

INFORMATION TOPOLOGY OF KINETIC MODELS OF METABOLISM

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Lei Huang

January 2017

© 2017 Lei Huang
ALL RIGHTS RESERVED

INFORMATION TOPOLOGY OF KINETIC MODELS OF METABOLISM

Lei Huang, Ph.D.

Cornell University 2017

Interpreting mathematical models in general as functions that map parameters to model behaviors (for example, how reaction rates depend on reactant concentrations, or how protein concentrations depend on time), and interpreting such functions geometrically, lead to the field of *information geometry*: model behaviors now can be represented as a manifold with parameters as its coordinates. Such a perspective leads to conceptual elegance and practical improvements for standard modeling tasks such as parameter estimation and model selection. Importantly, the task of model reduction now becomes approximating a manifold by its boundaries, and the limiting model behaviors at the boundaries also give rise to scientific insight on how collective behaviors of a system emerge from its microscopic mechanisms, a central focus for systems science including systems biology. Realizing the importance of manifold boundaries, the incipient field of *information topology* attempts a topological characterization of all boundaries of a manifold, and constitutes the starting point of this thesis.

We propose a *system-component formulation* of system models, a class of models that encompass typical systems biology models. The simple formulation greatly organizes our understanding of the boundary structure of a system model: first, it partitions the set of boundaries into *component boundaries* and *emergent boundaries*, and explains their origins; second, it explains the origin of *structural nonidentifi-*

bility and the difficulty of resolving it; third, it helps explain additional features of the boundary structure, namely *combinatoriality* and *symmetry*. Next, we interpret structural nonidentifiability geometrically and develop a general method for resolving it using the topological characterization of a manifold; we call the method *manifold boundary identification method*. Lastly, these results naturally lead to an *algorithm* for constructing the topological characterization of a system model.

Applying these general insights to kinetic models of biochemical reaction networks, especially metabolic networks, yield some partial results. First, we provide complete topological characterizations of some rate laws commonly used in systems biology such as Michaelis-Menten, which pave way for similar characterizations of system models constructed using the rate laws. Second, we formalize the common system behaviors of metabolic networks, cast them into the system-component formulation, and characterize the resulting functional compositional structure, which for some behaviors translates to improved algorithms of boundary exploration for systems models. Third, focusing on the metabolic behaviors of how network fluxes depend on external metabolite concentrations, which are commonly measured in metabolic research, we characterize some of its mathematical structures, namely *similarity*, which gives metabolic networks interpretations as *generalized reactions* with the system behaviors as *generalized rate laws*, and *modularity*, which decomposes a metabolic network into interpretable modules in a formal and precise way; these insights, together with the ideas and techniques from information geometry and information topology, hold the promise of shedding light on some of long-standing problems in metabolic modeling.

BIOGRAPHICAL SKETCH

The author was born in the city of Wuhan in central China, and spent his early childhood there with his grandparents who instilled in him a love of nature and mathematics, as his grandfather often took him to the countryside to see the cattle and encouraged him to develop the "mathematical brain" of his grandmother who was an engineering professor at a top university of the country. Later he moved to a small city in southern China to live with his parents, grew up and received the standard Chinese pre-college education there. On the morning of deciding his college major, having narrowed down to mathematics and biology he was leaning towards mathematics, but a family friend who happened to be visiting at this time convinced his mother that biology was better and he followed them. Consequently, he spent his first a few years in college memorizing the latin name of the magnolia family and the chemical structures of the 20 common amino acids, while sitting through calculus classes designed for humanities students that discussed the Marxist dialectic between differentiation and integration. Also during this time, he befriended a butterfly taxonomist and they together spent uncountable hours in the wild collecting specimen, rearing caterpillars and pursuing wildlife photography, which provided him with not only much simple joy, but also first-hand experience with natural history, biogeography and polymorphism which got him interested in evolution. In his junior year, he joined the lab of Professor Chung-I Wu who convinced him that "nothing in evolution makes sense except in the light of genetics", and began researching on evolutionary genetics under Professor Wu's guidance, developing statistical models for testing

evolutionary hypotheses using genomic data, etc. He came to Cornell intending to continue studying evolutionary genetics. One September Friday afternoon that year, physicist-turned-systems-biologist Eric Siggia went to his genetics department and gave a talk, in which the audience were challenged "why is biology more than genetics and biochemistry?" After the talk and a few weeks of reading and thinking, he became convinced of the talk's point that systems biology in a sense reveals something more fundamental about life than traditional biology, and decided to pursue it. However, soon he realized that studying systems biology requires a different mathematical background from evolutionary genetics, which he lacked. Despite his lack of background, Professor Chris Myers kindly took him and gave him ample time to learn. Later in graduate school, he attended a series of short talks by Professor James Sethna on information geometry, a metatheory that promises to explain "why is science possible?" and clicks with his philosophical penchant. Hence this thesis, applying information geometry (and its extension information topology) to systems biology modeling. Upon finishing graduate school, his intermediate-term plan is to better understand the technical machineries that define our era, epitomized by data science and biomedical technology, and their implications on humanity.

To the loving memory of my grandfather,
and to my grandmother

ACKNOWLEDGEMENTS

It gives me profound pleasure to express my heart-felt gratitude at the end of this long journey in graduate school.

I thank Professor Chris Myers, for the opportunity of this journey, the supreme patience and gentleness in his approach on advising students, the great intellectual freedom afforded me and the innumerable teachings over the years. I thank Professor James Sethna, who, through a rare combination of seriousness, wisdom, intellect, positivity, articulation and charm, exerted powerful influence on any student coming into contact with him, and in this way I learned much from his mere presence. Also, I thank him, Mark Transtrum and others for their great work on information geometry and information topology, which helped shape my project ideas for this thesis, my world view and the scholarly standard I aspire to. However scientifically shoddy the work herein may be, in terms of personal emotion I take it as an homage to their work. I thank Professor Jason Locasale, who, with his breadth of knowledge introduced me to modeling projects with grounded biological relevance, and helped me become productive in graduate school; that dose of biological realism and the recognition of its importance stayed with me in my later independent theoretical studies. I thank Professor Giles Hooker, for his support in the last stage of the journey and a teaching opportunity. I thank Dr. Maria Terrell and Professor Andrew Clark for the other teaching opportunities; together, they constitute some of my most fulfilling time in graduate school. I thank Professors John Guckenheimer, Alex Vladimirov, Gregory Ezra and Reyer Sjamaar for generously allowing me to use some of their office hours for my research questions.

I thank Oleg Kogan, Eli Bogart, Brandon Barker, Yiping Wang and Colin Clement for discussions. I thank Mathav Kishore and Alex Fok for instilling a curiosity and appreciation for pure mathematics in me and answering my most naive questions on it. I thank Jason Hindes, whose intellectual intensity, concentration and precociousness I have not seen equalled among my peers leave a lasting inspiration in me. I thank Professors Chung-I Wu, Suhua Shi and Xionglei He from my undergraduate university for leading me to the road of science and their invaluable mentoring and encouragements at its beginning.

I thank S. Duan and R. Ghods for their love, support and the opportunities of growth. I thank Y. Luo, who, with his penetrating intellect and deeply descriptive and dispassionate approach to life, more than anyone else shook me out of my intellectual sloppiness, self-centeredness and naivety, which prompted me to start seeing and understanding this world as it is. I thank H.N. Wang, who, with his profound wisdom and kindness, helped me better understand myself and grow. I thank R. Ding, J. Skrovan and M. Hartsuyker for helping me grow. I thank H. Holmes, E. Ryan, A. King and S. Iams for inspirations. I thank C. Baker, F. Baker, S. Bhowmik, A. Chong, X. Cui, K. Farrell, B. Fichera, V. Ialenti, L. Jin, D. Lau, Y. Lodha, E. Mehrez, A. Poschadel, X. Serey, G. Spencer, L. Staley, L. Thorborgsson, H.Y. Wang, T. Wang and M. Zhang for friendships. I thank Q. Ma, L. Gang, M. Zhang and T. Betts for maintaining our friendships, notwithstanding time or space. I thank my family for their unfailing support. Lastly, I thank all those in this world doing honest work, academic or not, which gives strength to the idealism of a fellow human.

TABLE OF CONTENTS

Biographical Sketch	iii
Dedication	v
Acknowledgements	vi
Table of Contents	viii
List of Tables	x
List of Figures	xi
1 Introduction	1
1.1 Overview	1
1.2 Mathematical formalism	10
1.3 Rate laws	16
1.4 The parameter problem	25
1.5 Data in metabolism research	33
1.6 A list of toy models	37
2 Information Topology	43
2.1 An overview of information geometry	43
2.2 From information geometry to information topology	60
2.3 Model composition	63
2.4 Features of system topology	68
2.4.1 Structural nonidentifiability	69
2.4.2 Compositionality	71
2.4.3 Emergent boundaries	73
2.4.4 Combinatoriality	74
2.4.5 Symmetry	76
2.5 Information topology helps resolve structural nonidentifiability	80
2.5.1 Introduction	80
2.5.2 Some examples	82
2.5.3 Surjectivity analysis	93
2.5.4 Augmentation	100
2.5.5 Hasse-isocurve diagram and stitchwork	108
2.5.6 Summary	113
2.6 Construction of system topology	116
2.7 Appendix: Implementation	119

3	Information topology of kinetic models of metabolism	131
3.1	Introduction	131
3.2	Rate law topology	137
3.3	Kinetic behaviors	142
3.4	<i>JC</i> behaviors	149
3.4.1	Similarity	149
3.4.2	Modularity	151
3.4.3	Pathway module	163
3.4.4	Calvin cycle	167
4	Estimating relative changes of metabolic fluxes using kinetic flux profil-	179
	ing	
4.1	Introduction	180
4.2	Results/Discussions	186
4.2.1	Extending KFP to estimate relative flux changes	186
4.2.2	Missing data: effects and pitfalls	191
4.2.3	Modeling reversible reactions	199
4.2.4	Selecting measuring times	203
4.2.5	Applying rKFP to glucose deprivation data	206
4.3	Materials and methods	211
4.3.1	Computation	211
4.3.2	Experiments	213
	Bibliography	215

LIST OF TABLES

1.1	Common data types in metabolic studies. Capitalized symbols denote the space of variables (that is, the set of all possible values that the variables can take). Z : dependent variables; U : independent variables; X : (kinetic or steady-state) reactant concentrations; V : reaction rates; T : time; J : steady-state fluxes; E : enzyme concentrations. XC data: steady-state metabolite concentrations as a function of external metabolite concentrations; JC data: steady-state fluxes as a function of external metabolite concentrations; XE data: steady-state metabolite concentrations as a function of internal enzyme concentrations; JE data: steady-state fluxes as a function of internal enzyme concentrations.	36
2.1	A list of models and their model maps. Some models are formulated in the model-map form in the first place (those with “—” as their model maps indicating the same as the model), while others are formulated as something else that give rise to some model maps upon selection of some z , u , p and c . Universal gravitation: three different partitions of u , p and c are given, where in the last two cases G is considered constant making p an empty vector in the second case, and the third case corresponds to the common “inverse-square law” interpretation of the model; technically, M have different domains in all three cases and is hence different, despite the same formula. Predator-prey: an example where the model map does not admit an explicit formula and is represented computationally. Neural network: a single-layer artificial neural network model that has a logistic function with parameter k controlling the steepness as its activation function.	45
3.1	A partition of the set of behavioral spaces in Table 1.1 along importance and usefulness in metabolism.	133
4.1	Definitions of variables and their identifiabilities in rKFP	189

LIST OF FIGURES

- 1.1 Some examples of metabolic networks. In the diagrams, labels such as GLU represent metabolites (the boxed ones are considered external species outside the system that are sufficiently buffered and stay constant in concentration throughout the system dynamics) and edges between them represent biochemical reactions that transform some metabolites to others. (a) Glycolysis, the metabolic network that breaks down glucose to smaller building blocks and harvests its energy. (b) Calvin cycle, the metabolic network that assimilates CO_2 and synthesizes sugars using the energy from light in photosynthesis. Note the choice of words here: glycolysis "breaks down" while Calvin cycle "synthesizes"; one transforms big molecules to smaller ones and gets energy and the other transforms small molecules to bigger ones and uses energy. This contrast in direction and energy use between the two networks represents a general dichotomy in metabolic networks: those that "break down" and those that "synthesize".¹ Metabolic networks that exist in nature typically involves tens of metabolites and reactions like glycolysis and Calvin cycle; but they can also exist as a purely mental construct, usually in the form of "toy networks" that are smaller in size and more amenable for theoretical investigations. (c) shows an example of such "particle-in-a-box" type of metabolic network, which converts C_1 to C_2 in a sequence of two reaction transformations. 4
- 1.2 A Michaelis-Menten reaction scheme for an enzyme-catalyzed reaction $S \rightarrow P$. Appearing in almost every biochemistry textbook, the scheme describes the microscopic details underlying the net reaction $A \rightarrow P$ in terms of two elementary steps: the first step involving the formation (with rate constant k_f) and dissociation (with rate constant k_r) of enzyme-reactant complex C from an enzyme E and a substrate S , and the second step involving the transformation (with rate constant k_c) of the complex to an enzyme and a product P 17

1.3	A Michaelis-Menten reaction scheme for an enzyme-catalyzed reaction $A + B \leftrightarrow P + Q$. The reaction scheme involves two substrates and two products forming complexes in any order, and is often called "random bi-bi" in literature. The equilibrium constants, in this context more often called dissociation constants, of all the reversible enzyme-reactant formation-dissociation elementary steps, are labeled.	20
1.4	Histogram of measured Michaelis constant of GAPDH I for GAP in human. The data are taken from BRENDA, a manually-created repository of enzymatic data [95].	25
1.5	Woes of enzymatic data fitting. Taken out verbatim from [109], it shows how parameter estimates intimately depend on the fitting algorithm. The model is standard: $v = VS/(K_m + S)$, which "non-linearized fitting" directly uses, while all other fitting algorithms involve nonlinear transformations of it first, as shown in the figure. ENO 2: enolase 2.	28
1.6	Three toy networks that are used repeatedly in the next two chapters. (a) <code>path2</code> , a pathway of two single-substrate single-product reactions. (b) <code>path3</code> , a pathway of three single-substrate single-product reactions. (c) <code>cycle3</code> , a cycle of three reactions: $C_1 + X_1 \leftrightarrow 2X_2$; $X_2 + C_2 \leftrightarrow X_1$; $X_2 \leftrightarrow C_3$. All chemicals in boxes (C_1 , C_2 and C_3) represent external metabolites with fixed concentrations.	38
2.1	Parameter space, behavioral space and behavioral map for the Michaelis-Menten model $v = \frac{V_m x}{K_M + x}$. The left subfigure shows the parameter space P consisting of all possible (V_m, K_M) 's, of which six representatives are plotted for illustration (the six colored dots). The behavioral map F maps any point (V_m, K_M) in the parameter space to a point in the behavioral space which is a curve $v = v(x, V_m, K_m) = \frac{V_m x}{K_M + x}$. The right subfigure shows the behavioral space B consisting of all such curves, of which six are plotted corresponding to the points in parameter space with the same color (the six colored curves).	47

