and then slowly over-fits the probabilities while its error stays constant; its held-out accuracy is the highest of all models. (iii) The classes that remain difficult are the same for every model and do not improve with more rounds: for the boosted models the validation error of Classes 2 and 4 increases slightly with the number of rounds while that of Class 5 decreases, i.e. the models shift the boundary between these classes rather than resolve it (Supplementary Fig. 13). CatBoost is the only model that has not converged at 300 rounds (5.8% change over the last 20%), which explains its lower accuracy and higher balanced accuracy. Within Iterations 3 and 4 every model converges to a per-class validation error of at most 2.5% (Supplementary Fig. 14).

Supplementary Table 11. Convergence of the boosted models and the random forest. Validation log-loss at the first and the last checkpoint, its relative change over the last 20% of rounds, the round with the smallest validation log-loss (of 300; 200 for gradient boosting), and the class with the largest validation error at the end. Feature set A. Code: `ML_v3_10class_parallel.m` (folder A).

Scope	Model	Log-loss, first	Log-loss, last	Change, last 20%	Best round	Largest class error
Pooled	LightGBM	1.670	0.264	−6.09%	63	Class 4, 39.3%
	XGBoost	1.963	0.188	1.98%	300	Class 4, 30.4%
	Gradient boosting	1.727	0.232	2.40%	200	Class 4, 28.8%
	Random forest	5.268	0.261	0.09%	300	Class 4, 66.0%
	CatBoost	1.885	0.302	5.78%	300	Class 5, 37.9%
Iter. 1	LightGBM	1.221	0.419	−4.68%	73	Class 2, 42.1%
	XGBoost	1.282	0.307	0.73%	300	Class 2, 31.6%
	Gradient boosting	1.238	0.359	−0.30%	166	Class 2, 38.3%
	Random forest	10.218	0.467	0.00%	269	Class 2, 66.2%
	CatBoost	1.282	0.340	3.69%	300	Class 5, 19.2%
Iter. 2	LightGBM	1.238	0.543	−9.07%	63	Class 4, 45.5%
	XGBoost	1.300	0.346	0.09%	300	Class 4, 34.6%
	Gradient boosting	1.266	0.386	0.36%	200	Class 4, 35.1%
	Random forest	9.396	0.471	−0.06%	176	Class 4, 67.5%
	CatBoost	1.313	0.399	3.76%	300	Class 5, 20.3%
Iter. 3	LightGBM	0.598	$< 10^{-4}$	0.00%	218	0.0%
	XGBoost	0.599	0.004	0.00%	218	Class 10, 0.2%
	Gradient boosting	0.598	$< 10^{-4}$	–	200	0.0%
	Random forest	0.209	0.001	3.52%	32	0.0%
	CatBoost	0.437	0.003	−0.03%	104	Class 10, 0.2%
Iter. 4	LightGBM	0.916	0.038	−0.31%	42	Class 7, 2.3%
	XGBoost	0.962	0.024	−1.06%	114	Class 7, 2.3%
	Gradient boosting	0.920	0.022	1.34%	159	Class 7, 2.0%
	Random forest	1.608	0.057	0.76%	22	Class 8, 2.5%
	CatBoost	0.938	0.031	4.50%	300	Class 8, 2.0%



Supplementary Fig. 13. Per-class convergence of the pooled classifiers. Validation error of each class as a function of the number of boosting rounds (trees for the random forest), feature set A. Classes 6, 7, 8 and 10 converge quickly to a small error; for the boosted models the error of Classes 2 and 4 increases with the number of rounds while that of Class 5 decreases. CatBoost has not converged at 300 rounds. Code: `ML_v3_10class_parallel.m` (folder A).



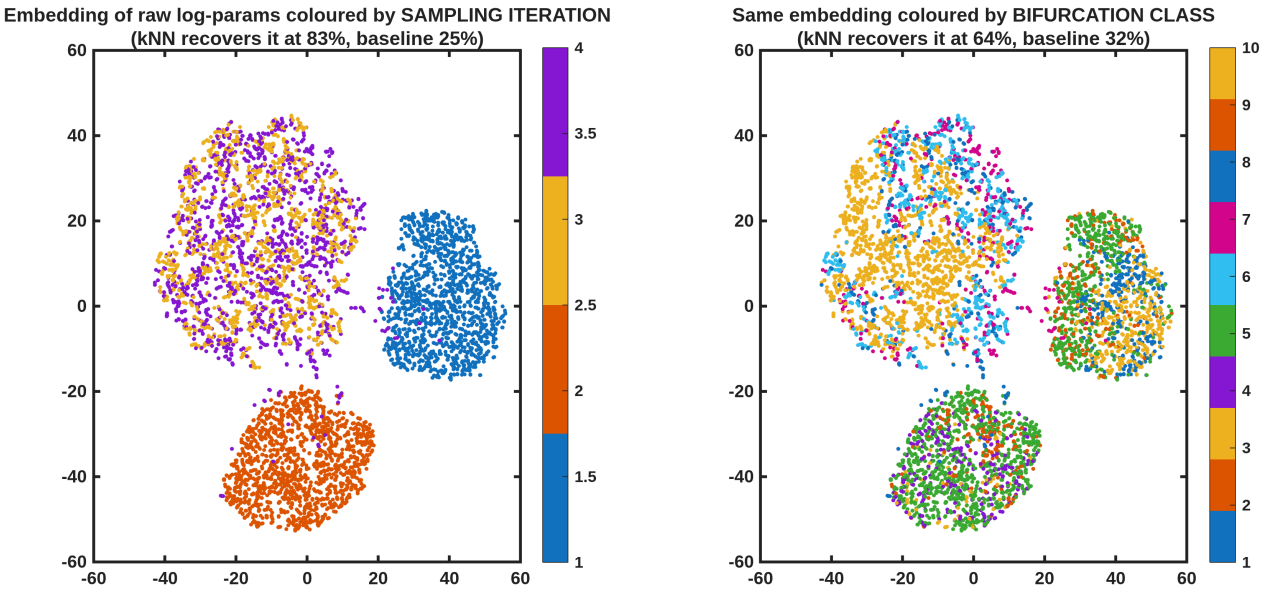
Supplementary Fig. 14. Convergence within each sampling iteration. Training and validation log-loss and error against the number of rounds in Iterations 1–4 (top left to bottom right; as in Supplementary Fig. 25B). The per-class validation errors of every iteration are among the machine-learning results (Supplementary File 3): in Iterations 3 and 4 every model converges to a per-class validation error of at most 2.5%; in Iterations 1 and 2 the largest remaining errors are those of Classes 2 and 4, as in the pooled data (Supplementary Fig. 13). Code: `ML_v3_10class_parallel.m` (folder A).

Supplementary Note 12. Embedding audit

Unsupervised embeddings of the parameters are often used to show that classes occupy distinct regions of parameter space. We tested whether such an embedding carries information about the class beyond the sampling design. A two-dimensional t-SNE embedding (perplexity 30) of 6,002 random parameter sets was computed from three representations of the parameters, and a k-nearest-neighbour classifier ($k = 15$, five-fold cross-validation) was asked to recover the iteration and the class from the embedding (Supplementary Table 12; Supplementary Fig. 15). The embedding recovers the iteration at 83–86% (baseline 25%) but the class at only 64–67% (baseline 32%), whereas a supervised classifier on the same 14 log-parameters reaches 89%. Within a regime of the design, the embedding recovers between −7% and 68% of the improvement over the baseline that the supervised classifier achieves; in the regime $K_{\mathrm{TOT}} < P_{\mathrm{TOT}}$ it is no better than the majority class. An embedding of the parameters therefore displays the sampling design rather than the class boundaries, and the main text uses mechanistic coordinates instead (main-text Fig 7).

Supplementary Table 12. Embedding audit. Accuracy (%) with which a k-nearest-neighbour classifier recovers the class (and, globally, the iteration) from a two-dimensional t-SNE embedding, compared with a supervised LightGBM classifier on the same representation. Lift recovered: (embedding − baseline)/(supervised − baseline). Representations: raw, the 14 log-parameters; CLR, centred log-ratio of the 14 log-parameters; CLR rates, centred log-ratio of the 12 rate constants and $\log(K_{\mathrm{TOT}}/P_{\mathrm{TOT}})$. Code: `ML_v3_10class_parallel.m` (folder A).

Scope	Representation	Sets	Baseline	Embedding	Supervised	Lift recovered
All (class)	raw	6,002	31.8	64.4	–	–
	CLR	6,002	31.8	65.3	–	–
	CLR rates	6,002	31.8	66.8	–	–
All (iteration)	raw	6,002	25.2	82.7	–	–
	CLR	6,002	25.2	83.4	–	–
	CLR rates	6,002	25.2	85.7	–	–
$K_{\mathrm{TOT}} > P_{\mathrm{TOT}}$	raw	3,000	37.5	62.3	77.0	63%
	CLR	3,000	37.5	62.1	76.0	64%
	CLR rates	3,000	37.5	63.6	76.2	68%
$K_{\mathrm{TOT}} < P_{\mathrm{TOT}}$	raw	3,000	64.3	64.1	77.5	−2%
	CLR	3,000	64.3	63.5	75.8	−7%
	CLR rates	3,000	64.3	63.3	77.7	−7%
$K_{\mathrm{TOT}} = P_{\mathrm{TOT}}$	raw	3,000	50.5	71.0	92.5	49%
	CLR	3,000	50.5	72.6	92.0	53%
	CLR rates	3,000	50.5	74.0	92.8	56%



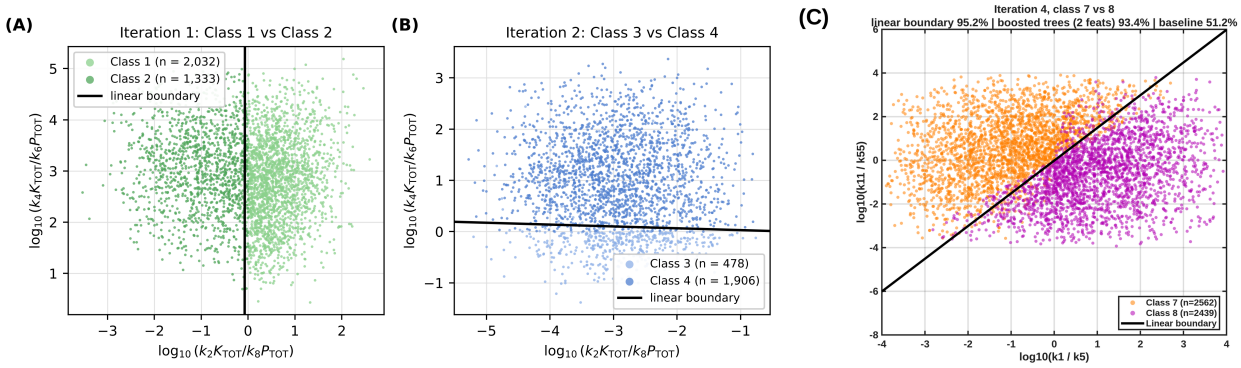
Supplementary Fig. 15. A neighbourhood-preserving embedding encodes the sampling design rather than the bifurcation class. Two-dimensional t-SNE embedding of the 14 log-parameters of 6,002 random parameter sets, coloured by sampling iteration (left) and by bifurcation class (right). A k-nearest-neighbour classifier ($k = 15$, five-fold cross-validation) recovers the iteration from the embedding at 82.7% (baseline 25.2%) but the class at only 64.4% (baseline 31.8%), whereas a supervised LightGBM classifier reaches 89.1% with the same 14 log-parameters and 92.7% with all 105 features (Supplementary Table 12). Code: `ML_v3_10class_parallel.m` (folder A).

Supplementary Note 13. Decisive two-class tests

Three pairs of classes differ by exactly one singularity whose existence is given in closed form: Classes 1 and 2 (Iteration 1) and Classes 3 and 4 (Iteration 2) by the fold on a detached branch, which exists if and only if $k_2 K_{\mathrm{TOT}} < k_8 P_{\mathrm{TOT}}$ and $k_4 K_{\mathrm{TOT}} > k_6 P_{\mathrm{TOT}}$ (main-text criterion (10), derived in Mathematica, Supplementary File 6, `Analytical_Derivations_StepByStep_final.nb`, Section 7); Classes 7 and 8 (Iteration 4) by the orientation of the crossing branch, set by the first-binding asymmetries ($K_{M5} < K_{M1}$ or $K_{M5} > K_{M1}$, main-text criterion (14), derived in Mathematica, `Analytical_Derivations_StepByStep_final.nb`, Section 8). Both criteria are evaluated on one ensemble parameter set of every class in `Excel_Parameter_Examples_Condition_Checks_final.nb` (Sections 3 and 4). Each pair was classified with all non-constant features, with the two mechanistic log-ratios only, and with a linear support-vector classifier on the two log-ratios (Supplementary Table 13 and Supplementary Fig. 16). For the detached-branch pairs, two log-ratios are as good as, or better than, all features, and a straight line is as good as boosted trees. In Iteration 1 every set of Classes 1 and 2 satisfies $k_4 K_{\mathrm{TOT}} > k_6 P_{\mathrm{TOT}}$ (the second log-ratio ranges from 0.43 to 5.2), so criterion (10) reduces to the sign of $\log_{10}(k_2 K_{\mathrm{TOT}}/k_8 P_{\mathrm{TOT}})$ and the boundary is a vertical line; in Iteration 2 every set of Classes 3 and 4 satisfies $k_2 K_{\mathrm{TOT}} < k_8 P_{\mathrm{TOT}}$, and the boundary is the horizontal line $\log_{10}(k_4 K_{\mathrm{TOT}}/k_6 P_{\mathrm{TOT}}) = 0$. The 2–5% of sets on the wrong side of the line are not errors of the criterion: they have a detached branch whose fold lies above the window $A_{\mathrm{TOT}} \leq 10^6$, so their label is that of the class without the detached fold. For Classes 7 and 8 the tree ensembles reach 93–95% with the two binding asymmetries and 98–99% with all features; this boundary is not linear in the two ratios and remains the one class boundary without a closed form (main text).