2.2	<p>One mathematical model can give rise to multiple behavioral maps. For a model \mathcal{M}, different choices of z and partitions of u, p and c give rise to different model maps $M : U \times P \rightarrow Z$, which lead to different behavioral maps $F : P \rightarrow U^Z$, each with its own parameter space P and behavioral space B. Of particular interest are those behavioral maps sharing the same parameter space as the domain (dashed box).</p>	49
2.3	<p>Parameter space, prediction space and prediction map for the Michaelis-Menten model $v = \frac{V_m x}{K_M + x}$. Imposing a sampling $X = \{x_1, x_2, x_3\}$ on the behavioral space of the model (Fig. 2.1) produces a prediction map f that maps any point (V_m, K_M) in the parameter space to a point in the prediction space which is a vector $\left(\frac{V_m x_1}{K_M + x_1}, \frac{V_m x_2}{K_M + x_2}, \frac{V_m x_3}{K_M + x_3} \right)$. The left subfigure shows the parameter space P in the same way as in Fig. 2.1. The right subfigure shows how the sampling keeps only a three points in each curve. Note that the right subfigure illustrates only the origin of the prediction space Y but not Y itself, as Y in this case is a three-dimensional Euclidean space (Fig. 2.6).</p>	50

- 2.4 A unifying view of model prediction, parameter estimation, experimental design, model selection and collective modeling. The functional perspective of information geometry casts a few common modeling tasks into some kind of "commutative diagrams". (a): The diagram describes the fundamental relations between parameters, behaviors and predictions: A model \mathcal{M} describes how some behaviors depends on parameters through the behavioral map F mapping from a parameter space P to a behavioral space B ; a sampling X of B produces the prediction space Y and the prediction map f that maps from P to Y ; given a $p \in P$, f generates model predictions $y = f(p)$. (b) Parameter estimation can be thought as the inverse problem of model prediction: $f^{-1} : Y \rightarrow P$ (technically f^{-1} is a kind of pseudoinverse, with a projection preceding functional inversion). (c): Experimental design can be thought as an optimization over B , illustrated here by two competing X and X' , which produce two different Y and Y' with their f and f' ; the task is to select an X from B that produces the optimal f by some criteria. ²(d): Model selection can be thought as comparing in the same B two different models, represented here by two F and F' , or when an X and an X' are chosen, comparing the f and f' in the same Y . (e): Collective modeling, a term we coined in Section 1.4 and emphasizes using collective data to predict collective behaviors without focusing on parameters, can be thought as a map from a B to another B' , or upon choosing an X and an X' , from a Y to another Y' . 53
- 2.5 Homotopy morphs parameter space to model manifold for the `sum2exp` model. The top plot with $\lambda = 0$ shows the parameter space, with its boundaries schematically labeled (the colored edges of the square; the parameter space has its boundaries at infinity, so what is shown here can be thought as a "processed" parameter space that is compactified using, for example, the arctan map in Footnote 11). The lower rightmost plot with $\lambda = 1$ shows the model manifold, with its boundaries corresponding to those of the parameter space and, because of symmetry (Section 2.4.5), part of the parameter space interior as well (the red curve). The rest of the plots represent intermediate steps in between with $0 < \lambda < 1$. The black dot on the parameter space is mapped to a black dot on the manifold along the process. 56

- 2.6 Geometry of parameter estimation and model reduction of the Michaelis-Menten model. (a) Using the sampling $X = \{x_1, x_2, x_3\}$ illustrated in Fig. 2.3, the prediction space Y is a three-dimensional Euclidean space with axes corresponding to $v(x_1)$, $v(x_2)$ and $v(x_3)$, and the model manifold \mathcal{M} is a two-dimensional surface embedded in Y (the blue surface). Geometrically, data with measurement noise corresponds to a point in Y with some deviation from the manifold (the red dot), and parameter estimation using the least-square method corresponds to finding the point on the manifold with the shortest distance to the data vector (the black line segment projects the data vector onto the manifold and finds the closest point). Geometrically, model reduction is translated by MBAM to traveling on the manifold towards boundaries (the magenta and green dashed rays) via geodesic motion (the magenta and green curves). (b) Parameter trajectories along the geodesics, which reveal parameter limits at the boundaries. One geodesic (the magenta curve) hits a boundary (the magenta dashed ray) where $K_M \rightarrow 0$ while $V_m \sim O(1)$ (the left half of the parameter trajectories; note that the parameters have been logarithmically transformed), meaning that the model $v = \frac{V_m x}{K_M + x}$ becomes $\lim_{K_M \rightarrow 0} v = V_m$; similarly, geodesic along the opposite direction (the green curve) hits the other boundary (the green dashed ray) where $V_m \rightarrow \infty$ and $K_M \rightarrow \infty$ while $k = V_m/K_M \sim O(1)$, meaning that the model becomes $\lim_{K_M \rightarrow \infty} \frac{k K_M x}{K_M + x} = k x$. In this way, MBAM finds the linear and saturate regimes semi-automatically, the two commonly-used simplifications of the Michaelis-Menten rate law. 59
- 2.7 Hasse diagram of the Michaelis-Menten model. Fig. 2.6 shows that for the Michaelis-Menten model $v = \frac{V_m x}{K_M + x}$, there are two boundaries, $v = V_m$ and $v = k x$, both of which geometrically are rays emanating from the origin to infinity. Such information can be intuitively encoded in a hierarchical graph called Hasse diagram (dashed arrow and boxes represent unbounded singular limits, that is, infinity). High-dimensional model manifolds are hard to visualize, but their Hasse diagrams encoding the information of their boundary structures can be readily constructed (see Figs. 3.2, 3.4 and 3.9 for examples). 63

2.8	Hasse diagram of the topology of rate law <code>mm11</code> . The names are given in red which contain information about the atomic reductions involved: for example, <code>f/r</code> means forward/reverse unidirectionalization, <code>A_{lin}</code> means linearization of substrate A, and <code>AP_{sat}</code> means saturation of both A and P. Functional forms are given in black and unbounded singular limits are boxed in dashed lines.	75
2.9	Singular value spectra of a few behavioral maps from metabolic modeling. All behavioral maps have structural nonidentifiability, manifested by the gap in the spectra, except <code>(path2mah, XT)</code> . <code>(path2mmh, XT)</code> has some of its singular values close not to "machine epsilon", but to the preset numerical tolerance in integration (10^{-12}).	81
2.10	An isocurve with its singular vector signature of <code>(path2mah, XC)</code> with $k_1 = 1$. The time information along the x-axis comes from the fact that we trace out the isocurve using geodesic motion with a constant speed in parameter space.	90
2.11	Applying sampling-fitting to the $(k_1 = 1, b_4 = 0)$ resolution of <code>(path2mm, XC)</code> . In a sample of 100 points, the blue histogram (a) shows the cost distribution when the fitting is done in log parameters, thereby enforcing positivity; the green histogram (b) shows the cost distribution when the fitting is done in bare parameters without log -transformation, thereby allowing parameters to take on negative values; the red histogram (c) shows the distribution of the fitted value of parameter b_2 , which, upon inspection, is found to be the only parameter ever taking negative values in the fit.	94
2.12	Enumerating all four classes of isocurves for <code>(path2mm_{k₁=1}, XC)</code>	98
2.13	"Connectome" of isocurve classes overlaid on Hasse diagram. The colors and line styles of the two subplots match each other.	99
2.14	Manifold resurrection for <code>(path2mah, XC)</code> with XT as the augmenting behavioral space. <code>(path2mah, XC)</code> has $(n, r, d) = (2, 1, 1)$, so its manifold \mathcal{M} is a collapsed curve represented by the magenta line segment. Using XT as the augmenting behavioral space resurrects the manifold to $\tilde{\mathcal{M}}$ represented by the blue trapezoid. The red and green curves represent the currying reparametrizations of <code>(path2mah, XC)</code> with k_1 and k_2 set to 1 respectively. The black curve represents an isocurve on $\tilde{\mathcal{M}}$, with its starting point and its projection on \mathcal{M} represented by the two solid dots.	103

2.15	Illustration of a resurrected manifold $\tilde{\mathcal{M}}$ without a disjoint covering set of identifying boundaries for \mathcal{M}	108
2.16	Manifold resurrection and Hasse-isocurve diagram for a hypothetical model with $d = 2$. (a) Resurrection of a collapsed line segment to a tetrahedron. The solid black line segment along the y_1 axis represents the collapse manifold \mathcal{M} , and the tetrahedron represents the resurrected manifold $\tilde{\mathcal{M}}$. (b) The associated Hasse-isocurve diagram. To get a covering set of identifying boundaries, one needs to travel two ranks down, first from the interior of the tetrahedron $ABCD$ to its faces, then from the faces to the edges. Enumerating isocurves in $ABCD$ gives the isocurves and the faces they connect, as shown in the upper set of arching edges connecting nodes of corank one; the colored ABD and ACD nodes represent the chosen faces for descension which form a covering set of \mathcal{M} . Further enumerating isocurves in ABD and ACD gives the lower set of arching edges connecting nodes of corank two (AD is shared by ABD and ACD , hence in a different color), from which a covering set of identifying boundaries can be found: any of $\{BD, CD\}$, $\{AB, AC\}$ or $\{BC\}$ works.	110
2.17	A simple example of stitchwork. Using $\{BD, CD\}$ to represent \mathcal{M} in our tetrahedron example, one need to remove the boundaries shared between BD and CD , that is D . Hence the resulting topology is a line segment, like \mathcal{M}	113

2.18	Relationships between the core classes in the package <code>infotopo</code> . The central class is <code>Predict</code> (ie, a prediction map f), derived from a <code>Model</code> object (specifying a mathematical model \mathcal{M}) and an <code>Experiments</code> object (specifying a behavioral space B and a sampling X). With a <code>Predict</code> object, one can perform a number of standard modeling tasks: generating predictions, fitting it to data (by creating a <code>Residual</code> object and computing a <code>Fit</code> object), and sampling the parameter space using different combinations of priors and posteriors (represented by <code>Ensemble</code> objects). A <code>Predict</code> object is also key to some information geometric and information topological analysis: it can generate a <code>Geodesic</code> object, which can be integrated to form a <code>Trajectory</code> object and the limiting behaviors can be examined in a <code>Limit</code> object (only partially implemented so far; hence the dashed box) constructed from an integrated geodesic; sampling many limiting behaviors starts to accumulate global and topological information, stored in a <code>HasseDiagram</code> object.	120
2.19	The structure and functions of the codes for reaction network models.	121
3.1	Information flows between a directed pair of behavioral spaces. Judgements on importance and usefulness of behavioral spaces impose directionality on the edges between parameter space P and behavioral spaces B	134
3.2	Hasse diagrams of a Michaelis-Menten rate law and the same one with the Haldane constraint. (a) The diagram of Michaelis-Menten rate law, a re-plot of the one in Fig. 2.8 using VK parametrization to facilitate interpretation. (b) The diagram of Michaelis-Menten-Haldane rate law.	140
3.3	Hasse diagram of the same Michaelis-Menten rate law in Fig. 3.2 with $V_f = 1$	141

3.4	Partial construction of the topology for model $(\text{path2mm}, \text{XT})$, the kinetic behaviors of a two-reaction pathway with Michaelis-Menten rate laws. The Magenta curves correspond to the isocurve edges in a Hasse-isocurve diagram, and encode the resolution of SN by MBIM. All the red nodes represent the component product topology upon selecting an identifying set of boundaries in MBIM. The black node represents the shared boundary between the two disjoint identifying boundaries which corresponds to system manifold interior and is removed by the stitchwork step in MBIM. The two pairs of blue and green nodes represent the two emergent boundaries discussed as examples in the main text: the upper pair represents the emergent vector field boundary in Example 1, and the lower pair represents the emergent trajectory boundary in Example 2. Some nodes are stacked in the middle to highlight the bilateral symmetry of the model.	148
3.5	Modularization of a three-reaction pathway path3 . The first two reactions in the red dashed box form a module, and can be conceptually thought as a single reaction \tilde{R}_{12} , thereby effectively coarse-graining the path3 (top) to a two-reaction pathway path2 (bottom).	152
3.6	Modularization of an example network. (a) The example network. (b) Applying module-searching algorithm to the example network yields the six modules.	156
3.7	The steady state reached through integration may change upon modularizing path3 to path2 in Fig. 3.5. The dynamics of path2 in its reduced state space (the X_2 line on the left) is equivalent to the projection (the horizontal arrows pointing left) of the dynamics on the slow manifold (the black curve) of the original network in the full state space (the X_1 - X_2 plane on the right), defined by the equations that give rise to the module rate law, onto the reduced state space. As long as the initial condition and its projection onto the slow manifold (the two blue dots in the X_1 - X_2 plane) belong to different basins of attraction, they would reach different steady states (the green and red dots in the X_1 - X_2 plane), thereby having path3 and path2 reach different steady states.	158
3.8	A collection of common modules of metabolic networks.	162

3.9	The topology of a two-reaction pathway with Michaelis-Menten-Haldane rate laws describing <i>JC</i> behaviors. The model has a bilateral symmetry which allows us to explore only half of its topology (colored nodes). The green nodes represent models with a 0 rate law and thereby describe inactive pathways. The magenta and red nodes represent models with SN and are resolved by MBIM. The blue nodes represent emergent boundaries with one reaction set to equilibrium. The cyan nodes represent component boundaries with various rate law reductions. Notation scheme of rate law reductions on the nodes: <i>A</i> for substrate A, <i>P</i> for product P, subscript \bullet_l for the reactant(s) in the linear regime and subscript \bullet_s for the reactant(s) saturating (Fig. 3.2).	165
3.10	The core structure of Calvin cycle.	168
3.11	The size distribution of all modules in the Calvin cycle in Fig. 3.10.	169
3.12	The regeneration module of Calvin cycle. The magenta box contains all the species and reactions in the module, effectively regenerating RuBP from GAP.	171
3.13	The energization module of Calvin cycle. The orange box contains all the species and reactions in the module, effectively harvesting the energy from light to energize the cycle. The magenta reaction represents the coarse-grained regeneration module.	172
3.14	(left) The replenishment module of Calvin cycle. The red box contains all the species and reactions in the module, effectively converting <i>PGA</i> all the way to <i>RuBP</i> supported by the energy provided by light. The magenta and orange reactions represent the coarse-grained regeneration and energization modules, respectively. (right) Calvin cycle in fixation, replenishment and transport modules.	173
3.15	Reaction condensation diagram for our modularization of Calvin cycle. The modules are in wide rectangular boxes, and we stop at the colored ones, without further adopting the modularizations represented by the dashed boxes.	174

4.1	Understanding KFP and rKFP. (a) A schematic diagram of KFP applied to a toy metabolic network. At $t = 0$, the system is switched from a ^{12}C -labeled environment to ^{13}C -labeled one, and A^* is measured at a few time points thereafter. (b) For a given trajectory of $A^*(t)$ (the black solid curve), the three time regimes (linear, mixed and constant) are marked and three measurements are made (two in the mixed regime and one in the constant). Normalizing it gives $\hat{A}^*(t)$ between 0 and 1 (the black dashed curve), parametrized by a single parameter μ , which can be estimated by comparing the normalized measurements to \hat{A}^* 's of different μ 's (the red and blue dashed curves). (c) A schematic diagram of rKFP applied to the same network in (a). Relative quantitation is performed on A^* in two conditions (with subscripts x and y respectively) with the goal of estimating $r_J = J_y/J_x$. (d) The ratio in μ between $\hat{a}_x^*(t)$ and $\hat{a}_y^*(t)$ is r_J/r_A (Eq. 4.6), and since μ 's and r_A are identifiable from relative quantitation, so is r_J	183
4.2	Metabolite removal in KFP. (a) A schematic diagram of getting the reduced model through metabolite removal in KFP. Dashed squares represent metabolites removed in the reduced model; thick dark arrow represents reduction; \tilde{J} represents the estimated J (potentially biased). (b) The estimation results for the three options. The solid curves represent \tilde{J} , the dashed curves represent the cost of fitting (normalized by the number of data points to be comparable across options), and three colors represent the three options. Parameter values used for generating the simulated data: $A_1 = A_3 = J = 1$ (overall patterns independent of the choice here). (c) The estimation results in (b) in terms of the three summary statistics.	193
4.3	Metabolite removal in rKFP. (a) A schematic diagram of getting the reduced model through metabolite removal in rKFP. The same figure scheme as in Figure 4.2a applies here. Parameter values used in generating the simulated data: $A_{1x} = A_{3x} = J_x = 1$, $A_{1y} = A_{3y} = 3$, and $J_y = 2$. (b) Dependence of bias on the pool size of the missing metabolite in two conditions. (c) Dependence of error ratio on the pool size of the missing metabolite in two conditions.	197

4.4	Modeling reversible reactions in KFP and rKFP. (a) A schematic diagram of the toy system considered here; for rKFP two copies of each network are similarly made as in Figure 4.3a. Parameter values used in generating the simulated data: $A_1 = A_2 = A_{1x} = A_{2x} = J = J_x = 1$, $A_{1y} = A_{2y} = 3$, and $J_y = 2$. (b) Dependence of the summary statistics on ΔG in KFP. (c) Dependence of the bias of r_J on ΔG of the two conditions in rKFP. The red solid diagonal line corresponds to $\Delta G_x = \Delta G_y$ where there is no bias. The red dashed curves correspond to a five-fold difference in the relative pool size changes between the substrate and product, a range we observe in our data.	202
4.5	Analysis of experimental data.	208

CHAPTER 1

INTRODUCTION

This thesis is about studying mathematical models of metabolism using the approaches of information geometry and information topology. As such, in this chapter we provide a gentle introduction to metabolism and some general relevant biology, and their mathematical models.