Supplementary Table 13. Decisive two-class tests. Accuracy (%, mean ± SD over three stratified splits) with all non-constant features and with the two mechanistic log-ratios only; linear SVM on the two log-ratios. Log-ratios: $\log_{10}(k_2 K_{\mathrm{TOT}}/k_8 P_{\mathrm{TOT}})$ and $\log_{10}(k_4 K_{\mathrm{TOT}}/k_6 P_{\mathrm{TOT}})$ for the first two pairs, $\log_{10}(k_1/k_5)$ and $\log_{10}(k_{11}/k_{55})$ for the third. Code: `ML_v3_10class_parallel.m` (folder A).

Test (baseline)	Model	All features	Two log-ratios
Iter. 1: Class 1 vs 2 (60.3)	LightGBM	97.37 ± 0.82	98.02 ± 0.82
	XGBoost	97.28 ± 0.70	98.27 ± 0.23
	CatBoost	96.83 ± 0.70	98.32 ± 0.31
	Random forest	95.74 ± 0.60	98.56 ± 0.09
	Naive Bayes	91.08 ± 0.65	96.43 ± 0.68
	Linear SVM	–	97.77 ± 0.26
Iter. 2: Class 3 vs 4 (79.9)	LightGBM	95.88 ± 0.12	95.18 ± 0.55
	XGBoost	95.60 ± 0.36	95.04 ± 0.85
	CatBoost	95.11 ± 0.61	94.97 ± 0.21
	Random forest	93.99 ± 0.64	96.37 ± 0.44
	Naive Bayes	85.60 ± 0.79	85.81 ± 0.32
	Linear SVM	–	95.11 ± 0.79
Iter. 4: Class 7 vs 8 (51.2)	LightGBM	98.37 ± 0.40	93.43 ± 0.15
	XGBoost	98.67 ± 0.25	93.63 ± 0.85
	CatBoost	98.70 ± 0.30	94.50 ± 0.85
	Random forest	98.03 ± 0.15	94.33 ± 0.58
	Naive Bayes	93.03 ± 1.21	92.40 ± 0.36
	Linear SVM	–	95.20 ± 0.44



Supplementary Fig. 16. Decisive two-class tests. (A) Iteration-1 parameter sets of Class 1 (no detached fold) and Class 2 (detached fold) against the two log-ratios of criterion (10); every set has $k_4 K_{\mathrm{TOT}} > k_6 P_{\mathrm{TOT}}$, and the boundary is the vertical line $\log_{10}(k_2 K_{\mathrm{TOT}}/k_8 P_{\mathrm{TOT}}) \approx 0$. (B) Iteration-2 parameter sets of Class 3 and Class 4; every set has $k_2 K_{\mathrm{TOT}} < k_8 P_{\mathrm{TOT}}$, and the boundary is the horizontal line $\log_{10}(k_4 K_{\mathrm{TOT}}/k_6 P_{\mathrm{TOT}}) \approx 0$. (C) Iteration-4 parameter sets of Class 7 (orange; $n = 2{,}562$) and Class 8 (violet; $n = 2{,}439$) against the first-binding asymmetries $\log_{10}(k_1/k_5)$ and $\log_{10}(k_{11}/k_{55})$; the linear boundary classifies 95.2% of test sets correctly (boosted trees on the same two features 93.4%; majority baseline 51.2%; LightGBM on all 59 features of Iteration 4, 98.4%). Black: linear boundary fitted to the sets shown (accuracies in Supplementary Table 13). Axes span the range of the data. Code: `ML_v3_10class_parallel.m` (folder A).

Supplementary Note 14. Misclassified parameter sets

Of the 7,994 test sets of the pooled LightGBM classifier, 591 are misclassified (7.4%). Supplementary Table 14 lists every confusion. The six largest (Supplementary Fig. 26) account for 463 of the 591 errors, and all six involve classes that differ only by the folds of the main branch: hysteresis on the main branch present or absent, with a detached fold in both classes (Classes 2 and 4 against Class 5) or in neither (Classes 1 and 10). Classes 7 and 8 are exchanged for 16 sets. No set of a branch-point class (6, 7, 8) is predicted as a class of another iteration, and no set with a singularity is predicted as monotone except Class 1 sets with a narrow loop (24) and two Class 5 sets. Every misclassified test set, with its parameters and singularities, is listed in `Misclassified_test.csv` (Supplementary File 3). A second kind of error, absent from Supplementary Fig. 26, comes from the labelling window: 42 errors are sets whose detached branch lies just beyond $A_{\mathrm{TOT}} = 10^6$ (Supplementary Note 19).

Supplementary Table 14. All confusions of the pooled LightGBM classifier (feature set A, first held-out test set, 591 misclassified of 7,994). Percentage: of the test sets of the true class. Code: `ML_v3_10class_parallel.m` (folder A).

True → predicted	Sets	of	%	True → predicted	Sets	of	%
4 → 5	157	381	41.2	7 → 8	7	512	1.4
2 → 5	91	267	34.1	1 → 2	7	406	1.7
5 → 4	85	2,049	4.1	3 → 4	6	96	6.2
5 → 2	66	2,049	3.2	4 → 3	4	381	1.0
10 → 1	40	2,558	1.6	1 → 5	3	406	0.7
1 → 10	24	406	5.9	10 → 2	3	2,558	0.1
9 → 5	18	243	7.4	3 → 5	2	96	2.1
3 → 9	12	96	12.5	9 → 4	2	243	0.8
5 → 9	12	2,049	0.6	5 → 3	2	2,049	0.1
2 → 1	11	267	4.1	5 → 10	2	2,049	0.1
9 → 3	9	243	3.7	4 → 9	1	381	0.3
8 → 7	9	488	1.8				
5 → 1	9	2,049	0.4				
10 → 5	9	2,558	0.4				

Supplementary Note 15. Computational cost

Supplementary Table 15 gives the cost of the labelling pass of each iteration on a Linux server with an AMD EPYC 9334 processor (32 cores), 251 GB RAM and MATLAB R2026a, with 64 workers. Core-hours are the sum of the per-set times, i.e. the serial CPU time; wall-clock times include the start-up of the parallel pool and the export of the results. The exact labelling of all 40,000 parameter sets, including the normal-form coefficients, takes 0.13 core-hours (11 ms per set); the MATCONT validation of 4,000 sets seeded on every branch takes 8.9 core-hours (8.0 s per set), about 700 times more per set. Almost all of the CPU time of the four main loops (48 min wall clock) is spent in the validation runs. Single-start continuation of all 40,000 sets (`SingleStart_v3.m`, same machine, 64 workers) takes 4.9 core-hours (0.44 s per set on average, median 0.30 s) and 4.8 min wall clock: about 40 times the cost per set of the exact labelling and about one eighteenth of that of MATCONT seeded on every branch. It is cheaper than the seeded runs because it follows only the branches reachable from one starting point and stops after a fixed number of continuation points, which is also why it misses detached branches and singularities at large A_{TOT} (Supplementary Note 5).

Supplementary Table 15. Computational cost of the labelling. Core-hours: sum of the per-set times. Times per set: mean (median). Main loop and whole script refer to `Ensemble_LHS_v3.m` (exact labelling and validation).

	Iter. 1	Iter. 2	Iter. 3	Iter. 4	Total
Exact labelling, core-h (10,000 sets each)	0.040	0.034	0.027	0.026	0.126
per set, ms	14.3 (11.0)	12.2 (11.5)	9.7 (9.6)	9.2 (9.9)	11.4
of which labels and normal forms, ms	1.61	0.32	0.46	0.44	0.71
MATCONT validation, core-h (1,000 sets each)	2.27	2.13	2.30	2.16	8.87
per set, s	8.2 (7.2)	7.7 (7.5)	8.3 (5.8)	7.8 (5.5)	8.0
Single-start MATCONT, core-h (10,000 sets each)	0.89	0.84	1.61	1.59	4.93
per set, s (mean)	0.32	0.30	0.58	0.57	0.44
wall clock, s (64 workers)	52	49	95	94	290
Main loop, wall clock, min (64 workers)	11.3	18.4	8.1	9.8	47.6
Whole script, wall clock, min	13.2	20.0	9.6	11.4	54.2

Supplementary Note 16. Supplementary files and code

The supporting package contains the files listed in Supplementary Table 16; Supplementary Table 17 gives the code that produced each figure and table of the main text and of this Supplementary Information. The order in which the codes are run, their inputs, outputs, settings and run times, and the required software are described in `README.txt` (Supplementary File 5). The code is publicly available on GitHub (`https://github.com/HarikaGL/High_throughput_bifurcation`), and the code together with all data and results is archived in Zenodo (doi:10.5281/zenodo.23609482). Parameters are in nM (totals) and $\mathrm{nM}^{-1}\,\mathrm{s}^{-1}$ or s^{-1} (rate constants), as in main-text Table 1.

Supplementary Table 16. Supplementary Files.

File	Content
Supplementary File 1 (CRNToolbox)	Basic Report (`CRNT_basic_report_MAPKFULL.txt`) and Higher Deficiency Report (`CRNT_higher_deficiency_report_MAPKFULL.txt`) of the Chemical Reaction Network Toolbox, version 2.35, and the network input file `MAPKFULL.NET` (Supplementary Note 1).
Supplementary File 2 (ensemble labelling)	MATLAB code of the exact labelling pipeline (`Ensemble_LHS_v3.m`, helper functions and the model file `MAPKfull.m`), the figure scripts, `Manuscript_numbers_v3.m` and the results of all four iterations (folder A). The labelling code must be run with the folder placed inside the MATCONT 7p6 folder.
Supplementary File 3 (machine learning)	`ML_v3_10class_parallel.m` (set `cfg.nWorkers = 0` to run it serially), `ML_v3_misclassified_diagrams.m`, `ML_v3_misclassified_figure17.m` and all machine-learning results, including the list of misclassified test sets; `ML_v3_SHAP_per_class.m` (per-class TreeSHAP contributions, Supplementary Note 17); the folder `Class_transitions` (ready reckoner of main-text Fig 7 and the transition analysis of Supplementary Note 18 and the sets beyond the window of Supplementary Note 19, run with `TR_00_run_all.m`) (folder A); and `SingleStart_v3.m`, which runs single-start continuation on the same parameter sets (outputs in `Ensemble_iter<i>_v3_1e6/SingleStart/`).
Supplementary File 4 (single-start comparison)	`make_comparison_figures.m` (an independent implementation of the exact labelling, the comparison of exact and single-start labels, and the verification tests of the main-text Methods) and its results (folder B).
Supplementary File 5 (`README.txt`)	Organisation of the supporting package, the order in which the codes are run, their inputs, outputs, settings and run times, and the required software.
Supplementary File 6 (Mathematica notebooks)	Wolfram Mathematica 15 notebooks of the analytical results (folder C, `C_Mathematica_derivations`): `Analytical_Derivations_StepByStep_final.nb` (species, complexes and reactions, Section 1; conservation laws and deficiency, main-text Eqs (3)–(6), Section 2; reduced model and analytic Jacobian, Eqs (1), (2), Section 3; compact form of the Jacobian, Section 4; steady-state parametrisation, Eqs (7)–(9), Section 5; symbolic checks of the flux and total balances, Section 6; detached-branch and branch-point conditions, Eqs (10), (11), Section 7; slope of the crossing branch and the Class 7/8 criterion, Eqs (13), (14), Section 8; normal-form coefficients, Section 9) and `Excel_Parameter_Examples_Condition_Checks_final.nb` (the conditions evaluated on one ensemble parameter set of every class, with the sampling iteration and row of each set in the ensemble files). Each section of both notebooks ends with a cell that plots its result numerically (for example the reaction network, the parametrisation and complete diagram of the Fig. 3 set, the dose-up and dose-down sweeps, the normal forms, and the exact diagram of every example set); `Analytical_Derivations_StepByStep_final.pdf` and `Excel_Parameter_Examples_Condition_Checks_final.pdf` are the evaluated notebooks with all outputs and figures.

Supplementary Table 17. Code used for each figure and table. Folder A: `A_Ensemble_labelling_and_ML` (Supplementary Files 2 and 3); folder B: `B_Single_start_comparison` (Supplementary File 4); folder C: `C_Mathematica_derivations` (Supplementary File 6). Supplementary figures and tables are prefixed "S".