1.1 Overview

What is metabolism? To answer this question, let us position ourselves more broadly and ask: *what is life?* Here it is unnecessary for us to follow the tradition and attempt to distill down to a watertight definition, but suffice it to acknowledge three key features that stand out among those exhibited by all extant living organisms:

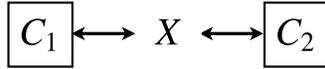
- They grow and reproduce;
- They maintain internal homeostasis;¹
- They respond to external signals.

Growth makes an organism larger, and reproduction multiplies an organism: they require both building blocks and energy to assemble the building blocks into

¹A biology jargon that roughly means internal stability.

new organisms or part of them (collectively termed biomass). Maintaining homeostasis and responding to signals also require energy. *Metabolism is the system in an organism that transforms environmental intake (such as food, light and CO₂) into biomass to achieve growth and energy to support other functions*².

²The concept of metabolism has been generalized beyond the organic realm; for example, *urban metabolism* may be thought as "the system in a city that transforms environmental intake into city-mass to achieve growth and energy to support other urban functions."



(c) A toy network

Figure 1.1: Some examples of metabolic networks. In the diagrams, labels such as GLU represent metabolites (the boxed ones are considered external species outside the system that are sufficiently buffered and stay constant in concentration throughout the system dynamics) and edges between them represent biochemical reactions that transform some metabolites to others. (a) Glycolysis, the metabolic network that breaks down glucose to smaller building blocks and harvests its energy. (b) Calvin cycle, the metabolic network that assimilates CO_2 and synthesizes sugars using the energy from light in photosynthesis. Note the choice of words here: glycolysis “breaks down” while Calvin cycle “synthesizes”; one transforms big molecules to smaller ones and gets energy and the other transforms small molecules to bigger ones and uses energy. This contrast in direction and energy use between the two networks represents a general dichotomy in metabolic networks: those that “break down” and those that “synthesize”.⁴ Metabolic networks that exist in nature typically involves tens of metabolites and reactions like glycolysis and Calvin cycle; but they can also exist as a purely mental construct, usually in the form of “toy networks” that are smaller in size and more amenable for theoretical investigations. (c) shows an example of such “particle-in-a-box” type of metabolic network, which converts C_1 to C_2 in a sequence of two reaction transformations.

Hence, if we study life as a system rather than its tiniest constituents such as genes and proteins – an undertaking assumed by the field of *systems biology* – a natural demarcation can be drawn between the system that performs metabolism

⁴They are called *catabolism* and *anabolism* respectively; together they make up metabolism.

and those that perform "other functions" such as maintaining homeostasis and responding to signals. In fact, the major bulk of systems biology falls along this demarcation, as a typical study claims to tackle one of three classes of networks: metabolic networks (that carry out metabolism), and signal transduction networks and gene regulatory networks (that maintain homeostasis and respond to signals).⁵

Some remarks are in order. First, we use the words "system" and "networks" somewhat interchangeably in the last paragraph, for the system here takes a form of a network consisting of a discrete set of *components*. Not all systems studied in science take such a form⁶, but when they indeed do, it may be easier to think about⁷. We will revisit this topic in Section 2.3. Second, we use the word "system" in singular but "networks" in plural, implying, for example, a metabolic system consists of many metabolic networks. This is possible because of *modularity* ([1, 14]): one can break down a system into many subsystems, and reason each of its subsystems independently as a system in itself.⁸ As one of greatest cognitive toolkits available for reasoning about systems, this concept will be revisited in Section 3.4. For these two facts, the two words, singular or plural, will be used interchangeably hereafter.

⁵The two classes are divided not by their biological functions, but by how the functions are achieved: gene regulatory networks involve nucleus, while signal transduction networks do not; this distinction entails various downstream differences between the two classes, from system properties (such as timescale) and the approach needed to study them.

⁶For example, much of modern physics establishes its mental image as some collection of *fields*, some amorphous and intangible entities that permeate throughout the space

⁷But not necessarily easier to do mathematics about, hence the other mental pictures.

⁸This nested structure of modularity spawns a related concept of *hierarchy*.

For the three classes of networks studied in systems biology, their fundamentally different biological functions as discussed above lead to some fundamentally different definitions of their *phenotypes*, which is worth emphasizing here. The first distinction is steady state vs. dynamics: steady-state performances of metabolic networks matter a great deal more than their transient behaviors that precede the steady state. The second distinction is flux vs. concentration: flux rather than concentration takes the central role in defining metabolic performances. Both can be traced back to the functions of metabolism: as long as the infrastructure and supply chain (that is, metabolism) supports other functions in an organism, why should it be obsessed with the little hick-ups that occasionally come up (that is, transients) or the details of how the feat is actually accomplished (that is concentration)? Just like a professor whose main concern is a student's steady-state productivity and cares much less whether he did a subpar presentation at the group meeting three months ago or whether he uses Python or Matlab codes to generate the plot.

The opposite of both distinctions can be said for the other two classes of networks. There, the very information that an organism responds to and conveys among different systems is often encoded in transients, usually in concentration. *E. coli* famously executes a biased random walk when it senses a temporal change in surrounding food concentration, accomplished through an intricate network that relays the signal around, also in a temporally changing fashion. The two classes of networks are no longer the professor's students, but his colleagues and friends in funding agencies with whom he restlessly discusses about the latest

research or chatters about the next hottest topics.

This distinction is not absolute. An important exception lies at the interface between metabolic networks and the other two classes, where often it is the information in the steady-state metabolite concentrations that is picked up and processed. Response to low blood glucose is triggered when it drops to a certain level, not when it drops at a certain rate.

There are also practical reasons to such a distinction of focus: metabolism is fast and metabolites are small. If it were slow, experimenters could probe its temporal profiles more easily.⁹ If metabolites were bigger, experimenters could potentially use the same trick as they do to proteins, attaching fluorescent tags to the metabolites and watching the colors wax and wane as metabolite concentrations rise and fall.

For these reasons, *metabolic networks has steady-state flux as its phenotype*, unlike the two classes. It is worth keeping this firmly in mind. Perhaps too often, a similar mathematical formalism shared by a metabolic network and a gene regulatory network (Section 1.2) spawns similar quests.

We conclude this section with a bit of history. To study any system, a natural starting point is to figure out its "part list", something metabolic networks

⁹My experimental colleagues have told me fascinating stories about it takes a student years of practice and some elusive talent to join the 0.1 Second Club, consisting of people who can hold a live plant tissue sample on one hand, precariously hold a container of liquid nitrogen on the other, and manage the cutting-tissue-pouring-liquid-nitrogen stunt within 0.1 second, a standard that the field have somehow come to tacitly agree upon.

have been ahead of the other two. The “part list” of important metabolic networks have long been worked out: for example, citric acid cycle (1930s), glycolysis (1940s), and Calvin cycle (1950s).¹⁰ Studies of network components, *ie* individual metabolic reactions, started even earlier. By late 19th century, around the time when *vitalism* retreated, scientists started to describe biochemical processes in physical terms, and decades of effort culminated in 1913 when Michaelis and Menten published their celebrated now eponymous rate law, setting up a paradigm that we have not left since and a line of research (that is, *enzyme kinetics*) that remained remarkably active for decades. By 1950s, with the wiring diagram of major metabolic networks worked out, two new major efforts joined in: building what we now call kinetic models of metabolic networks (eg, [23]) and developing theories that help understand the models (notably metabolic control analysis [48, 51]). Research on metabolic networks was at its heyday.¹¹

With all these achievements and exciting discoveries emerging in other fields such as molecular biology and gene regulation in the second half of the century, momentum in metabolic network research began to shift away towards later decades. By 1990s and 2000s, there was still some work of the sort but it was often done in the context of metabolic engineering and papers appeared in journals bearing that name. Around the turn of the century, with the completion of Human Genome Project, systems biology became fashionable. Among all the system models that then started to blossom and occupy people’s discourse, the tradi-

¹⁰The two cycles were Nobel-prize winning discoveries.

¹¹The Ngram dynamics of “*enzyme kinetics*” tells a story.

tional type of metabolic network models that have been studied for decades was not one of them; metabolism in general was considered “well-understood”, if not downright “boring”, and did not quite catch the high-speed train of systems biology that was running at a frantic pace (save the exception of constraint-based modeling). In retrospect, this is understandable: people were too busy trying to understand how life does its all sorts of wizardry such as adaptation and differentiation to pay attention to the menial housekeeping that is supposed the role of metabolism. By early 2010s, some universities like Harvard have removed metabolism out of their undergraduate curricula [89].

Tides began to turn again with the discoveries in recent years that have found metabolism fundamental to epigenetics [52], relevant in development and immunity [84], implicated in pathogenesis [112, 111], intimate with nutrition [117] and critical for some grand engineering projects aiming to further crop yield and alleviate climate change [21]. Therefore, there has been a recent resurgence of interest not only in metabolism in general, but also in metabolic network modeling. There is an important consequence to this *punctuated* history of metabolism network modeling: old wisdom did not fully join new effort, and much modern modeling work was done out of apparent ignorance of the fundamentals and results that previous generations of scientists have distilled and developed. At the same time, some old nagging problems are made more pronounced by the new interest, requiring us not only to grasp the classical, but to think beyond it. The rest of this chapter is devoted to an account of this whole modeling landscape, from Michaelis and Menten to metabolome and manifold. We will start from the basic

mathematical formalism (Section 1.2), journey through the subtleties, highlight the fundamentals (Section 1.3), elaborate the challenge, describe the proposed solutions (Section 1.4), and pave way for the new developments detailed in later chapters.

1.2 Mathematical formalism

If you open a modern condensed matter physics journal and randomly pick a paper, you may well see its mathematics start with an equation defining a term called "Hamiltonian", whose form does not look all that different from the Hamiltonian-defining equations in other papers. Likewise, a typical statistics paper these days premises its mathematical discourse with an equation specifying what is called the "model" that explains how data is assumed generated. On the other hand, take a glance at the preprints submitted to arxiv under the category of "q-bio" (that is, quantitative biology) and you would have the impression of witnessing a menagerie of mathematical formalisms, from boolean networks to stochastic partial differential equation with delays.

This is the case for good reasons. With all its complexities, life is a huge elephant after all and systematic attempt to mathematically understand it as a system is a relatively recent phenomenon. True, all fields are fighting some giant elephants, but some elephants are more giant than others. To tame the wild beast, people apply all sorts of gizmos at it: petri net to temper its torso, lasso to subdue

its legs, all the while going only as far as they see fit so that it can still wiggle its trunk.

For metabolism, things are rosier. There is such a thing as “foundational formalism” to speak of, as shown below:

$$\frac{dx}{dt} = Nv(x, p) \tag{1.1}$$

where t is time, x is a vector of metabolite concentrations, v is a vector of reaction rates, p is a vector of parameters, and N is a matrix containing what are called *stoichiometric coefficients* of the reactions. ¹²

From this equation flows out various derived formalisms, including *kinetic modeling*, *constraint-based modeling* and *structural kinetic modeling*. The work in this thesis resides within the formalism of kinetic modeling, and we shall continue with its introduction in this section and the next. In section 1.4 we will briefly describe constraint-based modeling, the other major formalism in metabolism modeling, where an explanation of the choice on kinetic modeling is also given.

The mathematical nature of Eq. (1.1) is a system of ordinary differential equations (ODEs), an “ordinary” fact that is somewhat telltale. First, it is not a partial

¹²The following equation may help unpack Eq. 1.1 a bit: $\frac{dx_i}{dt} = \sum_j \frac{\partial x_i}{\partial R_j} \frac{dR_j}{dt}$, where R_j represents reaction j . It means, in a chain-rule fashion, that the rate of change for x_i depends on the firing rate of a reaction ($\frac{dR_j}{dt}$, or v_j) and how many x_i gets created or consumed each time R_j fires ($\frac{\partial x_i}{\partial R_j}$, or the stoichiometry coefficient N_{ij}), summed over all reactions.

differential equation. What it means is that the concentration of a metabolite is assumed the same at different spatial locations. Second, it is not stochastic (like a stochastic process or a system of stochastic differential equations). What it means is that the number of chemicals participating in a reaction per unit time is assumed large enough to have the randomness averaged out, making deterministic dynamics a good approximation. Third, it is not an ordinary difference equation, that is, its state space is continuous rather than discrete. What it means is that, again, the number of chemicals involved is large, so that we can think of a curve of its dynamics as a continuous one even though keeping zooming in would eventually uncover the tiny step jumps.¹³ These assumptions are usually considered valid for metabolism, as metabolites, which are relatively small (Section 1.1), diffuse fast and come in huge quantities (small molecules are cheap to make and a lot of them are needed to make any difference).

When any of the assumptions is considered dubious, Eq. (1.1) ceases being the default starting point. This is one of the main reasons why at present systems biology as a whole harbors a remarkable heterogeneity of formalisms: aside from other myriad complicating factors, DNA, RNA and protein molecules, the central objects of study in the other two classes of networks, may be too scarce, too bulky or too much under active control¹⁴ to justify the assumptions. But often enough, the assumptions are adopted for the two classes and the formalism is shared.

¹³Physicists may call the three assumptions *mean field approximation*, *thermodynamic limit* and *continuum limit* respectively.

¹⁴They may be "sequestered" through some "cellular scaffold" to increase local concentration.

However, seeing Eq. 1.1 merely as a system of ODEs would obscure some other important perspectives on a metabolic network. The ODE lens casts a metabolic network *mathematically* to a dynamical system, where v does not mean much unless multiplied by N , which now forms a vector field, and x is just a life-less point flowing in its state space along the vector field; inquiries along this line typically involve questions like "how many steady states are there?", "are they stable?", etc. However, as one of the prototypical far-from-equilibrium systems, metabolic networks also invite a *physical* interpretation, with x representing chemical potentials and v transducing free energy; important questions along this line include "is there any variational principle describing the steady state?", "what is the entropy production rate?", etc. Also, there is an emerging *economic* perspective on metabolism, which holds great promise in deepening our understanding of metabolism, for its view of metabolism under cellular regulation as a planned economy operating on limited enzyme resources with some type of optimization as the goal [60] formalizes much of the underlying biology; questions of this type often have the flavor of "what is the optimal metabolic strategy for optimizing growth rate, to increase the efficiency of nutrient uptake or yield of nutrient use?" and "how does the cell allocate its enzyme resources across the network to maximize its flux?" Lastly, there is always the *biological* perspective, in which case every term in the equation is laden with biological meaning; questions, then, are of a biological nature: "how is the system going to adapt to a new environmental CO₂ concentration?", "which reaction is the key regulation target?", etc. Needless to say, a solid understanding of metabolic networks requires all four perspectives.