Figure / table	Code	Folder
Fig 1	`Ensemble_LHS_tiles_v3.m` ([II]); `Ensemble_class_table_v3.m` ([III]); `Ensemble_class_wheel_v3.m` ([IV])	A
Fig 2	`Ensemble_iteration_panels_v3.m`	A
Fig 3	`Fig_inverse_method_v3.m` (equations of `support/ss_structure.m`)	A
Fig 4	`Fig_criterion_v3.m`	A
Figs 5, 6, 10B; S Figs 3–5	`make_comparison_figures.m` (single-start labels from `SingleStart_v3.m`, folder A)	B
Fig 7	`Class_transitions/TR_01_ready_reckoner.m` (open circles: `tr_lib/tr_window_sets.m`)	A
Fig 8	`ML_v3_10class_parallel.m`; block 3: `ML_v3_feature_engineering.m`	A
Figs 9, 10A; S Figs 10–16, 25	`ML_v3_10class_parallel.m` (feature colours: `support/feature_colors.m`; colour scale: `support/cm_graded.m`)	A
S Fig 24	`Fig_normal_forms_v3.m`	A
S Fig 26	`ML_v3_misclassified_figure17.m` (panels selected as in `ML_v3_misclassified_diagrams.m`)	A
S Figs 17, 18; S Table 18	`ML_v3_SHAP_per_class.m`	A
S Figs 19–21; S Tables 19, 20	`Class_transitions/TR_02_scan.m`, `TR_03_two_class_sweeps.m`, `TR_04_figures.m` (all run by `TR_00_run_all.m`)	A
S Figs 22, 23; S Table 21	`Class_transitions/TR_05_window_sets.m`	A
S Figs 1, 2	`Ensemble_LHS_v3.m` (MATCONT check and validation)	A
Tables 1, 2; S Table 22	`Ensemble_LHS_v3.m`, `Manuscript_numbers_v3.m`; first row of S Table 22 `make_comparison_figures.m`	A, B
Table 4; S Tables 26, 27	`ML_v3_10class_parallel.m`	A
Table 5	`ML_v3_SHAP_per_class.m`	A
S Table 25	`Ensemble_LHS_v3.m` (`support/normal_forms.m`), `Manuscript_numbers_v3.m`	A
S Tables 24, 28	`SingleStart_v3.m` (single start), `Ensemble_LHS_v3.m` (validation)	A
Eqs (1)–(14)	`Analytical_Derivations_StepByStep_final.nb` (Wolfram Mathematica 15)	C
S Table 1	CRNToolbox reports (Supplementary File 1)	–
S Tables 2–4, 15	`Ensemble_LHS_v3.m`: `Bifurcation_Analysis_Results_v3.xlsx`, `Core_Hours_iter<i>.txt`; `SingleStart_v3.m`: `SingleStart_summary.txt`	A
Table 3; S Figs 6–9; S Tables 6, 7	`ML_v3_feature_engineering.m` (accuracies of S Table 6: `ML_v3_10class_parallel.m`)	A
Classification time per set (Results, Methods)	`ML_v3_inference_time.m`	A
S Table 5, S Tables 8–14	`ML_v3_10class_parallel.m`: `ML_v3_summary.txt` and the tables in `labels_Exact_Class_v3/`	A

Supplementary Note 17. Per-class feature contributions (TreeSHAP)

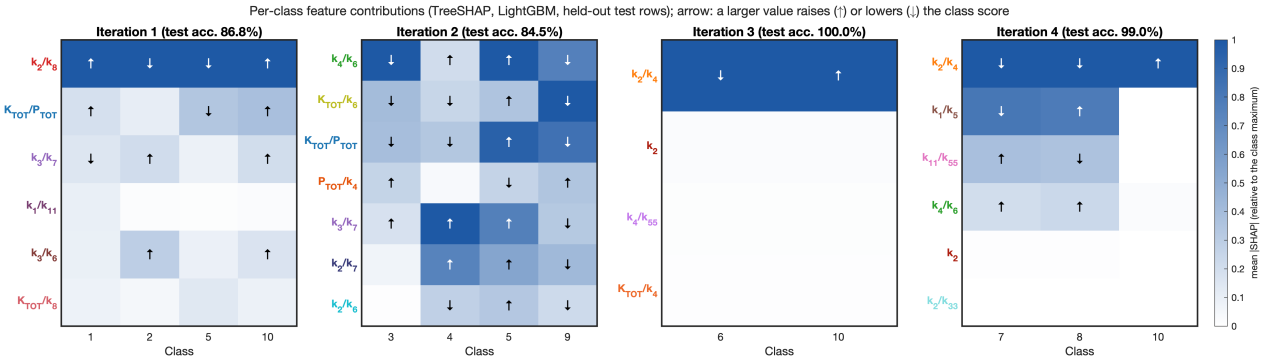
The importances of main-text Fig 10A are LightGBM split gains summed over all trees of the multiclass model; they rank features for the classifier as a whole but cannot say which class a feature informs. In multiclass LightGBM every boosting round builds one tree per class, and TreeSHAP (Lundberg *et al.*, *Nat. Mach. Intell.* 2020), which is exact for tree models and built into LightGBM (`pred_contrib`), decomposes the raw score of every class for every parameter set into additive contributions of the features, with a sign.

Method (`ML_v3_SHAP_per_class.m`, folder A). The data, features, model and splits are those of the within-iteration analysis of the main pipeline (Supplementary Note 8): rows with `Use_For_ML` = 1, feature set A (14 log-parameters and 91 log-ratios, constant and duplicated columns removed on the training rows), LightGBM with 300 rounds, 63 leaves, learning rate 0.1 and inverse-frequency weights, and the stratified 70/10/20 % splits with seeds $42 + 100a + r$. Within each iteration the model was refitted on the training rows of three splits, and the contributions were computed on the held-out test rows. For class k and feature j we report (i) the mean of $|\phi_{kj}|$ over all test rows, which measures how much the feature moves the score of class k anywhere in the sampled region; (ii) the same mean over the test rows of class k only (output file `SHAP_iter<i>_meanabs_own.csv`); and (iii) the direction, the sign of the correlation between the value of the feature and ϕ_{kj}. The analysis is done within each iteration, never on the pooled data, because pooled features carry the sampling design (Supplementary Note 7). Correlated log-ratios share credit; because SHAP values are additive, the contribution of each log-ratio $\log(a/b)$ was also split equally between a and b to give a per-parameter attribution (Supplementary Fig. 18).

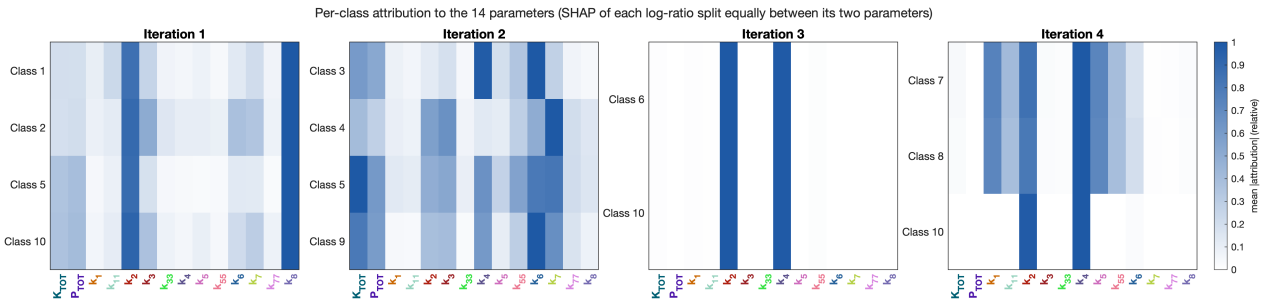
Results. Test accuracies were 86.8, 84.5, 99.98 and 99.0 % in Iterations 1–4. Supplementary Table 18 lists the five largest contributions of every class and Supplementary Fig. 17 shows them as heatmaps. The leading features and their directions agree with the exact conditions: k_2/k_8 and $K_{\mathrm{TOT}}/P_{\mathrm{TOT}}$ raise Classes 1 and 10 and lower Classes 2 and 5 (site-1 condition, $x = \log_{10}(k_2 K_{\mathrm{TOT}}/k_8 P_{\mathrm{TOT}}) < 0$ for a detached branch); k_4/k_6 and K_{TOT}/k_6 lower Classes 3 and 9 and raise Class 5 (site-2 condition); k_2/k_4 separates Classes 6–8 from Class 10 (branch point iff $k_4 > k_2$); k_1/k_5 and k_{11}/k_{55} separate Classes 7 and 8 with opposite signs ($K_{M5} < K_{M1}$). The features outside these conditions, k_3/k_7, k_3/k_6 and k_2/k_7, appear for the classes that differ from a neighbour only by the fold on the main branch (Classes 1, 2, 4, 5 and 10); they lead for Class 4. In the per-parameter view the fold classes load on k_3 and k_7, and Class 4 on k_7 more than on any other parameter. These features are the candidates for the open boundary tested in Supplementary Note 18.

Supplementary Table 18. Five largest per-class contributions in each sampling iteration. Feature (direction; mean |SHAP| over the held-out test rows, mean of three splits; share of the class total). ↑: a larger value raises the class score; ↓: lowers it. Source: `ML_v3_results_SHAP/SHAP_top_features_per_class.csv`.

Iter.	Class	Feature (direction; mean |SHAP|; share)
1	1	k_2/k_8 ↑ (4.94; 37%), $K_{\mathrm{TOT}}/P_{\mathrm{TOT}}$ ↑ (0.91; 7%), k_3/k_7 ↓ (0.75; 6%), k_1/k_{11} ↓ (0.47; 3%), k_3/k_6 ↓ (0.46; 3%)
	2	k_2/k_8 ↓ (2.87; 26%), k_3/k_6 ↑ (0.83; 8%), k_3/k_7 ↑ (0.60; 5%), $K_{\mathrm{TOT}}/P_{\mathrm{TOT}}$ ↓ (0.31; 3%), k_2/k_7 ↑ (0.26; 2%)
	5	k_2/k_8 ↓ (5.73; 38%), $K_{\mathrm{TOT}}/P_{\mathrm{TOT}}$ ↓ (1.95; 13%), K_{TOT}/k_8 ↓ (0.80; 5%), k_3/k_6 ↓ (0.50; 3%), k_3/k_7 ↓ (0.47; 3%)
	10	k_2/k_8 ↑ (4.98; 29%), $K_{\mathrm{TOT}}/P_{\mathrm{TOT}}$ ↑ (1.89; 11%), k_3/k_7 ↑ (1.02; 6%), k_3/k_6 ↑ (0.82; 5%), P_{TOT}/k_2 ↓ (0.76; 4%)
2	3	k_4/k_6 ↓ (1.91; 24%), K_{TOT}/k_6 ↓ (0.78; 10%), $K_{\mathrm{TOT}}/P_{\mathrm{TOT}}$ ↓ (0.75; 9%), P_{TOT}/k_4 ↑ (0.50; 6%), k_3/k_7 ↑ (0.34; 4%)
	4	k_3/k_7 ↑ (1.40; 14%), k_2/k_7 ↑ (1.04; 10%), $K_{\mathrm{TOT}}/P_{\mathrm{TOT}}$ ↓ (0.48; 5%), k_2/k_6 ↓ (0.39; 4%), k_3/k_6 ↓ (0.39; 4%)
	5	k_4/k_6 ↑ (0.91; 9%), $K_{\mathrm{TOT}}/P_{\mathrm{TOT}}$ ↑ (0.87; 9%), k_3/k_7 ↑ (0.70; 7%), k_2/k_7 ↑ (0.49; 5%), K_{TOT}/k_{55} ↑ (0.33; 3%)
	9	K_{TOT}/k_6 ↓ (1.98; 13%), $K_{\mathrm{TOT}}/P_{\mathrm{TOT}}$ ↓ (1.68; 11%), k_4/k_6 ↓ (1.50; 10%), k_2/k_7 ↓ (0.83; 6%), k_4/k_7 ↓ (0.76; 5%)
3	6	k_2/k_4 ↓ (15.07; 92%), k_2 ↓ (0.33; 2%), k_4/k_{55} ↑ (0.21; 1%), K_{TOT}/k_4 ↓ (0.15; 1%), k_4 ↑ (0.10; 1%)
	10	k_2/k_4 ↑ (15.07; 92%), k_2 ↑ (0.33; 2%), k_4/k_{55} ↓ (0.21; 1%), K_{TOT}/k_4 ↑ (0.15; 1%), k_4 ↓ (0.10; 1%)
4	7	k_2/k_4 ↓ (3.48; 30%), k_1/k_5 ↓ (2.84; 25%), k_{11}/k_{55} ↑ (1.32; 12%), k_4/k_6 ↑ (0.72; 6%), k_{11}/k_2 ↑ (0.63; 6%)
	8	k_2/k_4 ↓ (3.43; 30%), k_1/k_5 ↑ (2.71; 24%), k_{11}/k_{55} ↓ (1.25; 11%), k_4/k_6 ↑ (0.80; 7%), k_5/k_{55} ↓ (0.40; 3%)
	10	k_2/k_4 ↑ (8.19; 97%), k_4/k_6 ↓ (0.16; 2%), k_2 ↑ (0.01; 0%), k_2/k_{33} ↑ (0.01; 0%), k_2/k_3 ↑ (0.01; 0%)



Supplementary Fig. 17. Per-class feature contributions within each sampling iteration. Columns: classes; rows: the union of the four features with the largest mean |SHAP| of every class (LightGBM, feature set A, held-out test rows, mean of three splits). Colour: mean |SHAP| relative to the largest value of the class; arrow: a larger value of the feature raises (↑) or lowers (↓) the class score. Feature names are printed in the colour of the feature in every bar chart (main-text Fig 10A, Supplementary Fig. 10). Code: `ML_v3_SHAP_per_class.m` (folder A).



Supplementary Fig. 18. Per-class attribution to the 14 parameters. The SHAP value of every log-ratio $\log(a/b)$ is split equally between a and b and added to the SHAP value of the log-parameters themselves; colour: mean absolute attribution of each parameter to the class score, relative to the largest value of the class. Parameter names in the colour of the parameter in the bar charts. Code: `ML_v3_SHAP_per_class.m` (folder A).