Yet, often a focus of perspective is taken to the exclusion of others, especially the physical and economic ones.¹⁵

Continuing with our discussion of Eq. 1.1, a few steps are involved in translating a general reaction network to it:

1. Determine the system boundary: what reactants are considered internal to the system and what external.
2. Determine the stoichiometry of the reactions inside the system, that is N .
3. Determine the rate laws of the reactions, that is, v .
4. Determine the parameters of the rate laws, that is, p .
5. Determine the initial concentrations of reactants $x(0)$.

Let us see an example. For the toy network (c) in Fig. 1.1, the system boundary has already been decided: C_1 and C_2 lie outside the system with constant concentrations, and a single metabolite X lies inside. The first reaction consumes one C_1 and produces an X , while the second one consumes one X and produces a C_2 , making $N = \begin{pmatrix} 1 & -1 \end{pmatrix}$ with rows corresponding to metabolites and columns reactions. For the rate laws v , we assume they have the simple forms: $v_1 = k_1(C_1 - X)$ and $v_2 = k_2(X - C_2)$.¹⁶ Lastly, we choose parameter values $k_1 = 1$, $k_2 = 2$ together with $C_1 = 2$ and $C_2 = 1$, and initial condition $X(0) = 0$. Plugging all these modeling

¹⁵As an example of a lack of the physical perspective, often in metabolic models, reactions are set to be irreversible and unphysical predictions are thus generated.

¹⁶it is customary of the field to use the same symbol for both a chemical and its concentration.

choices into Eq. 1.1 we have $\frac{dX}{dt} = \begin{pmatrix} 1 & -1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = v_1 - v_2 = k_1(C_1 - X) - k_2(X - C_2) = 1(2 - X) - 2(X - 1) = 4 - 3X$ with $X(0) = 0$, which describes the dynamics of the system and can be used for solving steady states.

For any metabolic model, with the steps in the list completed and Eq. 1.1 at hand as shown for the toy network above, one can go about using the model to make predictions. For example, given a glycolysis model in the form of Eq. 1.1, it can tell you what the steady-state flux of glycolysis is, how it will change if external glucose concentration increases by 20%, which reactions in the system are thought to be slowing things down and a boost on them would make a big difference, etc.

This whole business of building models and making predictions sounds all very nice, except that two glaringly controversial steps on the list cloud any predictions churned out this way: people often disagree on the rate laws v , and rarely agree on the parameters p .¹⁷

The next two sections are devoted to a description of the controversies, first on rate laws, then on parameters.

¹⁷*N* enjoys a rare kind of consensus in metabolism modeling.

1.3 Rate laws

Let us start by defining rate laws¹⁸: the *rate law* of a reaction is a function that describes how the rate of the reaction depends on the concentrations of its reactants.

The second point we note is the generality of the notion. Viewed generally, this "amount-to-rate" type of notion appears in a variety of contexts:

- In physics, we have the Stefan-Boltzmann law that describes how the rate of energy release depends on an object's energy level;
- In history and economics, people study how the rate of commerce and exchange depend on the population size of a settlement;
- In sociology, people study how one's spending rate depends on his wealth.

The third point we note is that for a reaction involving multiple elementary steps, its rate law is not well-defined. To see this, let us look at the famous Michaelis-Menten reaction scheme.

Fig. 1.2 describes one particular reaction scheme supposed for an enzyme-catalyzed reaction $A \rightarrow P$: underlying the apparent net reaction $A \rightarrow P$, there are in fact a few elementary steps, involving the formation and transformation

¹⁸It is sometimes written as "rate-law" and is also known as "kinetic law" or "rate equation". The word "law" in the name can seem a bit of a misnomer, since the apparent lack of empirical validation or consensus on rate laws in systems biology modeling and their often ad-hoc derivations (see below) do not suggest a status as solid as, for example, Ohm's law. In this sense, rate laws are as laws as rate constants are constants (Section 1.4).

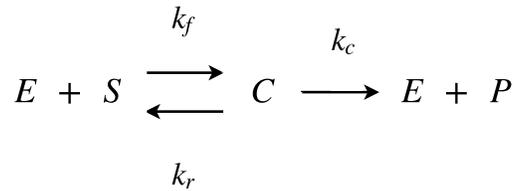


Figure 1.2: A Michaelis-Menten reaction scheme for an enzyme-catalyzed reaction $S \rightarrow P$. Appearing in almost every biochemistry textbook, the scheme describes the microscopic details underlying the net reaction $A \rightarrow P$ in terms of two elementary steps: the first step involving the formation (with rate constant k_f) and dissociation (with rate constant k_r) of enzyme-reactant complex C from an enzyme E and a substrate S , and the second step involving the transformation (with rate constant k_c) of the complex to an enzyme and a product P .

of the enzyme-reactant complex C ¹⁹ which are the microscopic details invisible from the net reaction. Typically a set of equations in the form of Eq. 1.1 describing the dynamics of chemical species involved in the reaction scheme is then written down.

$$\begin{aligned}
 \frac{dS}{dt} &= -k_f ES \\
 \frac{dE}{dt} &= -k_f ES + (k_r + k_c)C \\
 \frac{dC}{dt} &= k_f ES - (k_r + k_c)C \\
 \frac{dP}{dt} &= k_c C
 \end{aligned}$$

¹⁹We temporarily use C to denote enzyme-reactant complex in this section, and in the rest of the thesis use it to denote environmental, constant-concentration reactants (eg, Section 1.5).

The "reaction rate" v of reaction $A \rightarrow P$ is conventionally defined as $v = \frac{dP}{dt}$. To have a well-defined rate law, we need to express it as a function of reactant concentrations: $v = v(S, P)$. According to the equations, $v = k_c C$; yet, the equations also tell us that all S , C and P are dynamic variables and they are functions of the initial conditions and time t , rather than C is a function of S and P . We do not have a well-defined rate law here.

In this type of situations in biochemistry, it is common to make one of two kinds of assumptions, which allow for a valid notion of rate law for the net reactions despite the multiple elementary steps. First, one may assume that certain elementary steps are much faster than others: in fact, one has to assume that all but one reaction are so much faster than the slowest reaction that the dynamics of slowest reaction dominates at the time-scale of our observation. If the fast reactions are modeled as reversible, then we are essentially assuming them to have reached equilibrium, and implementing this separation-of-time-scale assumption leads to *quasi-equilibrium approximations* (QEA). For example, Michaelis and Menten assumed the first reaction of the formation and dissociation of C in Fig. 1.2 to be at equilibrium in deriving their eponymous rate law [72]: $\frac{SE}{C} = K_E$, and now C becomes a function of S . Second, one may assume that certain chemical species are much more abundant than others, and this separation-of-concentration-scale assumption, in a subtle way, translates to certain species being approximately stationary throughout the reaction dynamics, which leads to *quasi-steady-state approximations* (QSSA). For example, Briggs and Haldane [15] observed that, for the reaction scheme in Fig. 1.2, if the total enzyme concentra-

tion E_0 is assumed to be much smaller than the total substrate concentration S_0 , then the rate of change for complex C , $\frac{dC}{dt}$, would be much smaller than that of S and P ; that is, for our observation scale of $v = \frac{dP}{dt}$, $\frac{dC}{dt}$ can be taken to be zero: $\frac{dC}{dt} = k_f ES - (k_r + k_c)C = 0$ and now C again becomes a function of S . Both approximations lead to the well-known Michaelis-Menten formula $v = \frac{VS}{K + S}$.

Some remarks are in order.

- Despite the often simultaneous appearance of QEA and QSSA in chemistry parlance, it is important to realize that they are mathematically rather different approximations. Formally, on a reaction network represented as a species-reaction bipartite graph, QEA operates on reaction nodes and QSSA on species nodes.²⁰ That they lead to the same formula for Fig. 1.2 is an exception rather than the rule: they usually lead to drastically different approximations (see below).
- The connection between separation of concentration scale and QSSA can be formalized by singular perturbation [91]: $\epsilon = \frac{E_0}{S_0}$ now becomes the small parameter and, upon nondimensionalization, $\frac{dC}{dt}$ becomes $\epsilon \frac{dc}{d\tau}$, where c and τ are nondimensionalized C and t , respectively; $\frac{dC}{dt} \rightarrow 0$ as $\epsilon \rightarrow 0$.

Fig. 1.2 appears in almost every biochemistry textbook as an introduction to the two approximations in deriving the Michaelis-Menten rate law, and in almost every one of those textbooks, it is stated that the separation-of-concentration-scale

²⁰This has implications in the applications of MBAM to reaction networks (Section 2.1).

assumption and hence QSSA is more valid than the separation-of-time-scale assumption and QEA. This is reasonable for metabolic reactions: reactants are small metabolites that come in much greater quantities than the large enzymes that catalyze their reactions (Section 1.1), while the separation of time-scale is not universal for metabolic enzymes [8]. However, two issues arise here. First, the Michaelis-Menten formula is also often used in models for signal transduction and gene regulatory networks, where the reactants are proteins and come in comparable quantities to their enzymes. Second, despite the claim of greater validity for QSSA, almost invariably QEA is used for the rate laws in metabolic modeling. To see this, let us look at an example of a reversible enzyme-catalyzed reaction involving two substrates and two products $A + B \leftrightarrow P + Q$.

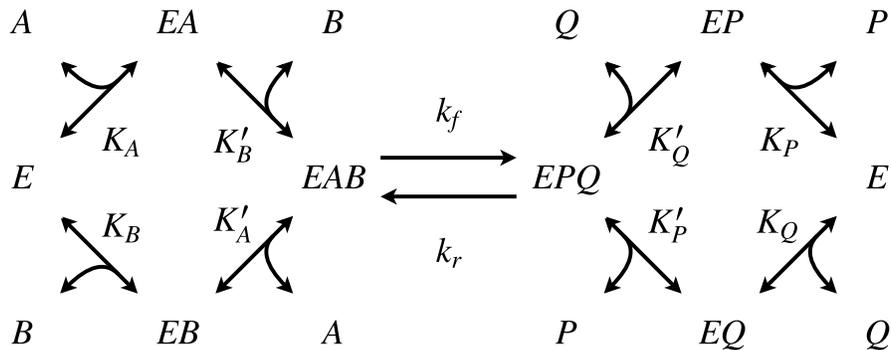


Figure 1.3: A Michaelis-Menten reaction scheme for an enzyme-catalyzed reaction $A + B \leftrightarrow P + Q$. The reaction scheme involves two substrates and two products forming complexes in any order, and is often called "random bi-bi" in literature. The equilibrium constants, in this context more often called dissociation constants, of all the reversible enzyme-reactant formation-dissociation elementary steps, are labeled.

Fig. 1.3 shows a supposed reaction scheme for reaction $A + B \leftrightarrow P + Q$. To derive Michaelis-Menten-like rate laws for it, one again write down the equations for all the involved chemicals. To apply QEA, one assumes that all the elementary steps involving enzyme-reactant complex formation and dissociation are at equilibrium, and the resulting rate law looks like this:

$$v = \frac{V_f \frac{A}{K_A} \frac{B}{K_B} - V_r \frac{P}{K_P} \frac{Q}{K_Q}}{1 + \frac{A}{K_A} + \frac{B}{K_B} + \frac{A}{K_A} \frac{B}{K_B} + \frac{P}{K_P} + \frac{Q}{K_Q} + \frac{P}{K_P} \frac{Q}{K_Q}}. \quad (1.2)$$

To apply QSSA, one sets $\frac{dEA}{dt} = \frac{dEB}{dt} = \frac{dEAB}{dt} = \frac{dEP}{dt} = \frac{dEQ}{dt} = \frac{dEPQ}{dt} = 0$, and solves the resulting algebraic equations. The resulting rate law looks like this.

$$v = \frac{N}{D}, \text{ where}$$

$$N = \tilde{k}_1 BPQ^2 + \tilde{k}_2 BP^2Q + \tilde{k}_3 APQ^2 + \tilde{k}_4 AP^2Q + \tilde{k}_5 AB^2Q + \tilde{k}_6 AB^2P + \tilde{k}_7 A^2BQ + \tilde{k}_8 A^2BP + \tilde{k}_9 PQ^2 + \tilde{k}_{10} P^2Q + \tilde{k}_{11} BPQ + \tilde{k}_{12} APQ + \tilde{k}_{13} ABQ + \tilde{k}_{14} ABP + \tilde{k}_{15} AB^2 + \tilde{k}_{16} A^2B + \tilde{k}_{17} PQ + \tilde{k}_{18} AB,$$

$$D = \tilde{k}'_1 ABPQ^2 + \tilde{k}'_2 ABP^2Q + \tilde{k}'_3 AB^2PQ + \tilde{k}'_4 A^2BPQ + \tilde{k}'_5 BPQ^2 + \tilde{k}'_6 BP^2Q + \tilde{k}'_7 B^2PQ + \tilde{k}'_8 APQ^2 + \tilde{k}'_9 AP^2Q + \tilde{k}'_{10} ABQ^2 + \tilde{k}'_{11} ABPQ + \tilde{k}'_{12} ABP^2 + \tilde{k}'_{13} AB^2Q + \tilde{k}'_{14} AB^2P + \tilde{k}'_{15} A^2PQ + \tilde{k}'_{16} A^2BQ + \tilde{k}'_{17} A^2BP + \tilde{k}'_{18} PQ^2 + \tilde{k}'_{19} P^2Q + \tilde{k}'_{20} BQ^2 + \tilde{k}'_{21} BPQ + \tilde{k}'_{22} BP^2 + \tilde{k}'_{23} B^2Q + \tilde{k}'_{24} B^2P + \tilde{k}'_{25} AQ^2 + \tilde{k}'_{26} APQ + \tilde{k}'_{27} AP^2 + \tilde{k}'_{28} ABQ + \tilde{k}'_{29} ABP + \tilde{k}'_{30} AB^2 + \tilde{k}'_{31} A^2Q + \tilde{k}'_{32} A^2P + \tilde{k}'_{33} A^2B + \tilde{k}'_{34} Q^2 + \tilde{k}'_{35} PQ + \tilde{k}'_{36} P^2 + \tilde{k}'_{37} BQ + \tilde{k}'_{38} BP + \tilde{k}'_{39} B^2 + \tilde{k}'_{40} AQ + \tilde{k}'_{41} AP + \tilde{k}'_{42} AB + \tilde{k}'_{43} A^2 + \tilde{k}'_{44} Q + \tilde{k}'_{45} P + \tilde{k}'_{46} B + \tilde{k}'_{47} A + \tilde{k}'_{48},$$

and the \tilde{k} s and \tilde{k}' s are products of rate constants of the elementary reactions.

This example vividly illustrates the difference between the two approximations: QEA usually begets simpler approximations than QSSA ²¹. For this reason, QEA is preferentially used in deriving rate laws in biochemistry despite that QSSA is generally considered to be more valid.

There are two more subtleties on applying QEA to derive Michaelis-Menten rate laws and they come from thermodynamics. First, Fig. 1.3 has different dissociation constants between, for example, $E+A \leftrightarrow EA$ (which is K_A) and $EB+A \leftrightarrow EAB$ (which is K'_A); this makes sense, as there are no *a priori* reasons to believe that the two different reactions share the same dissociation constant. Since E is at equilibrium with EA and EA is at equilibrium with EAB and equilibrium is a transitive relation ²², E is also at equilibrium with EAB ; since there are two paths between E and EAB , the equilibrium constants between E and EAB for the two paths should be the same: $K_A K'_B = K_B K'_A$, or $\frac{K_A}{K'_A} = \frac{K_B}{K'_B} = r$. That is to say, the four dissociation constants between E , EA , EB and EAB are not independent of each other, but are subject to a constraint arising from thermodynamics; this constraint is sometimes known as the *Wegscheider condition* in literature. Given such a constraint, one almost universal practice in the field is to equate K_A and K'_A , and similarly for the other dissociation constants; that is, r is set to one, giving rise to Eq. 1.2 which has no K'_A , etc. Second, when the reaction $A + B \leftrightarrow P + Q$ is at equilibrium, its

²¹The rate law from QSSA, horrendous as it looks, can be understood via the help of the following formalism: the enzyme-reactant complexes form a reaction network whose steady-state solutions can be described using the *matrix-tree theorem* on the corresponding *graph Laplacian* [41], which is equivalent to the *King-Altman method* commonly used in enzyme kinetics in deriving the type of rate laws [53].

²²Also known as the zeroth law of thermodynamics.

rate vanishes and so does the numerator of Eq. 1.2: $V_f \frac{A}{K_A} \frac{B}{K_B} - V_r \frac{P}{K_P} \frac{Q}{K_Q} = 0$; at the same time, the equilibrium condition means that $\frac{PQ}{AB} = K_E$, where K_E is the equilibrium constant for the reaction. The two equations constitutes another constraint on the parameters: $K_E = \frac{V_f/(K_A K_B)}{V_r/(K_P K_Q)}$; this constraint is often known as the *Haldane condition* in literature. Since equilibrium constants for metabolic reactions are often far better known than any other types of parameters [79], one often plugs the constraint back into Eq. 1.2 and eliminates one parameter V_r :

$$v = \frac{\frac{V_f}{K_A K_B} (AB - PQ/K_E)}{1 + \frac{A}{K_A} + \frac{B}{K_B} + \frac{A}{K_A} \frac{B}{K_B} + \frac{P}{K_P} + \frac{Q}{K_Q} + \frac{P}{K_P} \frac{Q}{K_Q}}. \quad (1.3)$$

We call such type of Michaelis-Menten rate laws compliant with the Haldane condition *Michaelis-Menten-Haldane* rate laws. Similarly, we call any mass-action rate laws compliant with the Haldane condition *mass-action-Haldane* rate laws (since for similar reasons the forward and reverse rate constants have to satisfy $k_f/k_r = K_E$), which look like the numerator of *Michaelis-Menten-Haldane* rate laws, or:

$$v = k_f (AB - PQ/K_E) \quad (1.4)$$

Eqs. 1.2 and 1.3 are parametrized using parameters V 's and K 's: they are the familiar maximal velocities and Michaelis constants. The rate laws can be reparametrized to yield some mathematically equivalent, and often simpler-to-read alternatives. For example, substituting $\frac{V_f}{K_A K_B}$ with k_f , $\frac{V_r}{K_P K_Q}$ with k_r , and $\frac{1}{K_A}$ with b_A , etc., gives the same rate laws in a different parametrization:

$$v = \frac{k_f AB - k_r PQ}{1 + b_A A + b_B B + b_A b_B AB + b_P P + b_Q Q + b_P b_Q PQ}$$

$$v = \frac{k_f (AB - PQ/K_E)}{1 + b_A A + b_B B + b_A b_B AB + b_P P + b_Q Q + b_P b_Q PQ}$$

We call Michaelis-Menten(-Haldane) rate laws in such forms *kb parametrization*, and will use them in places where mathematical simplicity rather than biological interpretability is preferred.

Lastly, we note that a few rate law families alternative to the one of Michaelis-Menten exist in literature. For example, lin-log rate laws [118] are partially inspired by thermodynamics, take a form of $\frac{v}{v^0} = \frac{E}{E^0} \left(1 + \sum_i \epsilon_i^0 \ln \frac{x_i}{x_i^0} \right)$ (superscript ⁰ denotes quantities at a reference state, E enzyme concentration, ϵ parameters called elasticities and x_i concentration of reactant i), and provide mathematical convenience in certain tasks such as MCA calculation. Power-law rate laws [94] are local linear approximations of Michaelis-Menten rate laws in logarithm rate and reactant concentrations. Here we focus on the Michaelis-Menten rate law family as they are the most commonly used, include mass-action rate laws as special cases, and are algebraic (unlike, eg, lin-log rate laws which involve transcendental functions).

1.4 The parameter problem

Among various topics in systems biology, few have attracted more controversies than the so-called “parameter problem” [40].

To have an appreciation of the problem, let us first take a look at the following histogram.²³

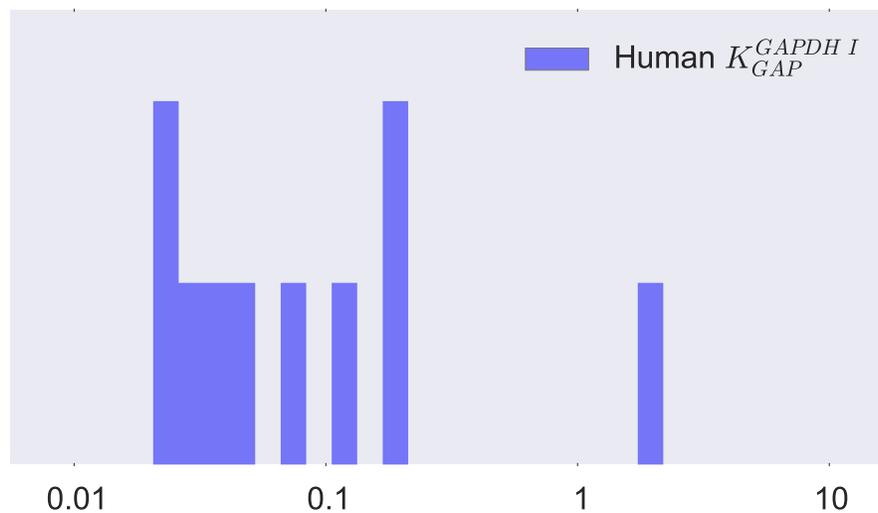


Figure 1.4: Histogram of measured Michaelis constant of GAPDH I for GAP in human. The data are taken from BRENDA, a manually-created repository of enzymatic data [95].

It plots the distribution of all recorded measurements in literature on a sin-

²³Thanks go to Marc Warmoes in helping prepare the data in this plot.

gle parameter: the Michaelis constant of the first isozyme ²⁴ of glyceraldehyde-3-phosphate dehydrogenase (GAPDH I) for substrate glyceraldehyde-3-phosphate (GAP) in species *Homo sapiens*.

This plot already tells quite a few stories:

- GAPDH is one of the best studied enzymes (it catalyzes the GAP-BPGA conversion in both glycolysis and Calvin cycle (Fig. 1.1), in opposite directions). For most of other enzymes, having such a histogram would be a dream. One may feel a sense of blessing and positive omen when he embarks upon a kinetic modeling project and finds that all parameters in the model have been measured and recorded at least once in human history.
- In our selection of data for the plot, we have carefully distinguished between different species (for example, chimpanzee data are excluded), different substrates (for example, data of similar substrates such as glyceraldehyde – without the phosphate group – are excluded) and different isozymes (for example, data for the second isozyme GAPDH II are excluded). In contrast, when one builds a kinetic model and the needed value of a parameter is missing in literature, a typical practice in the field is to resort to data of similar substrates, enzymes or species (if any) for proxys.
- Despite all the care taken, the histogram easily spans about two orders of

²⁴An enzyme can have multiple similar forms, called isozymes, that catalyze the same reaction, but have different kinetic and regulatory properties and are often differentially expressed among tissues.

magnitude, with no clear center of mass. ²⁵

What the plot does not reveal are two other perhaps more disconcerting facts:

- It was only recently that people started to realize the wild difference between the conditions under which traditional enzyme kinetic data were collected (that is, *in vitro* conditions) and the actual conditions inside a cell (that is, *in vivo* conditions), and the difference results in profound bias in parameter estimates. One aspect of those *in vitro* conditions is especially telling: they were decided to *optimize* enzyme activities. For example, if changing the pH of a test tube from the physiological level of 7.4 to 8.9 is found to increase enzyme activity, then a pH of 8.9 would be used for enzymatic assays.
- If *in vitro* has terribly betrayed *in vivo*, it turns out that *in silico* has been equally unfaithful to *in vitro*. The following figure, taken out verbatim from [109], vividly illustrates how using different data-fitting algorithms, especially those involving nonlinear transformations of variables in an attempt to reduce a nonlinear fitting problem to a linear one that are popular in enzyme kinetics, can profoundly affect the parameter estimates. We will leave the figure to tell the story.

We have provided a sketch of the parameter problem above. Now let us turn to its proposed solutions, which come no less interesting.

²⁵While one may argue that the rightmost point in the histogram represents an outlier and should be removed, the resulting distribution still spans an order of magnitude and conveys a qualitatively similar message.

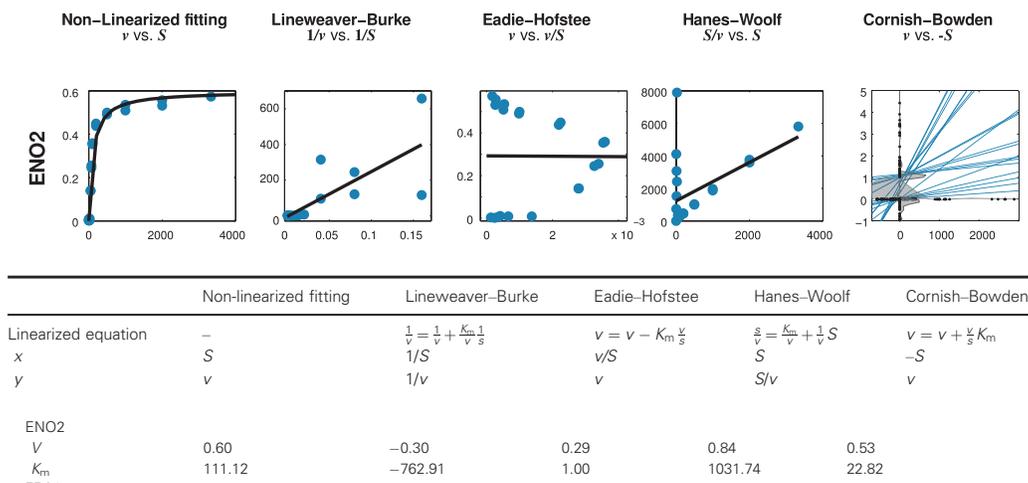


Figure 1.5: Woes of enzymatic data fitting. Taken out verbatim from [109], it shows how parameter estimates intimately depend on the fitting algorithm. The model is standard: $v = VS/(K_m + S)$, which “non-linearized fitting” directly uses, while all other fitting algorithms involve nonlinear transformations of it first, as shown in the figure. ENO 2: enolase 2.

The first class of proposed solutions essentially advocates still parametrizing the models somehow, and it can be further divided into a few possibilities. Aside from the traditional practice of plugging and chugging parameter values from literature that still lingers as the field is catching up to the problem, it has been proposed to re-measure all kinetic parameters, now under realistic and standard conditions [109, 110, 100] and hopefully with high-throughput technologies that may one day cover enzyme kinetic research, or alternatively, heuristics have been used in parametrizing kinetic models (such as equating Michaelis constants K_M for a substrate with its steady-state concentration) with an emphasis on the qualitative features of the resulting behaviors (eg, [99]).

Second, a vast body of work under the umbrella term *chemical reaction network theory* (CRNT) aims to understand what *dynamic properties* of a reaction network are determined from the *structure* of the network alone, that is, the stoichiometry of the network regardless of parameter values as long as the rate laws come from some general families such as mass action (eg, [32, 39]) or monotonic (eg, [55]). The approach tends to be highly mathematical, and the properties under study are generally qualitative and abstract (such as multistationarity, stability and persistence). The celebrated result of *deficiency zero theorem* gives an example of what this line of inquiry begets; trimming off some subtleties unimportant for the purpose of illustration here,²⁶ the theorem essentially says: if a chemical reaction network with mass-action rate laws is weakly reversible and has deficiency zero (a property that depends only on the network structure), then regardless of the values of rate constants, it has only one steady state and the steady state is locally asymptotically stable [33].

Third, another vast body of work under the umbrella term *constraint-based modeling* [59, 82] aims to predict metabolic behaviors involving steady-state fluxes without depending on knowledge of kinetic parameters. This body of work is specific to studying metabolic networks, and similarly makes heavy use of the network structural information, for the purpose of constraining the solution space (hence the name); unlike CRNT, however, it returns quantitative predictions, and to do so it introduces additional constraints, which further shrink the solution

²⁶They have something to do with conservation pools or "stoichiometric compatibility class" in the CRNT lingo.

space, and some type of *variational principles*, which break the tie in the remaining solution space and return explicit predictions. The popular *flux balance analysis* [81] provides an example of this class of methods. The first step uses the structural information N to constrain the solution space: the set of all steady-state fluxes satisfy $Nv = 0$ and hence span the right null space of N , geometrically shrinking the whole flux space to a convex *flux cone* in the flux space; the second step introduces additional constraints: the rate of each reaction v_i has to fall within a range $(-V_i^r, V_i^f)$, geometrically slicing out a finite flux cone out of the original infinite one; the third step introduces a variational principle, for example, the network flux should optimize biomass production rate, which upon optimization gives out a point as prediction, geometrically amounting to picking a vertex of the finite flux cone.

Fourth, *phenomenological modeling*²⁷ is sometimes used as an alternative approach (eg, [47]). Systems biology has been defined as "the study of how physiology emerges from molecular interactions" [43], and our parameter problem stems partly from molecular interactions being enormously complicated. In phenomenological modeling, one does not descend to and get drowned in the molecular details, but rather stay as close to physiology or phenotype as possible. The resulting models may hence lack mechanistic foundations, but are also free of the bulkiness and difficulty in parametrization that plague mechanistic models.

Fifth, with the deluge of omics data, *machine learning* has lately become a fash-

²⁷In biology it is also known as *phenotypic modeling* (eg, [58, 42]), and has a similar spirit to *geometric modeling* (eg, [29, 87]).

ionable approach (eg, [69]). The model class is statistical in nature and makes no attempt in faithfully representing the mechanisms. It sets its focus on achieving maximal predictive power and takes proper care of issues like generalizability and overfitting. As a result, despite the many parameters that are present in a typical machine learning model and can be similarly poorly constrained, it does not suffer the parameter problem like kinetic models do.

Sixth, *Bayesian statistics* provides a natural framework for dealing with parameter uncertainties (eg, [9, 61, 93]). Representing every quantity as a distribution, the flexible framework quantifies and encodes parameter uncertainties, and propagates them to predictions, all done systematically.