Supplementary Note 18. Transitions between classes

The code is in `A_Ensemble_labelling_and_ML/Class_transitions` (`TR_00_run_all.m`). Every parameter set is labelled with the labeller of `Ensemble_LHS_v3.m` (`ss_structure` and `exact_labels`, $A_{\mathrm{TOT}} \leq 10^6$, 10,000 grid points per branch). The exact conditions used for the moves, $x < 0$ and $y > 0$ for a detached branch (criterion (10)), $u^* = r$ for a branch point (Eq. (11)) and the sign of $K_{M5} - K_{M1}$ for Classes 7 and 8 (Eq. (14)), were derived in Mathematica (Supplementary File 6, `Analytical_Derivations_StepByStep_final.nb`, Sections 7 and 8).

Sets. In each iteration, 120 sets of every class (sets used for machine learning) were drawn with a portable random generator, giving 1,560 source sets (13 class–iteration combinations).

Single-parameter scan (`TR_02_scan.m`). Each of the 14 parameters, and in Iterations 3–4 the symmetry-preserving pairs (k_4, k_8) and (k_1, k_{11}), was multiplied by 10^d, $d \in \{-1, -0.5, 0.5, 1\}$ (92,160 labels). For every ordered pair of classes, Supplementary Table 19 gives the move that converts the largest fraction of the source sets (`TR_scan_summary.txt` lists all of them, with the classes reached by the remaining sets).

Exact-condition moves. $x = -0.3$ through k_8: Class 1 → 2 (43.3 %), 5 (54.2 %), 1 (2.5 %); Class 10 → 5 (70.8 %), 2 (19.2 %), 10 (10.0 %). $y = +0.3$ through k_4: Class 3 → 4 (67.5 %), 5 (20.0 %), 3 (12.5 %); Class 9 → 5 (45.8 %), 9 (54.2 %); through k_6, Class 3 → 4 (44.2 %), 5 (45.0 %). Of the 720 sets moved across $x = 0$ or $y = 0$, 77.5 % gained a detached branch inside the window; all 162 others had the detached fold above $A_{\mathrm{TOT}} = 10^6$. $k_4 = k_8 = 2k_2$: Class 10 of Iteration 3 → 6 (100 %); of Iteration 4 → 7 (53.3 %) or 8 (46.7 %). $\log_{10}(K_{M1}/K_{M5}) = +0.3$ or -0.3 through k_5: Class 6 → 7 or 8 (100 %).

Symmetry breaking. Every single-parameter move of a Class 6–8 set that changes u^* relative to r (all parameters except k_1, k_{11}, k_5, k_{55}; 14,400 moves) removed the branch point; the new set had a detached branch if and only if $x < 0$ and $y > 0$ in 99.2 % of the moves, and all 110 exceptions had their detached fold above the window. Moves of k_1, k_{11}, k_5 or k_{55} (which do not enter u^*; 5,760 moves) kept the branch point in all cases. A change of k_3 or k_7 gave Class 5 in 98 % of the moves; a change of K_{TOT} or P_{TOT} gave Class 5 (64 %), 9 or 10 (18 % each) (Supplementary Fig. 21).

Two-class sweeps (`TR_03_two_class_sweeps.m`). For each edge, one quantity was swept through the boundary at 41 points in 80 sets of the source class, all other parameters fixed (Supplementary Table 20, Supplementary Fig. 20). Sweeps of x through k_8 and of y through k_4 are two-class in 21–61 % of sets; along the others the main-branch fold appears or disappears at the same time (for example Class 1 → 5 instead of 2). Moving x through k_2, K_{TOT} or P_{TOT}, or y through k_6, K_{TOT}

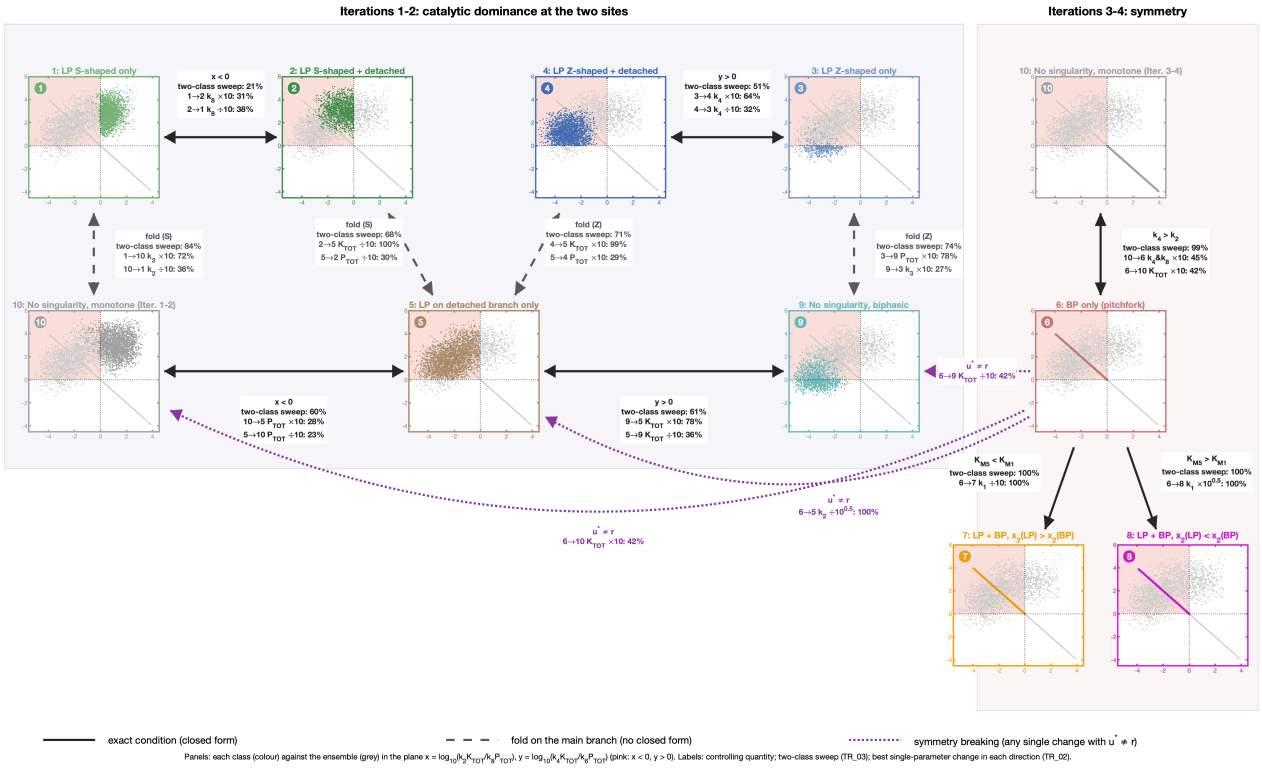
or P_{TOT}, never gave a two-class sweep. Sweeps of k_3 (or k_7), which change $\log_{10}(k_3/k_7)$ without changing x or y, are two-class in 68–84 % of sets for the four fold edges; their third classes are detached folds that leave or enter the window. The branch-point sweep $\log_{10}(k_4/k_2)$ with $k_4 = k_8$ is two-class in 98.8 % of sets, and the $\log_{10}(K_{M1}/K_{M5})$ sweep in all sets, with Class 6 only at $K_{M1} = K_{M5}$ exactly.

Supplementary Table 19. Edges of the transition map. Kind: exact (closed-form condition), fold (main-branch fold, no closed form) or symmetry (any change with $u^* \neq r$). Two-class: percentage of the 80 swept sets in which only the two classes occur along the sweep (Supplementary Table 20). Best move: the single-parameter change of the scan that converts the largest fraction of the 120 source sets, from the first to the second class and back. Source: `Class_transitions/results/`.

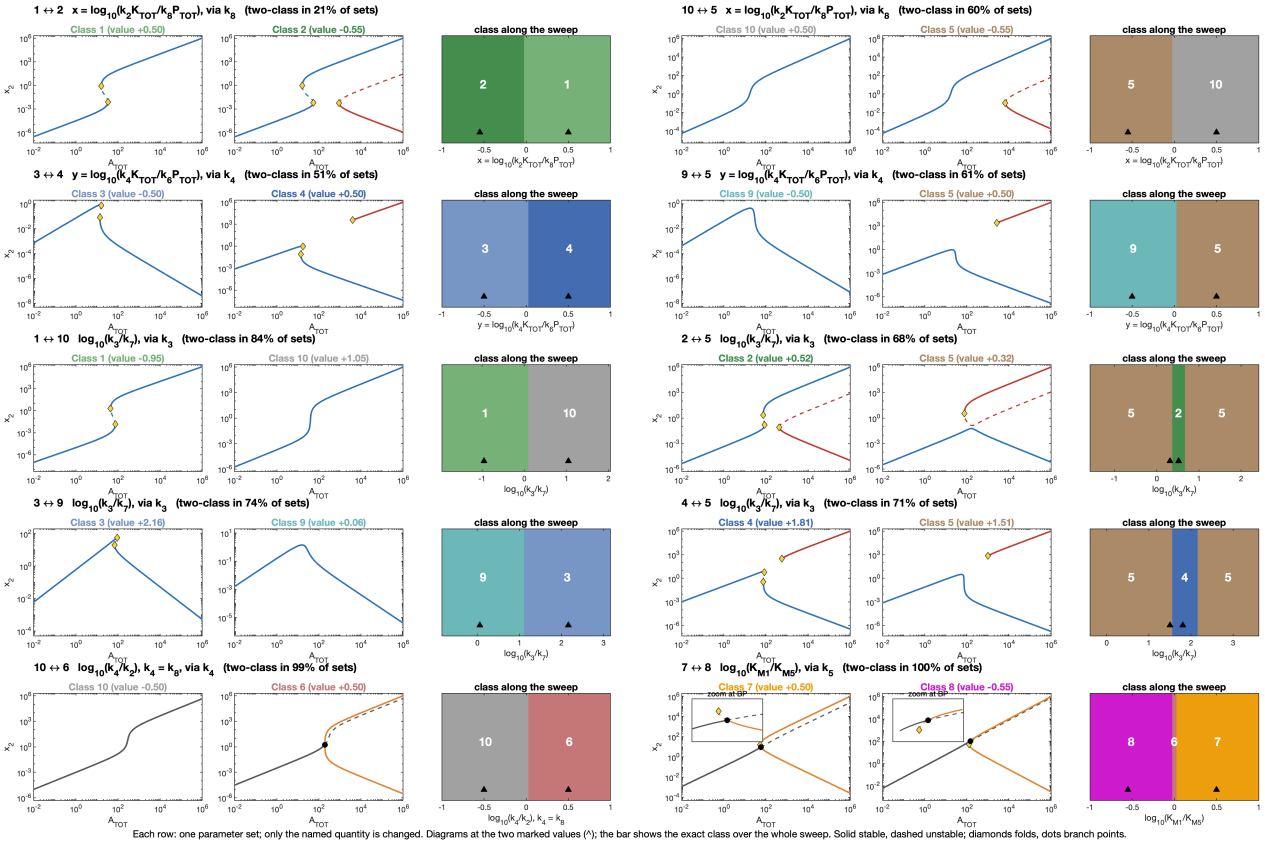
Edge	Kind	Controlling quantity	Two-class (%)	Best move, forward	Best move, back
$1 \leftrightarrow 2$	exact	$x < 0$	21	$k_8 \times 10$ (31%)	$k_8 \div 10$ (38%)
$10 \leftrightarrow 5$	exact	$x < 0$	60	$P_{\mathrm{TOT}} \times 10$ (28%)	$P_{\mathrm{TOT}} \div 10$ (23%)
$3 \leftrightarrow 4$	exact	$y > 0$	51	$k_4 \times 10$ (64%)	$k_4 \div 10$ (32%)
$9 \leftrightarrow 5$	exact	$y > 0$	61	$K_{\mathrm{TOT}} \times 10$ (78%)	$K_{\mathrm{TOT}} \div 10$ (36%)
$1 \leftrightarrow 10$	fold	main-branch fold (S)	84	$k_2 \times 10$ (72%)	$k_2 \div 10$ (36%)
$2 \leftrightarrow 5$	fold	main-branch fold (S)	68	$K_{\mathrm{TOT}} \div 10$ (100%)	$P_{\mathrm{TOT}} \div 10$ (30%)
$3 \leftrightarrow 9$	fold	main-branch fold (Z)	74	$P_{\mathrm{TOT}} \times 10$ (78%)	$k_3 \times 10$ (27%)
$4 \leftrightarrow 5$	fold	main-branch fold (Z)	71	$K_{\mathrm{TOT}} \times 10$ (99%)	$P_{\mathrm{TOT}} \times 10$ (29%)
$10 \leftrightarrow 6$	exact	$k_4 > k_2$	99	$k_4, k_8 \times 10$ (45%)	$K_{\mathrm{TOT}} \times 10$ (42%)
$6 \rightarrow 7$	exact	$K_{M5} < K_{M1}$	100	$k_1 \div 10$ (100%)	–
$6 \rightarrow 8$	exact	$K_{M5} > K_{M1}$	100	$k_1 \times 10^{0.5}$ (100%)	–
$6 \rightarrow 5$	symmetry	$u^* \neq r$	–	$k_2 \div 10^{0.5}$ (100%)	–
$6 \rightarrow 9$	symmetry	$u^* \neq r$	–	$K_{\mathrm{TOT}} \div 10$ (42%)	–
$6 \rightarrow 10$	symmetry	$u^* \neq r$	–	$K_{\mathrm{TOT}} \times 10$ (42%)	–

Supplementary Table 20. Two-class sweeps. One quantity swept over 41 points through the boundary in 80 sets of the source class (first class of the edge). Two-class: only the two classes of the edge occur along the sweep; switch: the second class occurs. Class 11: the labeller's "other" class. Source: `TR_sweep_summary.csv`.