The last approach has its philosophical root in *statistical mechanics*, a branch of physics that studies how behaviors of a system arise from its microscopic details (that is, properties of its components). In doing so, it stumbled upon one simple yet profound discovery: simple system behaviors can emerge out of complicated microscopic details. Two important corollaries flow from such a discovery: one may not need to know all the details in order to predict system behaviors; when thinking about the details one should focus on identifying *therelevant* details that matter in determining system behaviors. This philosophy seems especially apt for systems biology: on one hand, the details are represented by the microscopic parameters, which the parameter problem suggests are multitude and poorly-known; on the other hand, it is not the details, but the system behaviors (that is, phenotypes) and their causal connections with the details that are

of interest. Following the first corollary suggests an effective approach that bypasses the parameter problem: using *collective data* describing system behaviors to parametrize models, which often turn to be *falsifiable* for some other system behaviors [16, 44, 102]. We call such an approach *collective modeling*.²⁸ Following the second corollary requires systematically representing parameter dependence of system behaviors, and this is done by *information geometry* [2, 103, 65]. We note that when applying the statistical mechanic approach to overcome the parameter problem, many procedures bear formal resemblance to those in Bayesian statistics, but the spirit represented by the two corollaries above is unique to the statistical mechanic approach.

Among all these approaches, we follow the statistical mechanic one. We believe that the approach, when framed using the language of information geometry, is general and especially suited to studying systems biology models, where details are complicated and their connections to system behaviors are murky but important; as we will see in Chapter 2, this line of inquiry does reward us with promising methods in revealing the relevant details. Among the other approaches, CRNT asks questions of a different nature that are only occasionally of interest to biologists; constraint-based modeling, in attempt to generate explicit predictions, inadvertently recreates its own parameter problem since the bounds on the reaction rates are often as poorly known as the parameters in kinetic modeling, and introduces some variational principles for tie-breaking that are often controversial [96, 67]; phenomenological modeling is tempting, but as we will see, information

²⁸We make up this term so it is far from standard.

geometry provides means to arrive at phenomenological models from the usual mechanistic kinetic models [107] which at once provide both phenomenological relevance and mechanistic connections; lastly, machine learning reveals no mechanism and metabolic modeling has not benefited from a cornucopia of data as much as some other areas in the post-genomic era do.

1.5 Data in metabolism research

In this section we list some common types of data collected on metabolic networks. They represent the central objects in metabolic modeling, hence the importance of a careful characterization of them. As a unifying thread, all these data types take the form of measurements of how some system quantities of interest (which we call z) change as some other controllable quantities (which we call u) are tuned. We use Z and U to denote the space of z and u respectively.²⁹ In Chapter 2, we will see how this commonality between different data types allows for abstraction and common procedures in modeling them.

First, we have *enzymatic data*, also known as enzyme kinetic data. We assume that this type of data is like what rate laws describe: reaction rates as functions of reactant concentrations.³⁰ Here, z is reaction rates v , and u is the reactant concen-

²⁹The following notations sometimes break the pattern: for example, both steady-state fluxes and the space are denoted J . However, in all cases the context should indicate which one is intended without confusion.

³⁰An alternative type is kinetic data, concentrations of reactants in a reaction over a course of time, which was the main type of data in enzyme kinetic studies until Michaelis and Menten in-

trations x ; we sometimes also call this type of data *VX data*. There are three key features of this data type: it is not on the whole network but on individual reactions; a century of enzyme kinetic research [72] made it the most abundant data type; consequently, it is the most commonly used data type in metabolic modeling.

Second, there are *kinetic data*, also known as time-course data or time-series data. This type of data describes how reactant concentrations change over time in a chemical reaction system. Here, z is the metabolite concentrations x , and u is the time t ; we sometimes also call this type of data *XT data*. Note that it is the standard data type in studying gene regulatory networks, but is used less in studying signal transduction networks and metabolic networks.

Third, researchers of metabolism often manipulate the external conditions of a metabolic network, typically the concentrations of some external metabolites, and measure how steady-state metabolite concentrations change accordingly. For example, one may measure how the concentrations of metabolites in glycolysis depend on the glucose concentration of the media in a tissue culture. Here, z is the steady-state metabolite concentrations, which we still denote as x (with the implicit understanding that it now stands for $x(\infty)$), and u is the external metabolite concentrations, which we denote c ; we call this type of data *XC data*. To model *XC data*, one solves Eq. 1.1 for steady states, $Nv(x, p) = 0$, which, since external metabolite concentrations c are part of p , gives a function from c to x .

roduced the idea of measuring only the initial part of a time course before non-negligible product build-up [28].

Fourth, researchers of metabolism similarly measure how steady-state fluxes of a network depend on its external metabolite concentrations. Here, z is the steady-state fluxes, which we call J , and u is the external metabolite concentrations, c ; we call this type of data *JC data*. To model *JC data*, one starts with the function that maps c to x derived in modeling *XC data*, and plugs x into the rate laws v to get J , resulting in a function that maps c to J .

Lastly, researchers of metabolism also measure how steady-state concentrations or fluxes of a metabolic network depend on its internal enzyme activities. There are a few possibilities for changing internal enzyme activities, including changing *enzyme concentrations* (for example, through constructing transgenic antisense lines with lower enzyme expressions), changing *enzyme catalytic efficiencies* (for example, by generating mutant lines with altered enzyme catalytic efficiencies) and changing *enzyme inhibitor or activator concentrations*. In this thesis, we use the simple scenario of changing enzyme concentrations as a representative for this class of data types. Here, z is the steady-state concentrations x or fluxes J , and u is the internal enzyme concentrations E ; we call such type of data *XE* or *JE data*. To model *XE* or *JE data*, one selects a subset of parameters that are proportional to enzyme concentrations, such as the maximal velocities in Michaelis-Menten rate laws, and establishes the functions mapping from those parameters to the steady-state concentrations or fluxes in the same way as one would model *XC* or *JC data*.³¹

³¹Typically, readouts of enzyme concentration modification are *relative*, meaning that it is hard to know the absolute enzyme concentrations for the wildtype and mutants, but it is easy to know the percentage of change between two states. We nonetheless call such type of data with only

Table 1.1 summarizes all the data types.

Table 1.1: Common data types in metabolic studies. Capitalized symbols denote the space of variables (that is, the set of all possible values that the variables can take). Z : dependent variables; U : independent variables; X : (kinetic or steady-state) reactant concentrations; V : reaction rates; T : time; J : steady-state fluxes; E : enzyme concentrations. XC data: steady-state metabolite concentrations as a function of external metabolite concentrations; JC data: steady-state fluxes as a function of external metabolite concentrations; XE data: steady-state metabolite concentrations as a function of internal enzyme concentrations; JE data: steady-state fluxes as a function of internal enzyme concentrations.

Name	Z	U	References of exemplary data
Enzymatic or VX data	V	X	[116]
Kinetic or XT data	X	T	[71]
XC data	X	C	[6]
JC data	J	C	[6, 49],[114] and the references therein ³²
XE data	X	E	[98]
JE data	J	E	[56, 98]

Intuitively, these different data types describe different *behaviors* of a metabolic network that is modeled by Eq. 1.1: kinetic data describe kinetic behaviors, JC data describe JC behaviors, etc. Therefore, one can think that *measuring behaviors*

relative information of enzyme concentration changes XE or JE data, and to model it, one adds a ratio parameter r to each of the parameters proportional to enzyme concentrations, in which case the wildtype would have $r = 1$ and setting r to other values would simulate mutants.

³² JC data is the main data type in photosynthesis research, generated by the so-called gas-exchange experiments (eg, [19]).

of a certain type produces data of the type. All these notions will be formalized in Chapter 2 to pave way for the development of some general modeling concepts and techniques.

1.6 A list of toy models

To deepen our understandings of kinetic models of metabolic networks, we often employ the approach of analyzing *toy models*, kinetic models of some simple “toy” networks for which mathematical derivation is tractable and analytical insight accessible. In this section, we describe a few toy models that are used repeatedly in our later studies. First, Fig. 1.6 describes three toy networks.

Next, we list below for convenience the functional forms of some rate laws (see Section 1.3 for details). Furnishing the toy networks with these rate laws produces toy models.

- Mass-action rate laws (ma)
 - For a reaction $A \leftrightarrow P$, $v = k_f A - k_r P$
 - For a reaction $A + B \leftrightarrow 2P$, $v = k_f AB - k_r P^2$
 - For a reaction $A + B \leftrightarrow P + Q$, $v = k_f AB - k_r PQ$
- Mass-action-Haldane rate laws (mah)
 - For a reaction $A \leftrightarrow P$, $v = k_f \left(A - \frac{P}{K_E} \right)$

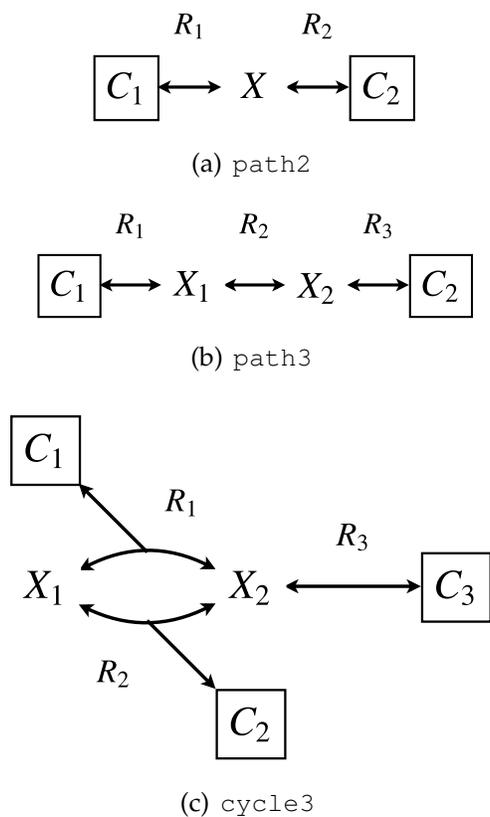


Figure 1.6: Three toy networks that are used repeatedly in the next two chapters. (a) `path2`, a pathway of two single-substrate single-product reactions. (b) `path3`, a pathway of three single-substrate single-product reactions. (c) `cycle3`, a cycle of three reactions: $C_1 + X_1 \leftrightarrow 2X_2$; $X_2 + C_2 \leftrightarrow X_1$; $X_2 \leftrightarrow C_3$. All chemicals in boxes (C_1 , C_2 and C_3) represent external metabolites with fixed concentrations.

- For a reaction $A + B \leftrightarrow 2P$, $v = k_f \left(AB - \frac{P^2}{K_E} \right)$
- For a reaction $A + B \leftrightarrow P + Q$, $v = k_f \left(AB - \frac{PQ}{K_E} \right)$
- For mathematical convenience and clarity, we often set $K_E = 1$ in our derivations. The results thereof are equally valid without such treatment.

- Michaelis-Menten rate laws (mm)

- For a reaction $A \leftrightarrow P$, $v = \frac{V_f \frac{A}{K_A} - V_r \frac{P}{K_P}}{1 + \frac{A}{K_A} + \frac{P}{K_P}} = \frac{k_f A - k_r P}{1 + b_A A + b_P P}$
- For a reaction $A + B \leftrightarrow 2P$,

$$v = \frac{V_f \frac{A}{K_A} \frac{B}{K_B} - V_r \frac{P^2}{K_P^2}}{1 + \frac{A}{K_A} + \frac{B}{K_B} + \frac{A}{K_A} \frac{B}{K_B} + \frac{2P}{K_P} + \frac{P^2}{K_P^2}}$$

$$= \frac{k_f AB - k_r P^2}{1 + b_A A + b_B B + b_A b_B AB + 2b_P P + 2b_P^2 P^2}$$

- For a reaction $A + B \leftrightarrow P + Q$,

$$v = \frac{V_f \frac{A}{K_A} \frac{B}{K_B} - V_r \frac{PQ}{K_P K_Q}}{1 + \frac{A}{K_A} + \frac{B}{K_B} + \frac{A}{K_A} \frac{B}{K_B} + \frac{P}{K_P} + \frac{Q}{K_Q} + \frac{P}{K_P} \frac{Q}{K_Q}}$$

$$= \frac{k_f AB - k_r PQ}{1 + b_A A + b_B B + b_A b_B AB + b_P P + b_Q Q + b_P b_Q PQ}$$

- The two formulae in each stoichiometry scenario represent the same rate law in different parametrizations: k 's represent ratios of V 's and K 's and b 's represent reciprocals of K 's.

- Michaelis-Menten-Haldane rate laws (mmh)

- For a reaction $A \leftrightarrow P$, $v = \frac{V_f \left(A - \frac{P}{K_E} \right)}{1 + \frac{A}{K_A} + \frac{P}{K_P}} = \frac{k_f \left(A - \frac{P}{K_E} \right)}{1 + b_A A + b_P P}$
- The rate laws for reactions $A + B \leftrightarrow 2P$ and $A + B \leftrightarrow P + Q$ are derived from `mm` similarly.

We adopt the following naming scheme.

- A model's name consists of three parts: network topology, number of reactions and assumed rate laws. For example, `path2mah` represents the model for a two-reaction pathway (Fig. 1.6(a)) with mass-action-Haldane rate law (`mah`) for both reactions; similarly, `cycle3mm` represents the model for a three-reaction cycle (Fig. 1.6(c)) with Michaelis-Menten rate law for all reactions.
- When a model is used to describe certain type of data (Section 1.5), we use a tuple to designate the (model, data type) pair. For example, `(path2mah, XT)` represents the modeling scenario in which model `path2mah` is used to describe kinetic data collected from a metabolic network. Since data of a certain type simply represent measurements of behaviors of the type, we sometimes also refer to the tuple as a (model, behavior type) pair. (Chapter 2 formalizes the notion here.)

There is one toy model, `path2mah`, that comes up a lot in our subsequent studies, so we solve it here for later references.

The model reads:

$$\frac{dx}{dt} = v_1 - v_2 = k_1(C_1 - x) - k_2(x - C_2), \quad x(0) = x_0 \quad (1.5)$$

First, we solve for *XC* behaviors of the model, which is the simplest. Setting the right hand side of the ODE to zero, we have $k_1(C_1 - x) - k_2(x - C_2) = 0$, which gives:

$$x(C, k) = \frac{k_1 C_1 + k_2 C_2}{k_1 + k_2}. \quad (1.6)$$

The solution can be interpreted as the mean of C_1 and C_2 weighted by the two rate constants. Next, we solve for the kinetic behaviors of the model, and the solution reads:

$$x(t, k) = \frac{k_1 C_1 + k_2 C_2}{k_1 + k_2} + \left(X_0 - \frac{k_1 C_1 + k_2 C_2}{k_1 + k_2} \right) e^{-(k_1 + k_2)t}, \quad (1.7)$$

where C_1 , C_2 and x_0 are considered constants, not parameters. We note that $\frac{k_1 C_1 + k_2 C_2}{k_1 + k_2}$ is the steady-state solution of the model (Eq. 1.6), and hence the equation can be interpreted as the steady-state solution plus a transient that decays exponentially.

Last, we solve for the *JC* behaviors by plugging Eq. 1.6 back into the rate laws, and the solution reads:

$$J(C, k) = \frac{k_1 k_2}{k_1 + k_2} (C_1 - C_2) = \tilde{k} \frac{C_1 - C_2}{2}, \quad (1.8)$$

where \tilde{k} represents the *harmonic mean* of k_1 and k_2 . Hence the solution can be interpreted as a new mass-action-Haldane rate law with \tilde{k} as the new rate constant and the chemical potential difference between C_1 and C_2 averaged by the number of reactions as the driving term. This interpretation is valid for an n-reaction pathway with mass-action-Haldane rate laws.

Before we conclude, we mention two more toy models that are often mentioned in the literature of information geometry and are used in Chapter 2.

Model 1: sum of two exponentials (sum2exp)

$$z(t, p) = e^{-p_1 t} + e^{-p_2 t}.$$

Model 2: difference between two exponentials (diff2exp)

$$z(t, p) = e^{-p_1 t} - e^{-p_2 t}.$$

CHAPTER 2

INFORMATION TOPOLOGY

This chapter starts with some introduction of information geometry and its natural extension information topology from a particular angle; beginning in Section 2.3, we build upon existing results and further the framework which help our studying kinetic models of metabolism in Chapter 3.

2.1 An overview of information geometry

A common purpose of developing a mathematical model \mathcal{M} for a system is to predict certain *behaviors* of the system, which typically come in two forms: one may want to predict certain quantities z , or their dependence on some variables u , that is, $z(u)$. For example, for the model `path2mah` (Section 1.6, Eq. 1.5), one may want to simply predict the steady-state flux J , or how J depends on external metabolite concentrations C , that is, $J(C)$ (Eq. 1.8).

In either case, quantities z often depend on some other variables as well. For example, for the model `path2mah` we have solved the formula of x as a function of t (Eq. 1.7): $x = x(t, k_1, k_2, C_1, C_2, x_0) = \frac{k_1 C_1 + k_2 C_2}{k_1 + k_2} + \left(x_0 - \frac{k_1 C_1 + k_2 C_2}{k_1 + k_2} \right) e^{-(k_1 + k_2)t}$, in which x depends not only on t , but also on five other variables. Some of these variables may have known values, say, $c = (C_1, C_2, x_0)$, and others do not, say, $p = (k_1, k_2)$; hence we have $x = x(t, p, c)$.

What we have above is essentially the following procedure. Given a mathematical model \mathcal{M} , we first identify some quantities of interest z that are *computable* from \mathcal{M} , and then *partition* the variables z depend on into three groups: variables u whose dependence relationships with z we are interested in, and among the rest, variables p whose values are unknown and variables c whose values are known. We call z **dependent variables**, u **independent variables**, p **parameters** and c **constants**. Symbolically, we have $z = M(u, p, c)$. Note that if we are only interested in z but not its dependence on some u , u is simply an empty vector and its space an empty space, which means that the form $z = M(u, p, c)$ incorporates this case as well. Similarly, p and c can also be empty vectors.¹ Also note that constants c can be absorbed into the definition of M by plugging in their values, hence we can simply write $z = M(u, p)$. Using Z , U and P to denote the spaces of z , u and p , respectively, we have $M : U \times P \rightarrow Z$, which we call the **model map**.

For the model `path2mah`, $X = X(t, k)$ (1.7) is a model map that describes kinetic behaviors, and so are $X = X(C, k)$ (Eq. 1.6) and $J = J(C, k)$ (Eq. 1.8), which describe XC and JC behaviors, respectively. Note that in this case *one model gives rise to multiple model maps*. It is not hard to understand why: given the procedure above, different choices of z and partition of u, p, c would lead to different model maps.²

Table 2.1 contains some of our most familiar models and some of their model

¹Physical scientists commonly praise their models for having “no free parameters” (and hence attribute their models’ ability of fitting an elephant to no fluke), which means that their p is an empty vector.

²One may say that different partitions assign different *semantics* to the variables, which are subject to different treatments by the *syntactic rules* of information geometry (see below).

Table 2.1: A list of models and their model maps. Some models are formulated in the model-map form in the first place (those with “—” as their model maps indicating the same as the model), while others are formulated as something else that give rise to some model maps upon selection of some z, u, p and c . Universal gravitation: three different partitions of u, p and c are given, where in the last two cases G is considered constant making p an empty vector in the second case, and the third case corresponds to the common “inverse-square law” interpretation of the model; technically, M have different domains in all three cases and is hence different, despite the same formula. Predator-prey: an example where the model map does not admit an explicit formula and is represented computationally. Neural network: a single-layer artificial neural network model that has a logistic function with parameter k controlling the steepness as its activation function.

Model name	\mathcal{M}	M	z	u	p
Michaelis-Menten	$v = \frac{V_m x}{K_M + x}$	—	v	x	V_m, K_M
Universal gravitation	$F = \frac{Gm_1 m_2}{r^2}$	—	F	$\frac{m_1, m_2, r}{m_1, m_2, r}$ r	$\frac{G}{()}$ m_1, m_2
Normal distribution	$y = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$	—	y	x	μ, σ
Linear regression	$Y = \beta_0 + \beta_1 X + \epsilon, \epsilon \sim N(0, \sigma^2)$	$y = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(\beta_0 + \beta_1 x)^2}{2\sigma^2}}$	y	x	β_0, β_1, σ
Predator-prey	$\dot{x} = \alpha x - \beta xy, x(0) = x_0$ $\dot{y} = \delta xy - \gamma y, y(0) = y_0$	$z = M(t, p)$	x, y	t	$\alpha, \beta, \delta, \gamma$ x_0, y_0
Neural network	$y = \frac{1}{1 + e^{-k \sum_i w_i x_i}}$	—	y	x	w, k
Ising model	$y = \frac{1}{Z} e^{-\beta \left(- \sum_{\langle i, j \rangle} J_{ij} \sigma_i \sigma_j - \mu \sum_i h_i \sigma_i \right)}$	—	y	σ	β, J, μ, h
path2mah	$\dot{x} = k_1(C_1 - x) - k_2(x - C_2)$ $x(0) = x_0$	$x = \frac{k_1 C_1 + k_2 C_2}{k_1 + k_2}$ $J = \frac{k_1 k_2}{k_1 + k_2} (C_1 - C_2)$	$\frac{x}{J}$	C_1, C_2	k_1, k_2

maps, suggesting that many models can be cast into the form a model map. Since the form serves as the foundation of an information geometric analysis (see below), the generality of the form implies the generality of information geometry as an approach to analyzing models.

Next, we formalize the notion of "behaviors". For a given model map $z = M(u, p)$, we are interested in how z depends on u , which inspires us to define the *functional space*, $B = Z^U$,³ consisting of all functions mapping from U to Z , which we call the **behavioral space**. Then a model map $M : U \times P \rightarrow Z$ induces a map $F : P \rightarrow Z^U$, which we call the **behavioral map**. In this way, we isolate model behaviors which are our central object of interest, and highlight their dependence on parameters. This is one fundamental perspective of information geometry, which we call the *functional perspective*.

In the light of this formalization, all model maps in Table 2.1 give rise to some behavioral spaces and their associated behavioral maps. For example, for the Michaelis-Menten model, we have behavioral space $B = V^X$, and behavioral map $F : P \rightarrow V^X$. Fig. 2.1 illustrates these concepts for this model.

Since one model map M corresponds to one behavioral map F and one model \mathcal{M} can give rise to multiple model maps M 's, it follows that *one model \mathcal{M} can give rise to multiple behavioral maps F 's, each with its own parameter space P and behavioral space B* . For example, Table 1.1 lists some common data types in metabolism re-

³It is standard in mathematics to denote the set of all functions mapping from set X to set Y as Y^X (because if X and Y are finite then the cardinality of the functional space is $|Y|^{|X|}$).

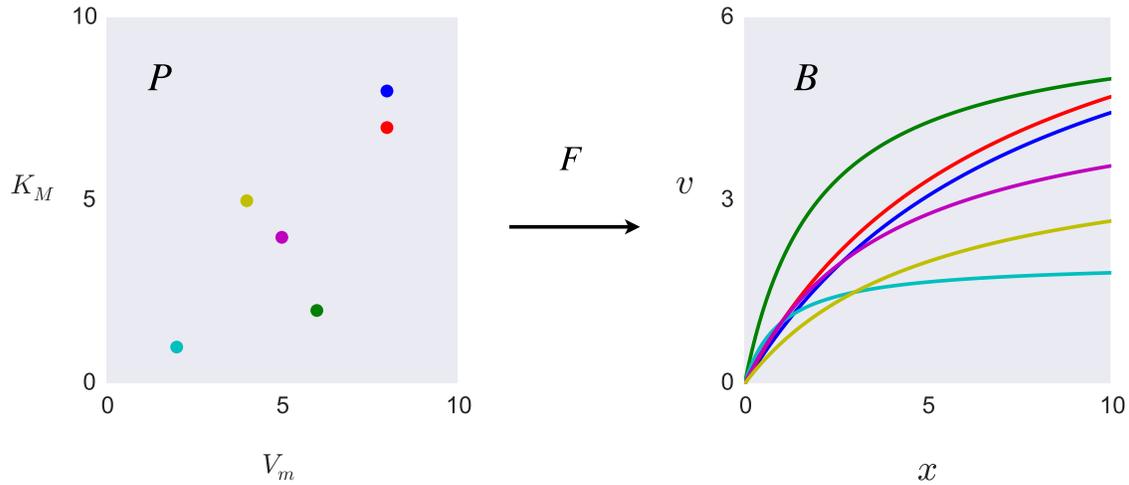


Figure 2.1: Parameter space, behavioral space and behavioral map for the Michaelis-Menten model $v = \frac{V_m x}{K_M + x}$. The left subfigure shows the parameter space P consisting of all possible (V_m, K_M) 's, of which six representatives are plotted for illustration (the six colored dots). The behavioral map F maps any point (V_m, K_M) in the parameter space to a point in the behavioral space which is a curve $v = v(x, V_m, K_M) = \frac{V_m x}{K_M + x}$. The right subfigure shows the behavioral space B consisting of all such curves, of which six are plotted corresponding to the points in parameter space with the same color (the six colored curves).

search, which we can now interpret as representing different behavioral spaces: kinetic data represent behavioral space X^T , JC data represent behavioral space J^C , etc; all these behavioral spaces can be derived from a single kinetic model (Eq. 1.1).⁴ As we will see later, from an information geometric viewpoint, among

⁴One can understand the multiplicity of behavioral spaces for a model in the following way. A model \mathcal{M} is often developed to describe a particular type of behaviors of a system: for example, a kinetic model like Eq. 1.1 is developed to describe the kinetic behaviors of a metabolic network

all the behavioral maps from a model those with same parameter space as the domain, achieved through some particular partitions of variables, are of special interest (Sections 2.5 and 3.1, and Fig. 2.4(e)). For example, for the model `path2mah`, the behavioral spaces X^C , X^T and J^C as defined in Table 2.1 share the same parameter space.⁵ Fig. 2.2 summarizes such a structure.

An element $z(u)$ from a behavioral space $B = Z^U$ describes how z depends on u for all u . For example, kinetic behaviors $x(t)$ of model `path2mah` predict x for any $t > 0$, and enzymatic behaviors $v(x)$ of the Michaelis-Menten model predicts v for any $x > 0$. However, when U is infinite, it is impossible to experimentally measure z for all $u \in U$; invariably, measurements can only be made on a *finite subset* of U . For example, kinetic data consist of measurements at a finite set of time $\{t_1, t_2, \dots, t_k\}$, and enzymatic data record v at a finite set of reactant concentrations

(hence the name “kinetic model”). However, a system can display many types of behaviors: for example, a metabolic network can display both kinetic behaviors and a variety of steady-state behaviors (Table 1.1). If it so happens that a model originally developed for describing a particular type of behaviors can be “recycled” to describe other types, then we have one model giving rise to multiple model maps. In our case, it happens that a kinetic model of a metabolic network can also describe its steady-state behaviors, likely because steady-state behaviors constitute a *subset* of kinetic behaviors as t is fixed to infinity. In general, we expect that most models can be “recycled” in this way and made able to describe system behaviors different from the originally intended ones.

⁵When functions in two functional spaces $Y_1^{X_1}$ and $Y_2^{X_2}$ share the domain $X_1 = X_2 = X$, the two functional spaces can be combined to form a new functional space Y^X , where Y is the cartesian product of the original two codomains $Y = Y_1 \times Y_2$. For this reason, two behavioral spaces with the same U can be combined into a new behavioral space (Z_1^U and Z_2^U form Z^U where $Z = Z_1 \times Z_2$) while behavioral spaces with different U 's must be considered distinct (because *by convention* we require that a behavioral space must have the Z^U structure). Also for this reason, two behavioral maps with the same parameter space P as the domain can also be similarly combined: $F_1 : P \rightarrow B_1$ and $F_2 : P \rightarrow B_2$ form $F : P \rightarrow B$ where $B = B_1 \times B_2$. As an example of the first scenario, in the predator-prey model in Table 2.1 behavioral spaces corresponding to $x(t)$ and $y(t)$ can be combined to form one corresponding to $z(t)$ where $z = (x, y)$; as an example of the second scenario, in the `path2mah` model in Table 2.1 the behavioral maps that map to X^C and J^C can be combined to form one that maps to $X^C \times J^C$ whose elements are $(x(C), J(C))$.

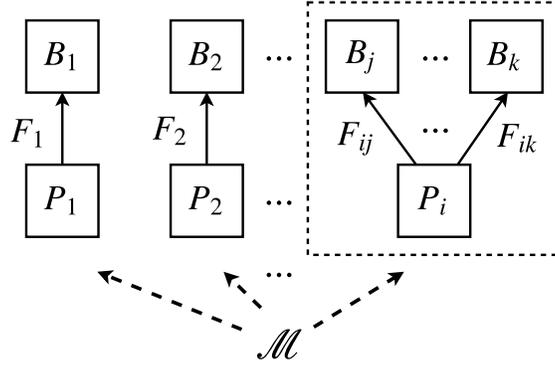


Figure 2.2: One mathematical model can give rise to multiple behavioral maps. For a model \mathcal{M} , different choices of z and partitions of u , p and c give rise to different model maps $M : U \times P \rightarrow Z$, which lead to different behavioral maps $F : P \rightarrow U^Z$, each with its own parameter space P and behavioral space B . Of particular interest are those behavioral maps sharing the same parameter space as the domain (dashed box).

$\{x_1, x_2, \dots, x_l\}$. Such a finite subset of U is usually chosen to *sample* the part of U that is presumed of interest (eg, where z changes appreciably). To model data thus generated, we are naturally led to definitions of finite-dimensional analogues of B and F , where we use the concept of "sampling" to formalize this "finite experiment effect".

Given a behavioral space B , a **prediction space** Y is a Euclidean space that results from any finite **sampling** X of B : $Y = X(B)$, which induces a **prediction map** $f : P \rightarrow Y$ through composing X with F : $f = X \circ F$.⁶ An f derived this way

⁶The technical definitions of X and Y are the following. Let $z = (z_1, \dots, z_k)$ and for each z_i define a sampling X_i that maps U to U_i , a finite subset of U . Then X can be represented by such a collection of X_i 's, and applying X to B returns a finite-dimensional Y : $Y = Z_1^{U_1} \times \dots \times Z_k^{U_k} = \prod_{i=1, \dots, k} Z_i^{U_i}$. We allow the sampling X_i to be different for different z_i 's to accommodate the common

is said to have a **sampling structure**.⁷ Fig. 2.3 illustrates these concepts using the Michaelis-Menten model.