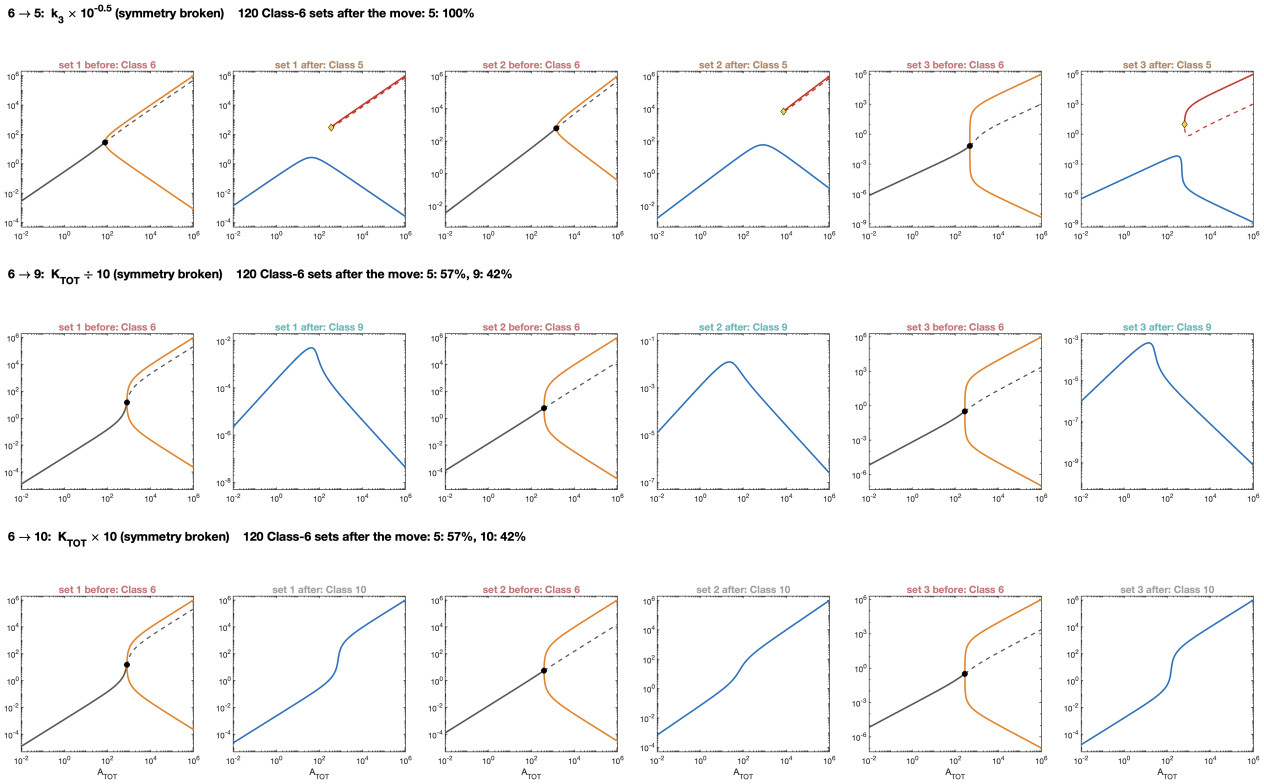
Edge	Swept quantity	Changed parameter	Two-class (%)	Switch (%)	Other classes met
$1 \leftrightarrow 2$	x	k_8	21.2	75.0	5, 10
$10 \leftrightarrow 5$	x	k_8	60.0	77.5	1, 2
$3 \leftrightarrow 4$	y	k_4	51.2	81.2	5, 9
$9 \leftrightarrow 5$	y	k_4	61.3	62.5	3
$1 \leftrightarrow 10$	$\log_{10}(k_3/k_7)$	k_3	83.8	86.2	2, 5
$2 \leftrightarrow 5$	$\log_{10}(k_3/k_7)$	k_3	67.5	100	1, 10
$3 \leftrightarrow 9$	$\log_{10}(k_3/k_7)$	k_3	73.8	86.2	4, 5
$4 \leftrightarrow 5$	$\log_{10}(k_3/k_7)$	k_3	71.2	100	3, 9
$10 \leftrightarrow 6$	$\log_{10}(k_4/k_2)$, $k_4 = k_8$	k_4, k_8	98.8	100	8
$7 \leftrightarrow 8$	$\log_{10}(K_{M1}/K_{M5})$	k_5	100	100	6 at $K_{M1} = K_{M5}$ only
$1 \leftrightarrow 2$	x	k_2	0	55.0	5, 10
$1 \leftrightarrow 2$	x	K_{TOT}	0	32.5	5, 9, 10, 11
$1 \leftrightarrow 2$	x	P_{TOT}	0	32.5	5, 9, 10, 11
$3 \leftrightarrow 4$	y	k_6	0	81.2	5, 9
$3 \leftrightarrow 4$	y	K_{TOT}	0	76.2	5, 9
$3 \leftrightarrow 4$	y	P_{TOT}	0	76.2	5, 9
$1 \leftrightarrow 10$	$\log_{10}(k_3/k_7)$	k_7	81.2	83.8	2, 5



Supplementary Fig. 19. Transition map of the ten classes. Each class is drawn as its own scatter panel in the plane of main-text Fig 7, $x = \log_{10}(k_2 K_{\mathrm{TOT}}/k_8 P_{\mathrm{TOT}})$ against $y = \log_{10}(k_4 K_{\mathrm{TOT}}/k_6 P_{\mathrm{TOT}})$: the class (colour) against the ensemble (grey), pink the detached-branch quadrant (left: Iterations 1–2; right: Iterations 3–4, where Class 10 is the symmetric monotone class and Classes 6–8 lie on the anti-diagonal). Edges connect the panels. Solid: exact condition; dashed: fold on the main branch (no closed form); dotted: symmetry breaking (any single change with $u^* \neq r$). Labels: the controlling quantity, the percentage of sets in which a sweep of that quantity alone switches between exactly the two classes, and the best single-parameter change in each direction with the percentage of the 120 source sets it converts. Code: `Class_transitions/TR_04_figures.m` (folder A).



Supplementary Fig. 20. Two-class switches. For every edge of the transition map, one parameter set of the source class in which a sweep of the controlling quantity (only that quantity changed) switches between exactly the two classes: exact bifurcation diagrams at the two marked values (▲) and the class over the whole sweep (bar). Detached-branch edges: x through k_8, y through k_4; fold edges: $\log_{10}(k_3/k_7)$ through k_3; branch point: $\log_{10}(k_4/k_2)$ with $k_4 = k_8$; Classes 7 and 8: $\log_{10}(K_{M1}/K_{M5})$ through k_5, with Class 6 only at $K_{M1} = K_{M5}$. Titles give the percentage of the 80 swept sets that are two-class. Blue: main branch; red: detached branch; grey: $u = r$; orange: crossing branch; solid stable, dashed unstable; diamonds folds, dots branch points. Code: `Class_transitions/TR_03_two_class_sweeps.m`, `TR_04_figures.m` (folder A).



Supplementary Fig. 21. Breaking the symmetry removes the branch point. Class-6 sets before and after one change that breaks $u^* = r$, three sets per move. A change of k_3 (which enters neither x nor y) keeps the set in the detached quadrant and gives Class 5: the upper half of the pitchfork becomes the detached branch. A tenfold change of K_{TOT} gives Class 9 if y becomes negative, Class 10 if x becomes positive, and Class 5 otherwise. Code: `Class_transitions/TR_04_figures.m` (folder A).

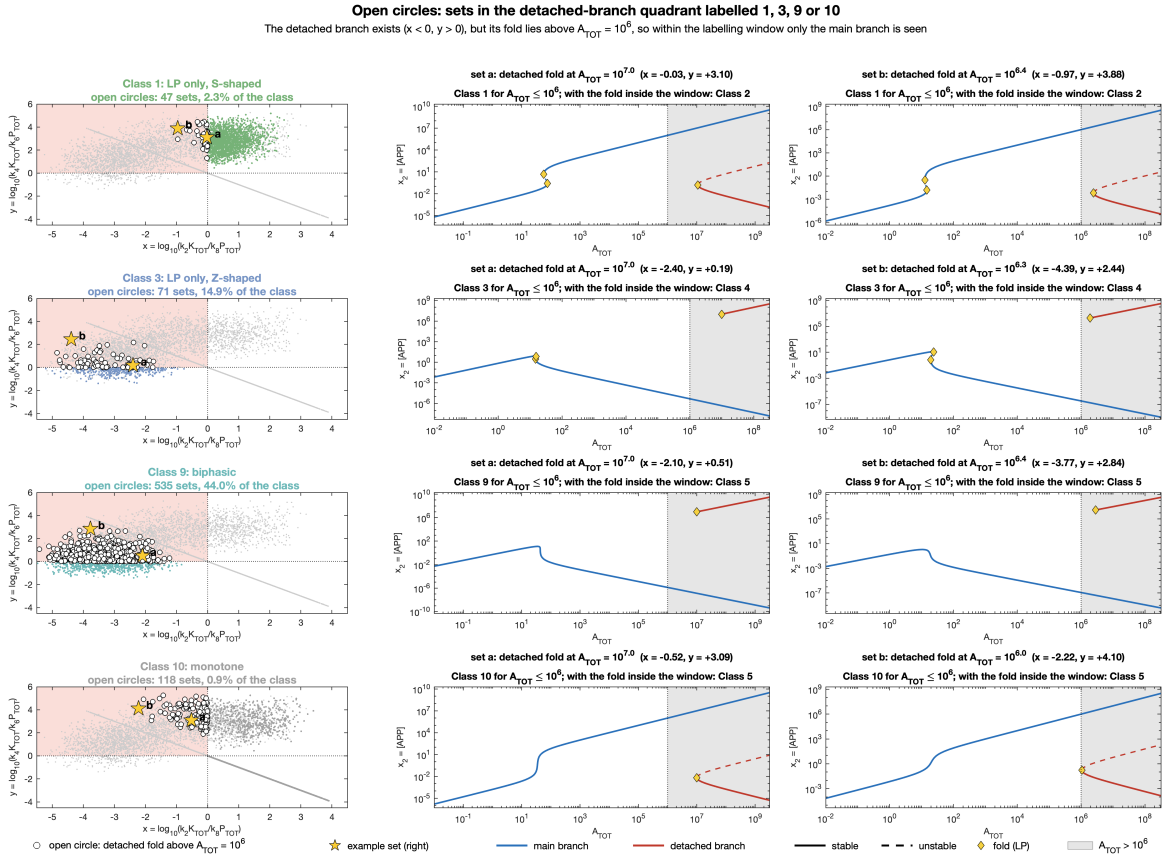
Supplementary Note 19. Detached branches beyond the labelling window

Classes are assigned from the singularities with $A_{\mathrm{TOT}} \leq 10^6$. A set that satisfies criterion (10) ($x < 0$, $y > 0$) has a detached branch, but if the fold of that branch lies above 10^6 the label counts only the main branch. Of the 39,968 sets used for machine learning, 771 (1.9%) are of this kind (the open circles of main-text Fig 7A): 47 of Class 1 (2.3% of the class), 71 of Class 3 (14.9%), 535 of Class 9 (44.0%) and 118 of Class 10 (0.9%; symmetric sets, which have no detached branch, excluded). In all 771 the detached fold lies between $A_{\mathrm{TOT}} = 10^{6.0}$ and $10^{10.7}$ (median $10^{6.7}$), and with the window widened to include it the sets become Class 2, 4, 5 and 5, respectively (Supplementary Fig. 22). They lie close to the edges of the quadrant: the median of $\min(-x, y)$ is 0.45 decades, against 1.21 decades for the sets of Classes 2, 4 and 5.

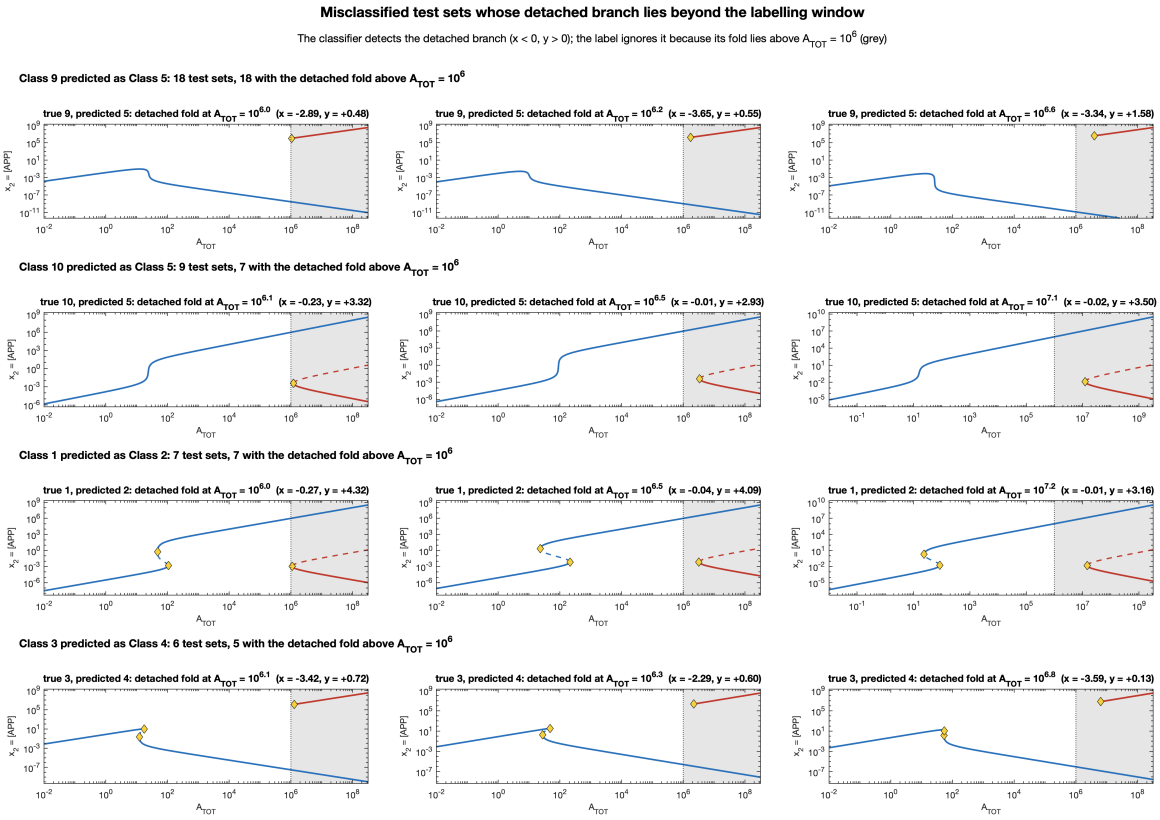
The classifier recognises a detached branch from the signs of x and y, and it predicts some of these sets as the class they would have with a wider window. Of the 591 errors of the pooled LightGBM classifier (LightGBM panel of Supplementary Fig. 12), 42 (7.1%) are such sets, against 1.9% of the ensemble (Supplementary Table 21, Supplementary Fig. 23); their detached folds lie at $A_{\mathrm{TOT}} = 10^{6.0}$–$10^{7.2}$, just beyond the window. They make up all 18 errors of Class 9 predicted as Class 5, all 7 of Class 1 predicted as Class 2, 7 of the 9 of Class 10 predicted as Class 5 and 5 of the 6 of Class 3 predicted as Class 4, whereas none of the six largest confusions (Supplementary Fig. 26) involves such a set. These errors arise from where the window is drawn and would disappear with a wider window; the errors of Supplementary Fig. 26, which concern the fold of the main branch, would not. The code is `Class_transitions/TR_05_window_sets.m` (folder A), adapted from the scripts of K. Sriram to the ensemble and labeller of this study.

Supplementary Table 21. Classifier errors from the labelling window. Pooled LightGBM, first held-out test set (591 errors of 7,994). Window sets: misclassified sets with $x < 0$, $y > 0$ and label 1, 3, 9 or 10 whose detached fold lies above $A_{\mathrm{TOT}} = 10^6$. All other confusions (Supplementary Table 14) contain no window set. Source: `Class_transitions/results/TR_window_confusions.csv`.

True → predicted	Errors	Window sets	%
9 → 5	18	18	100
9 → 3	9	1	11
10 → 5	9	7	78
1 → 2	7	7	100
3 → 4	6	5	83
10 → 2	3	2	67
9 → 4	2	2	100
All confusions	591	42	7.1



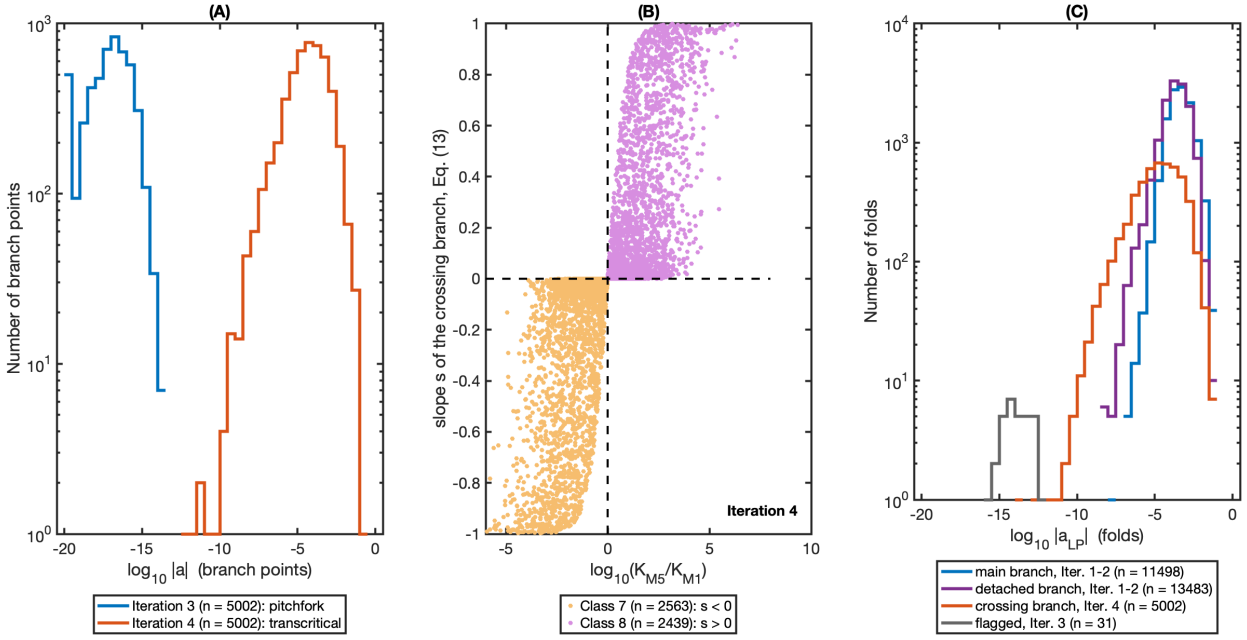
Supplementary Fig. 22. Open circles: detached branches beyond the labelling window. For Classes 1, 3, 9 and 10 (rows): left, the class (colour) against the ensemble (grey) in the plane of main-text Fig 7, with the sets inside the detached-branch quadrant (pink) as open circles and two of them starred (a: detached fold nearest to $A_{\mathrm{TOT}} = 10^7$, the set of main-text Fig 7B; b: the set deepest inside the quadrant); middle and right, the exact diagrams of sets a and b extended beyond the window (grey: $A_{\mathrm{TOT}} > 10^6$). Titles: the A_{TOT} of the detached fold, x and y, the label in the window and the class with the fold inside the window. Blue: main branch; red: detached branch; solid stable, dashed unstable; diamonds folds. Code: `Class_transitions/TR_05_window_sets.m` (folder A).



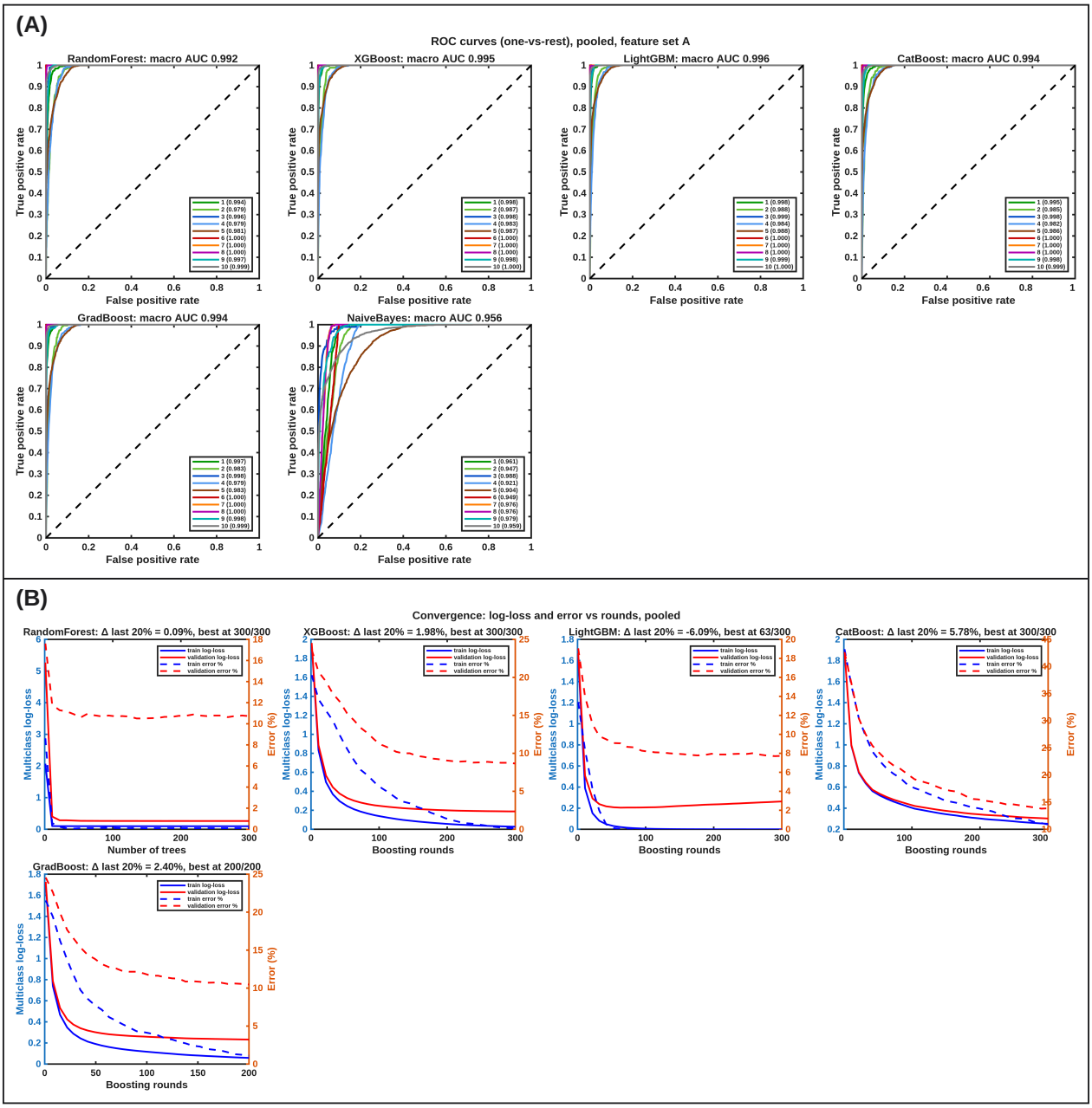
Supplementary Fig. 23. Misclassified test sets whose detached branch lies beyond the window. The four confusions with the most window sets (Supplementary Table 21); for each, up to three test sets spanning the range of detached-fold positions, with exact diagrams extended beyond the window (grey: $A_{\mathrm{TOT}} > 10^6$). The classifier predicts the detached-branch class; the label, restricted to $A_{\mathrm{TOT}} \leq 10^6$, does not count the detached branch. Companion to Supplementary Fig. 26, whose errors concern the main branch. Code: `Class_transitions/TR_05_window_sets.m` (folder A).

Supplementary Note 20. Figures and tables referred to from the main text

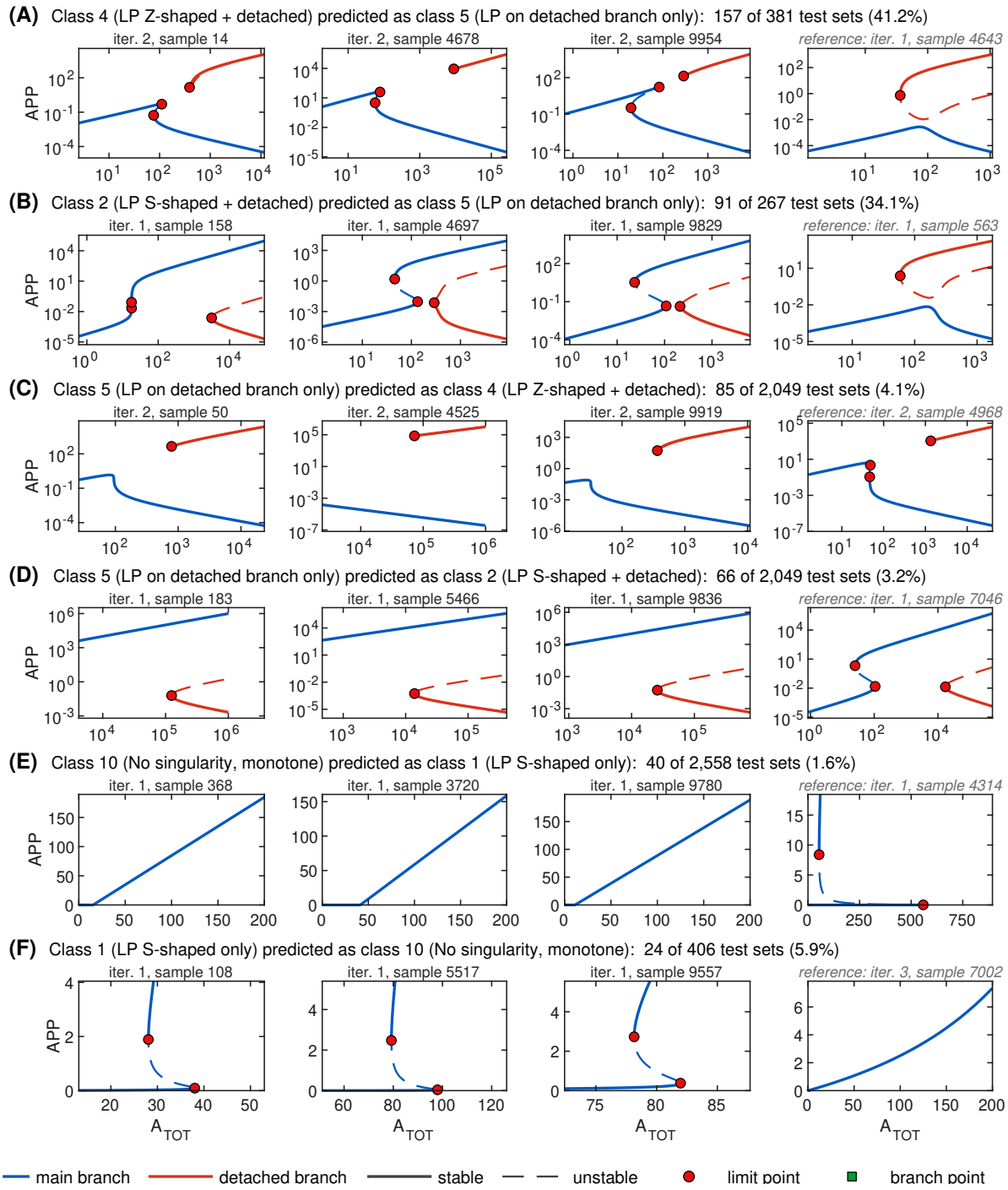
The four figures and eight tables below support the main text and are cited there; they give the normal-form coefficients of every singularity, the ROC and convergence curves, the pooled confusion matrix and representative misclassified sets, the dependence of the multistationary fraction on the A_{TOT} window, the comparisons with EMBED and with single-start continuation, the design-constrained feature ratios, the feature-set ablation, and the MATCONT settings and software used.



Supplementary Fig. 24. Normal-form coefficients at every branch point and fold (computed in MATCONT's conventions). (A) Quadratic coefficient $|a|$ of the branching equation at all 5,002 branch points of Iteration 3 (blue) and all 5,002 of Iteration 4 (orange); logarithmic counts, values below 10^{-20} drawn at 10^{-20}. a vanishes to round-off under full symmetry ($|a| \le 3\times10^{-14}$, pitchfork) and is non-zero once the first binding step is asymmetric ($|a| \ge 4.3\times10^{-13}$, transcritical). (B) Iteration 4: slope s of the crossing branch at the branch point, Eq. (13), against $\log_{10}(K_{M5}/K_{M1})$. All 2,563 Class-7 sets (orange) have $s<0$ and $K_{M5}<K_{M1}$, all 2,439 Class-8 sets (violet) $s>0$ and $K_{M5}>K_{M1}$, criterion (14). (C) Fold coefficient $|a_{LP}|$ at all 29,983 folds inside the window: main branch of Iterations 1–2 (11,498), detached branch of Iterations 1–2 (13,483) and crossing branch of Iteration 4 (5,002), and at the 31 flagged folds of Iteration 3; logarithmic counts. No fold is degenerate except the flagged folds at the pitchfork. Code: `Fig_normal_forms_v3.m` (folder A), from the files `Normal_Forms_v3_iter<i>.csv` written by `Ensemble_LHS_v3.m`.



Supplementary Fig. 25. ROC curves and convergence of the pooled ten-class classifiers. (A) One-vs-rest receiver operating characteristic curves on the held-out test set for each model (feature set A); the legend gives the AUC of every class, the panel title the macro-averaged AUC. Dashed: chance. The curves of Classes 2, 4 and 5 (detached-branch folds, with or without main-branch hysteresis) rise most slowly; Classes 6, 7 and 8 reach AUC ≈ 1 with every tree ensemble. ROC curves within each iteration are given in Supplementary Fig. 10. (B) Multiclass log-loss (solid; left axis) and error (dashed; right axis) on the training set (blue; subsample of 5,000) and the validation set (red) as a function of the number of boosting rounds (trees for the random forest), feature set A. Panel titles give the relative change of the validation log-loss over the last 20% of rounds and the round with the smallest validation log-loss. Per-class curves: Supplementary Fig. 13; within iterations: Supplementary Fig. 14. Code: `ML_v3_10class_parallel.m` (folder A).



Supplementary Fig. 26. Representative misclassified parameter sets: true class against predicted class. Exact bifurcation diagrams ($x_2 = [APP]$ against A_{TOT}) of held-out test sets that the pooled LightGBM classifier assigns to the wrong class, for the six most frequent confusions (A–F); each row heading gives the true and the predicted class and the number of test sets concerned. The first three panels of a row are misclassified sets, the fourth (grey label) a correctly classified set of the predicted class; iteration and sample number are given above each panel. Solid, stable; dashed, unstable. The six confusions account for 463 of the 591 misclassified test sets. Code: `ML_v3_misclassified_figure17.m` (folder A).

Supplementary Table 22. Dependence of the multistationary fraction on the A_{TOT} window. Fraction of parameter sets with at least one fold or branch point at A_{TOT} below the stated bound, for the 40,000 exactly labelled parameter sets; in the first row a failed single-start run (362 sets) counts as no singularity found. For all A_{TOT}, a set is counted if it has a singularity in the window, satisfies criterion (10) or, in Iterations 3 and 4, has $k_4 > k_2$; folds on the main branch above 10^6 are not computed, so the values for Iterations 1 and 2 are lower bounds. Code: `Ensemble_LHS_v3.m`, `Manuscript_numbers_v3.m` (folder A); first row `make_comparison_figures.m` (folder B).

	Iter. 1	Iter. 2	Iter. 3	Iter. 4
Single-start MATCONT (variable window, median ≈ 1,240)	27.4%	19.1%	44.8%	44.1%
Exact, $A_{\mathrm{TOT}} \leq 10^2$	45.9%	42.9%	16.9%	18.7%
Exact, $A_{\mathrm{TOT}} \leq 10^3$	62.8%	63.6%	43.7%	46.8%
Exact, $A_{\mathrm{TOT}} \leq 10^4$	67.6%	73.8%	49.1%	49.6%
Exact, $A_{\mathrm{TOT}} \leq 10^6$ (labels)	72.1%	87.8%	50.0%	50.0%
Exact, all A_{TOT}	≥73.3%	≥93.2%	50.0%	50.0%

Supplementary Table 23. EMBED and the present analysis.. Code: `ML_v3_10class_parallel.m` (folder A).

	EMBED	Present work
Model	reduced 2D Michaelis–Menten (QSSA)	full mass-action, six ODEs, 14 parameters
Ensemble	500–5,000 sets per strategy	40,000 sets, four symmetry-designed iterations
Labelling	single-start MATCONT, branch switching at BPs	inverse method (all branches); MATCONT seeded on every branch
Detached branches	not reachable	found by construction
Classes	seven dynamical classes	ten bifurcation classes
Bifurcation type	Sotomayor's conditions at representative sets	normal-form coefficients at all 40,018 singularities
Class boundaries	numerical	closed form (one boundary open)
Machine learning	–	seven classifiers; 93% accuracy
Class transitions	–	exact conditions; single-parameter scan and two-class sweeps

Supplementary Table 24. Single-start MATCONT, the inverse method and MATCONT seeded on every branch. Code: `SingleStart_v3.m` (single start), `Ensemble_LHS_v3.m` (validation) (folder A).

	MATCONT, single start	Inverse method	MATCONT, every branch
Starting point	ode15s state at $A_{\mathrm{TOT}} = 0.1$	none (grid in $u = K/P$)	stable state on each branch
Detached branches	not reachable	found by construction	found (seeds from inverse method)
A_{TOT} range	variable (median 1,244)	fixed, $\leq 10^6$	fixed, $\leq 10^6$
Bistable, Iter. 1/2	27.4% / 19.1%	72.1% / 87.8%	agreement 82.0% / 81.3%
Bistable, Iter. 3/4	44.8% / 44.1%	50.0% / 50.0%	agreement 99.4% / 86.6%
Cost per parameter set	0.44 s	0.011 s	8.0 s

Supplementary Table 25. Normal-form coefficients at every singularity of the exact branches. Coefficients in MATCONT's normalisation ($\|q\| = 1$, $p^{\mathsf{T}}q = 1$), computed from the analytic derivatives at every singularity inside $A_{\mathrm{TOT}} \le 10^6$; fold coefficients agree with MATCONT to a median relative difference of 1.4×10^{-8}. Ranges: minimum–maximum of the absolute value, medians in brackets. Code: `Ensemble_LHS_v3.m` (`support/normal_forms.m`), `Manuscript_numbers_v3.m` (folder A).

Singularity	n	Coefficients	Conclusion
Branch points, Iter. 3	5,002	$\lvert a\rvert \le 3\times10^{-14}$ (1.0×10^{-17})	$a = 0$: pitchforks
Branch points, Iter. 4	5,002	$\lvert a\rvert = 4.3\times10^{-13}$–0.12 ($5.2\times10^{-5}$)	$a \neq 0$: transcritical
Folds, main branch (Iter. 1–2)	11,498	$\lvert a_{LP}\rvert = 2.7\times10^{-8}$–0.091 ($4.2\times10^{-4}$)	non-degenerate saddle-nodes
Folds, detached branch (Iter. 1–2)	13,483	$\lvert a_{LP}\rvert = 3.9\times10^{-9}$–0.082 ($2.5\times10^{-4}$)	non-degenerate saddle-nodes
Folds, crossing branch (Iter. 4)	5,002	$\lvert a_{LP}\rvert = 3.1\times10^{-14}$–0.057 ($2.2\times10^{-5}$)	saddle-nodes; smallest where the fold approaches the BP
Flagged folds (Iter. 3)	31	$\lvert a_{LP}\rvert \le 9.4\times10^{-13}$ (1.1×10^{-14})	degenerate: turning point at the pitchfork

Supplementary Table 26. Design-constrained feature ratios. Value of each kinase–phosphatase ratio in the four sampling iterations. = 1: the ratio is fixed at one by the symmetry constraint in every parameter set of the iteration. Otherwise the ratio is free, and the range it can take under the sampling bounds of main-text Table 1 is given. Code: `ML_v3_10class_parallel.m` (folder A).

Iteration	$K_{\mathrm{TOT}}/P_{\mathrm{TOT}}$	k_1/k_5	k_{11}/k_{55}	k_2/k_6	k_3/k_7	k_{33}/k_{77}	k_4/k_8
1 (S-dominant)	1–50	10^{-4}–10^{-1}	10^{-2}–10^{4}	10^{-2}–10^{2}	10^{-4}–10^{1}	10^{-2}–10^{4}	10^{-2}–10^{4}
2 (Z-dominant)	0.02–1	10^{1}–10^{4}	10^{-4}–10^{2}	10^{-2}–10^{2}	10^{-2}–10^{4}	10^{-4}–10^{2}	10^{-2}–10^{2}
3 (full symmetry)	= 1	= 1	= 1	= 1	= 1	= 1	= 1
4 (binding asymmetry)	= 1	10^{-4}–10^{4}	10^{-4}–10^{4}	= 1	= 1	= 1	= 1

Calculation. The two parameters of a free ratio x/y are sampled independently, $x \in [x_{\min}, x_{\max}]$ and $y \in [y_{\min}, y_{\max}]$ (bounds of main-text Table 1). The ratio is smallest when x is at its lower and y at its upper bound, and largest in the opposite case:

$$\frac{x_{\min}}{y_{\max}} \;\le\; \frac{x}{y} \;\le\; \frac{x_{\max}}{y_{\min}}.$$

Iteration 1: $K_{\mathrm{TOT}}/P_{\mathrm{TOT}} \in [100/100,\ 500/10] = [1,\ 50]$; $k_1/k_5 \in [10^{-3}/10^{1},\ 10^{-2}/10^{-1}] = [10^{-4},\ 10^{-1}]$; $k_{11}/k_{55} \in [10^{-3}/10^{-1},\ 10^{1}/10^{-3}] = [10^{-2},\ 10^{4}]$; $k_2/k_6 \in [10^{-3}/10^{-1},\ 10^{-1}/10^{-3}] = [10^{-2},\ 10^{2}]$; $k_3/k_7 \in [10^{-3}/10^{1},\ 10^{0}/10^{-1}] = [10^{-4},\ 10^{1}]$; $k_{33}/k_{77} \in [10^{-3}/10^{-1},\ 10^{1}/10^{-3}] = [10^{-2},\ 10^{4}]$; $k_4/k_8 \in [10^{-1}/10^{1},\ 10^{1}/10^{-3}] = [10^{-2},\ 10^{4}]$.
Iteration 2: $K_{\mathrm{TOT}}/P_{\mathrm{TOT}} \in [10/500,\ 100/100] = [0.02,\ 1]$; $k_1/k_5 \in [10^{-1}/10^{-2},\ 10^{1}/10^{-3}] = [10^{1},\ 10^{4}]$; $k_{11}/k_{55} \in [10^{-3}/10^{1},\ 10^{-1}/10^{-3}] = [10^{-4},\ 10^{2}]$; $k_2/k_6 \in [10^{-3}/10^{-1},\ 10^{-1}/10^{-3}] = [10^{-2},\ 10^{2}]$; $k_3/k_7 \in [10^{-1}/10^{1},\ 10^{1}/10^{-3}] = [10^{-2},\ 10^{4}]$; $k_{33}/k_{77} \in [10^{-3}/10^{1},\ 10^{-1}/10^{-3}] = [10^{-4},\ 10^{2}]$; $k_4/k_8 \in [10^{-1}/10^{1},\ 10^{1}/10^{-1}] = [10^{-2},\ 10^{2}]$.
Iteration 4: k_1/k_5 and $k_{11}/k_{55} \in [10^{-3}/10^{1},\ 10^{1}/10^{-3}] = [10^{-4},\ 10^{4}]$.
A ratio fixed by a symmetry constraint ($x = y$ in every parameter set, e.g. $K_{\mathrm{TOT}} = P_{\mathrm{TOT}}$, $k_1 = k_5$; main-text Table 1) equals one.

Supplementary Table 27. Number of input features and LightGBM accuracy for each feature set. Each entry gives the number of features used and, in brackets, the held-out accuracy (%, mean over three stratified splits). Zero-variance and duplicated columns are removed before training; for set D the selected features are recomputed in each training split (mean number given). Code: `ML_v3_10class_parallel.m` (folder A).

Feature set	Pooled	Iter. 1	Iter. 2	Iter. 3	Iter. 4
A: parameters + log-ratios	105 (92.66)	105 (86.82)	105 (84.48)	43 (99.98)	59 (99.03)
B: A without constrained ratios	98 (91.43)	98 (85.33)	98 (82.83)	43 (99.98)	57 (98.65)
C: $\log_{10}$ parameters	14 (89.13)	14 (82.05)	14 (79.90)	7 (99.15)	9 (96.82)
D: JSD + broken stick	64.3 (92.71)	18.0 (84.33)	36.3 (85.05)	2.0 (99.98)	5.0 (97.33)
E: CLR of 14 log-parameters	14 (88.79)	14 (82.50)	14 (79.68)	7 (98.93)	9 (97.28)
F: CLR of rates + $\log(K_{\mathrm{TOT}}/P_{\mathrm{TOT}})$	13 (89.77)	13 (83.67)	13 (81.05)	6 (98.90)	8 (97.45)
Majority baseline	(32.00)	(38.45)	(64.00)	(50.15)	(50.00)

Supplementary Table 28. MATCONT continuation settings. Single-start labels and the validation runs of the exact pipeline, in which every branch of steady states is seeded separately. W is the upper end of the A_{TOT} window. MATCONT 7.6, MATLAB R2026a. Code: `SingleStart_v3.m`, `Ensemble_LHS_v3.m`, `support/matcont_check.m` (folder A).

`contset` option	Single start	Validation runs
InitStepSize	1	10^{-4}
MinStepSize	10^{-5}	10^{-10}
MaxStepSize	10^{-3}	$W/10^{4}$
MaxNewtonIters	10	10
MaxCorrIters	15	15
MaxTestIters	10	20
VarTolerance	10^{-7}	10^{-8}
FunTolerance	10^{-7}	10^{-8}
TestTolerance	10^{-6}	10^{-7}
TestFunctions	1	default
Singularities	1	1
Locators	[1 1 1]	[1 1 1]
Adapt	5	3
MaxNumPoints	1,500	20,000
CheckClosed	50	50
Direction	forward (BP extension: both)	both, on every branch
Starting point	`ode15s` end state at $A_{\mathrm{TOT}} = 0.1$	a stable steady state on each branch (exact)
A_{TOT} range	variable (1,500 points)	trimmed to $[0, W]$

Supplementary Table 29. Software dependencies. Steps that require each package; versions used in this study.

Package	Required for	Version
MATLAB	all steps	26.1 (R2026a)
Statistics and Machine Learning Toolbox	Latin hypercube sampling (`lhsdesign`); linear SVM, t-SNE, kNN in the ML analysis	26.1 (R2026a)
Parallel Computing Toolbox	parallel labelling (`parfor`); optional	26.1 (R2026a)
MATCONT	validation runs of the exact labels; single-start labels	7.6
Chemical Reaction Network Toolbox	structural network analysis (Basic and Higher Deficiency Reports)	2.35
Wolfram Mathematica	symbolic derivation and checks of Eqs (1)–(14) and of the normal-form coefficients (Supplementary File 6)	15
Signal Processing Toolbox	`findpeaks`, single-start labels only	26.1 (R2026a)
Python	machine learning (called through MATLAB's Python interface)	3.9
NumPy	numerical arrays	2.0.2
LightGBM	gradient-boosted trees (primary model)	4.6.0
XGBoost	gradient-boosted trees	2.1.4
CatBoost	gradient-boosted trees	1.2.10
scikit-learn	random forest, gradient boosting, naive Bayes, metrics	1.6.1

Supplementary Note 21. Experimentally accessible quantities that set the class

The class boundaries of the main text are inequalities on maximal rates ($V_{\max} = k_{\text{cat}}\, E_{\text{TOT}}$), on enzyme totals and on Michaelis constants, not on individual binding and unbinding rates. This note summarises which of these quantities can be tuned and measured, and therefore which of the class boundaries can be targeted in a synthetic construct (Supplementary Table 30).

Enzyme and substrate levels. The total substrate A_{TOT}, the bifurcation parameter of every diagram, and the enzyme totals K_{TOT} and P_{TOT} are the most accessible quantities. They can be set with inducible promoters [S1] and with designed ribosome binding sites over a range of about 10^5-fold [S2], and protein lifetimes can be shortened with degradation tags [S3]; absolute protein levels are measured by quantitative proteomics [S4] or by targeted mass spectrometry with isotope-labelled standards [S5]. $K_{\text{TOT}}/P_{\text{TOT}}$ sets the orientation of main-branch hysteresis (S- or Z-shaped) and enters both inequalities of criterion (10): a twofold change in phosphatase expression shifts $\log_{10}(k_8 P_{\text{TOT}}/k_2 K_{\text{TOT}})$ by 0.3, enough to move a construct across the detached-branch boundary when it lies close to it.

Maximal rates: criterion (10). The two ratios that decide whether a detached branch exists, $k_8 P_{\text{TOT}}/k_2 K_{\text{TOT}}$ at the first site and $k_4 K_{\text{TOT}}/k_6 P_{\text{TOT}}$ at the second, are ratios of maximal rates. A maximal rate is what an activity assay at saturating substrate reports, so these ratios can be measured without separating catalytic constants from enzyme levels [S6]. The individual phosphoforms needed to assign the rates to the two sites can be resolved by Phos-tag electrophoresis [S7] and quantified by targeted mass spectrometry [S5]. The ratios can be tuned through expression levels and through catalytic mutations or the sequence context of the phosphosites.

Site order and binding. The branch-point condition $k_4 > k_2$ compares the catalytic constants of the kinase at the two sites; such site-specific rates were measured for the dual phosphorylation of MAP kinase, which established its distributive, two-step mechanism [S8]. The orientation of the transcritical crossing (Classes 7 and 8) is set by K_{M1} against K_{M5}, which can be measured by Michaelis–Menten kinetics [S6] and modified through docking interactions: docking motifs recruit kinases, phosphatases and substrates in MAP kinase pathways [S9], engineered docking peptides and binding-domain variants of graded affinity have been used to build synthetic phospho-regulons [S10], and scaffold recruitment has been used to reshape MAP kinase signalling dynamics [S11].

What cannot be controlled. Individual binding and unbinding rates ($k_1, k_{11}, k_3, k_{33}, \ldots$) are not separately accessible in vivo; they enter the classes only through K_{M1}–K_{M7} and the catalytic constants. Exact kinase–phosphatase symmetry, on which the branch-point classes (6–8) depend, cannot be engineered or maintained exactly, so these classes are not robust design targets. The robust targets are the classes whose boundaries are inequalities on maximal rates and enzyme totals (Classes 1, 2 and 5 and their Z-shaped counterparts). Distributive dual phosphorylation, the mechanism of the model, can generate bistability without explicit feedback [S12], and bistable switches have been observed in MAP kinase signalling [S13] and built from engineered phosphorylation networks [S14]; the class map states which measured ratios a construct needs for its switch to lie on the main branch or on a detached branch.

Implication for prediction. A classifier restricted to measurable features, $\log_{10}(K_{\text{TOT}}/P_{\text{TOT}})$, the two maximal-rate ratios and, where available, Michaelis-constant ratios, is the deployable form of the classifier; the decisive two-class tests (Supplementary Note 13) indicate that two such ratios suffice where two classes differ in one condition. Sampling parameter sets within the measurement uncertainty of a construct and classifying them gives a class probability that shows how close the construct lies to a boundary.

Supplementary Table 30. Experimentally accessible quantities that set the class. Tier 1, tunable and measurable in cells; tier 2, the ratios of criterion (10), measurable as maximal rates; tier 3, accessible with in vitro kinetics and affinity engineering. References as in Supplementary Note 21.

Tier	Quantity	Role on the class map	How to tune	How to measure
1	A_{TOT}	bifurcation parameter (dose axis)	inducible promoter [S1]; RBS design [S2]	fluorescent fusion; absolute proteomics [S4, S5]
1	$K_{\mathrm{TOT}}/P_{\mathrm{TOT}}$	S vs Z orientation; enters both ratios of criterion (10)	promoter and RBS strength [S1, S2]; degradation tags [S3]	absolute protein quantification [S4, S5]
2	$k_8P_{\mathrm{TOT}}/k_2K_{\mathrm{TOT}}$	site 1 of criterion (10): detached branch (Classes 2, 4, 5)	expression; catalytic mutations; phosphosite sequence	activity assay at saturating substrate ($V_{\max}$) [S6]
2	$k_4K_{\mathrm{TOT}}/k_6P_{\mathrm{TOT}}$	site 2 of criterion (10)	as above, at site 2	Phos-tag electrophoresis [S7]; targeted mass spectrometry [S5]
3	k_4/k_2	branch-point condition $k_4 > k_2$ (Classes 6–8)	priming, site order, sequence context	time-resolved phosphoform analysis [S8]
3	K_{M1}/K_{M5}	Class 7 vs Class 8	docking motifs and binding-domain affinity [S9, S10]; scaffolds [S11]	Michaelis–Menten kinetics [S6]
–	individual k_{on}, k_{off}	enter only through K_M	not separately controllable in vivo	in vitro only

Supplementary References

References

[S1] Gossen M, Bujard H. Tight control of gene expression in mammalian cells by tetracycline-responsive promoters. Proc Natl Acad Sci USA. 1992;89(12):5547–51. doi:10.1073/pnas.89.12.5547.

[S2] Salis HM, Mirsky EA, Voigt CA. Automated design of synthetic ribosome binding sites to control protein expression. Nat Biotechnol. 2009;27(10):946–50. doi:10.1038/nbt.1568.

[S3] Andersen JB, Sternberg C, Poulsen LK, Bjorn SP, Givskov M, Molin S. New unstable variants of green fluorescent protein for studies of transient gene expression in bacteria. Appl Environ Microbiol. 1998;64(6):2240–6. doi:10.1128/AEM.64.6.2240-2246.1998.

[S4] Schwanhäusser B, Busse D, Li N, et al. Global quantification of mammalian gene expression control. Nature. 2011;473:337–42. doi:10.1038/nature10098.

[S5] Gerber SA, Rush J, Stemman O, Kirschner MW, Gygi SP. Absolute quantification of proteins and phosphoproteins from cell lysates by tandem MS. Proc Natl Acad Sci USA. 2003;100(12):6940–5. doi:10.1073/pnas.0832254100.

[S6] Cornish-Bowden A. Fundamentals of Enzyme Kinetics. 4th ed. Weinheim: Wiley-Blackwell; 2012.

[S7] Kinoshita E, Kinoshita-Kikuta E, Takiyama K, Koike T. Phosphate-binding tag, a new tool to visualize phosphorylated proteins. Mol Cell Proteomics. 2006;5(4):749–57. doi:10.1074/mcp.T500024-MCP200.

[S8] Ferrell JE Jr, Bhatt RR. Mechanistic studies of the dual phosphorylation of mitogen-activated protein kinase. J Biol Chem. 1997;272(30):19008–16. doi:10.1074/jbc.272.30.19008.

[S9] Tanoue T, Adachi M, Moriguchi T, Nishida E. A conserved docking motif in MAP kinases common to substrates, activators and regulators. Nat Cell Biol. 2000;2(2):110–6. doi:10.1038/35000065.

[S10] Gordley RM, Williams RE, Bashor CJ, Toettcher JE, Yan S, Lim WA. Engineering dynamical control of cell fate switching using synthetic phospho-regulons. Proc Natl Acad Sci USA. 2016;113(47):13528–33. doi:10.1073/pnas.1610973113.

[S11] Bashor CJ, Helman NC, Yan S, Lim WA. Using engineered scaffold interactions to reshape MAP kinase pathway signaling dynamics. Science. 2008;319(5869):1539–43. doi:10.1126/science.1151153.

[S12] Markevich NI, Hoek JB, Kholodenko BN. Signaling switches and bistability arising from multisite phosphorylation in protein kinase cascades. J Cell Biol. 2004;164(3):353–9. doi:10.1083/jcb.200308060.

[S13] Xiong W, Ferrell JE Jr. A positive-feedback-based bistable 'memory module' that governs a cell fate decision. Nature. 2003;426:460–5. doi:10.1038/nature02089.

[S14] Mishra D, Bepler T, Teague B, Berger B, Broach J, Weiss R. An engineered protein-phosphorylation toggle network with implications for endogenous network discovery. Science. 2021;373(6550):eaav0780. doi:10.1126/science.aav0780.