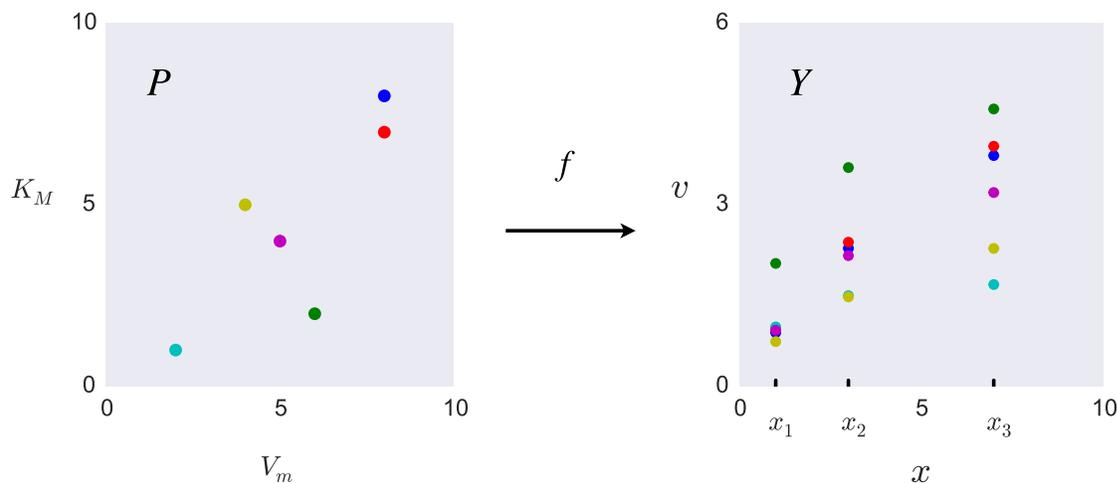


Figure 2.3: Parameter space, prediction space and prediction map for the Michaelis-Menten model $v = \frac{V_m x}{K_M + x}$. Imposing a sampling $X = \{x_1, x_2, x_3\}$ on the behavioral space of the model (Fig. 2.1) produces a prediction map f that maps any point (V_m, K_M) in the parameter space to a point in the prediction space which is a vector $\left(\frac{V_m x_1}{K_M + x_1}, \frac{V_m x_2}{K_M + x_2}, \frac{V_m x_3}{K_M + x_3}\right)$. The left subfigure shows the parameter space P in the same way as in Fig. 2.1. The right subfigure shows how the sampling keeps only a three points in each curve. Note that the right subfigure illustrates only the origin of the prediction space Y but not Y itself, as Y in this case is a three-dimensional Euclidean space (Fig. 2.6).

We need to make a few remarks on terminology.

scenario in which different interested quantities (say, metabolite concentrations) are measured at quantitatively different conditions (say, different times).

⁷As we will see, the sampling structure imposes much constraint and makes a prediction map different from a general map between two Euclidean spaces P and Y .

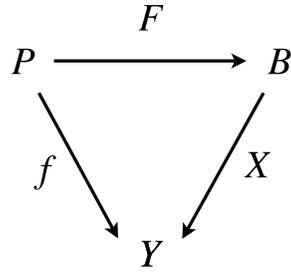
- When people talk about “models”, they sometimes mean behavioral maps or prediction maps as defined here. In many cases, this is justified, as the behavioral space and its sampling are tacitly agreed upon. In some other cases, this is unfortunate, as there is not an *a priori* consensus on the behavioral space or the sampling, and controversies sometimes follow.⁸ By separating models and behavioral spaces in our definitions, we wish to incorporate the common scenario where for a given model there is not a shared assumption on the behavioral space under discussion (eg, Table 1.1 in metabolism modeling), and highlight the fact many properties of a model actually depend on the choice of behavioral spaces. However, in the subsequent writings, when there is no risk of confusion, we sometimes also use the word “model” in place of “behavioral map”, to make it read more in accord with the mainstream lingo.
- Prediction space Y is also known as *data space*, especially when data are actually collected. The rationale for our choice of terminology is that terms like “experiments” and “data” imply the existence of actual measurements, which is not always the case as one sometimes studies those maps as mathematical objects in their own right and independent of or prior to experimental measurements. In this sense, “sampling” and “prediction” can be thought as *generalizations* of “experiments” and “data”, respectively; that is,

⁸For example, in [45] it is suggested that “most systems biology models are sloppy” (a property that will be explained later), meaning that the behavioral maps with respect to *kinetic behaviors* $x(t)$ are sloppy, while in [5], it is hinted that “some systems biology models may not be sloppy”, meaning that the behavioral maps with respect to *perturbation-kinetic behaviors* $x(t, C, E)$ are not sloppy, which prompts discussions on whether the models are “intrinsically sloppy” [22].

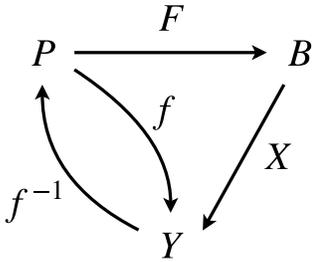
they are mathematical equivalents with a more general connotation.

- We refer to a behavioral space Z^U as describing "ZU behaviors" and modeling "ZU data". (This explains our terminology in Table 1.1.)

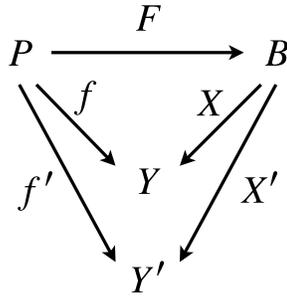
We note that the functional perspective and its ramifications admit some diagrammatic representations, which we summarize in Fig. 2.4.



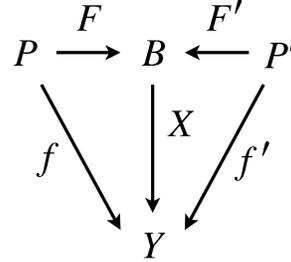
(a) Model prediction



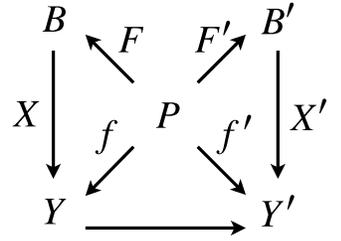
(b) Parameter estimation



(c) Experimental design



(d) Model selection



(e) Collective modeling

Figure 2.4: A unifying view of model prediction, parameter estimation, experimental design, model selection and collective modeling. The functional perspective of information geometry casts a few common modeling tasks into some kind of "commutative diagrams". (a): The diagram describes the fundamental relations between parameters, behaviors and predictions: A model \mathcal{M} describes how some behaviors depends on parameters through the behavioral map F mapping from a parameter space P to a behavioral space B ; a sampling X of B produces the prediction space Y and the prediction map f that maps from P to Y ; given a $p \in P$, f generates model predictions $y = f(p)$. (b) Parameter estimation can be thought as the inverse problem of model prediction: $f^{-1} : Y \rightarrow P$ (technically f^{-1} is a kind of pseudoinverse, with a projection preceding functional inversion). (c): Experimental design can be thought as an optimization over B , illustrated here by two competing X and X' , which produce two different Y and Y' with their f and f' ; the task is to select an X from B that produces the optimal f by some criteria.¹⁰(d): Model selection can be thought as comparing in the same B two different models, represented here by two F and F' , or when an X and an X' are chosen, comparing the f and f' in the same Y . (e): Collective modeling, a term we coined in Section 1.4 and emphasizes using collective data to predict collective behaviors without focusing on parameters, can be thought as a map from a B to another B' , or upon choosing an X and an X' , from a Y to another Y' .

The functional perspective of information geometry described above may be conceptually tidying and satisfying, but so far offers little actual advantage to practical calculation or technical problem solving. It is only after another perspective joins in when real power begins to emerge. We call it the *geometric perspective*: information geometry interprets behavioral maps *geometrically*.

We will focus on the case in which a model's behavioral map F is *differentiable*, as are most models in science (and all models in Table 2.1) and has also been the traditional focus of information geometry. Here the geometric perspective is most intuitive and the new power most apparent, all owing to a single fact: the image of F becomes a *differentiable manifold*, to which the powerful machinery of differential geometry is applicable.¹¹ For f , its image is not only a manifold, but also embedded in the Euclidean space Y . Since we rarely have the images of both F and f simultaneously appear, they will be collectively called *model manifold*, or simply manifold, denoted by \mathcal{M} , without confusion.

¹⁰For this connection with experimental design, B or unions of multiple B 's are also known as *design space* in this context.

¹¹Those who know linear algebra are accustomed to viewing functions algebraically (eg, $f_1(x) + f_2(x) = (f_1 + f_2)(x)$). Those who know calculus are also accustomed to analytical viewpoints (eg, a differentiable function f has its derivative vanish at extrema). However, it is underappreciated that, underneath the static and austere looks, functions also have a vivacious geometric life. It takes shape, metamorphoses, invites imagination, and rewards those who do with ample sense of wonder and insight. As an example, consider the arctan map $(x, y) \mapsto (\arctan x, \arctan y)$ and the radial map $(x, y) \mapsto \left(\frac{x}{\sqrt{x^2 + y^2 + 1}}, \frac{y}{\sqrt{x^2 + y^2 + 1}} \right)$; algebraically and analytically they are similar (eg, both are differentiable everywhere in \mathbb{R}^2), but geometrically one sends \mathbb{R}^2 to a square and the other to a disk that is in a sense qualitatively different. Why? What does it tell us about the two functions? We humans have evolved to think superbly in geometric terms, thus we ought not waste this gift.

A good way to appreciate our new-found geometric power is to examine our diagrams again, now in the new geometric limelight, and see how they cease being cold symbols, but come to life, start dancing and put on a spectacle on the stage of modeling. We will do this next, and along the process we will mix in some key findings of the field that ground and motivate further developments.

First, we start with a brief description of the general geometric features of a typical manifold: it is *bounded* with a hierarchy of boundaries, and both the spectrum of its local metric and global widths form a geometric sequence with a constant progression in orders of magnitude. Both facts are highly nontrivial [103] and have many implications (see below). Two remarks are in order. (a) When a manifold is unbounded in a direction, we call that direction an *unbounded singular limit*,¹² which, as we will see, can sometimes be treated similarly as boundaries. It is convenient to have a name for the two types of singular limits as a whole: we call them *generalized boundaries*, denoted ∂M . (b) The sampling structure of f has been invoked to explain the geometric hierarchy in manifold widths through interpolation theory [103].

Second, as a continuous map implied by its differentiability, f now takes on a geometric life not only in its final image (that is, a manifold), but also the whole *process* from the domain to the image: it can be represented geometrically through defining a *homotopy* between two continuous maps. To do this, first embed P in Y

¹²In that direction, the determinant of the metric tensor becomes divergently singular, just like a boundary can be thought as a (bounded) singular limit where the determinant becomes vanishingly singular.

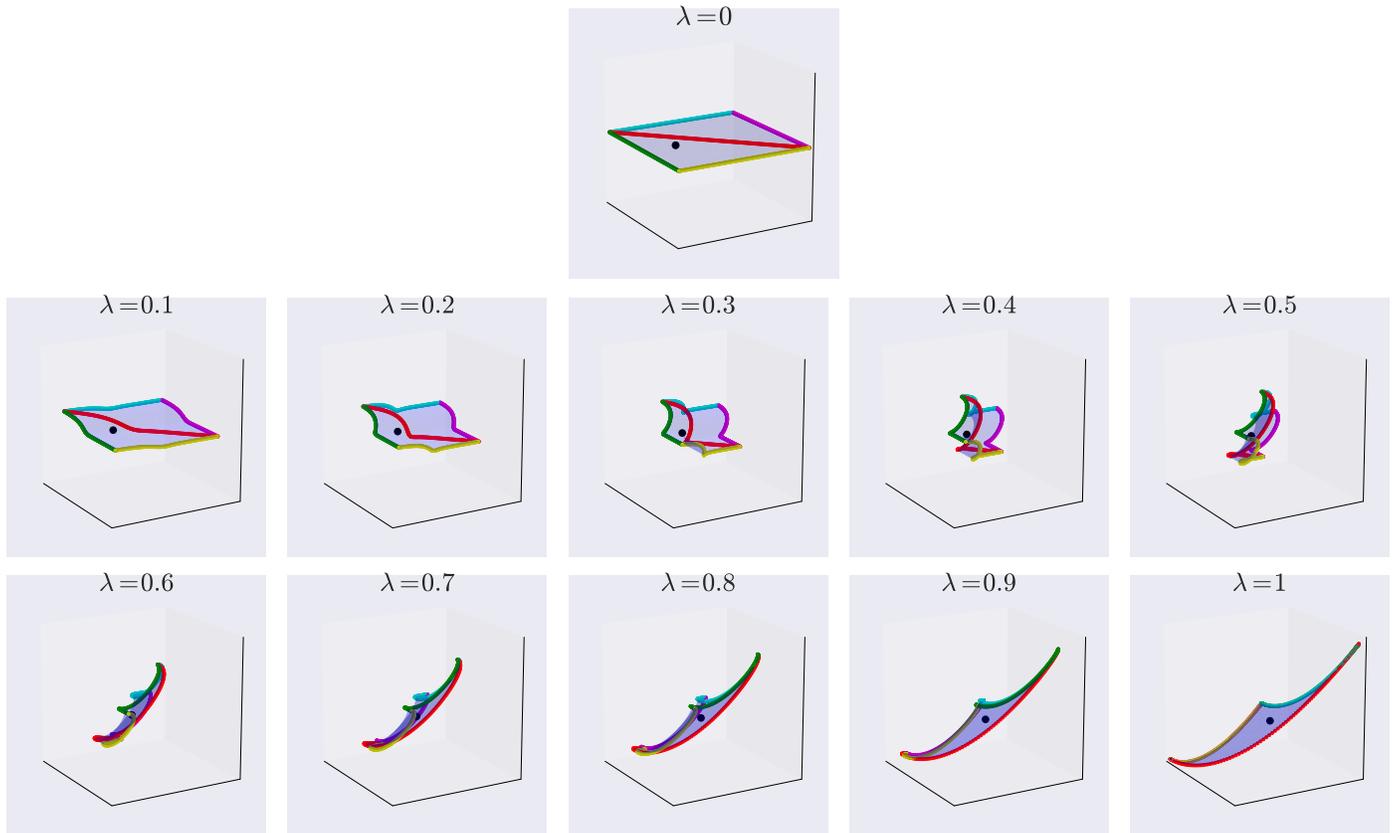


Figure 2.5: Homotopy morphs parameter space to model manifold for the `sum2exp` model. The top plot with $\lambda = 0$ shows the parameter space, with its boundaries schematically labeled (the colored edges of the square; the parameter space has its boundaries at infinity, so what is shown here can be thought as a “processed” parameter space that is compactified using, for example, the arctan map in Footnote 11). The lower rightmost plot with $\lambda = 1$ shows the model manifold, with its boundaries corresponding to those of the parameter space and, because of symmetry (Section 2.4.5), part of the parameter space interior as well (the red curve). The rest of the plots represent intermediate steps in between with $0 < \lambda < 1$. The black dot on the parameter space is mapped to a black dot on the manifold along the process.

through defining an *embedding* g , also a continuous map (eg, $g : p \mapsto (p, 0) \in Y$).¹³ Then define an “interpolating” continuous map: $h(p, \lambda) = \lambda f(p) + (1 - \lambda)g(p)$, $0 \leq \lambda \leq 1$; moving λ from 0 to 1 “morphs” P into the manifold \mathcal{M} .¹⁴ Fig. 2.5 illustrates such a morphing process for the `sum2exp` model. Note that this process can go both ways by simply moving λ from 1 to 0, and that any map along way is also differentiable with respect to p as $D_p h(p, \lambda) = \lambda Df(p) + (1 - \lambda)Dg(p)$. With this perspective, model prediction in Fig. 2.4 becomes a geometric process that sends a point in $p \in P$ through an exhilarating ride of homotopy to its final destination $f(p) \in \mathcal{M}$, and parameter estimation can be thought as homotopy in reverse action following first a projection of the data vector onto the manifold.

Third, geometrically parameter estimation in its popular least-square incarnation becomes a problem of finding the point at \mathcal{M} that has the shortest Euclidean distance to the point in Y representing data, and the geometric knowledge of \mathcal{M} described above explains common challenges and inspires improvement [103, 108]. Fig. 2.6 illustrates this perspective using the Michaelis-Menten model.

Fourth, geometrically model selection becomes comparing two different manifolds \mathcal{M} and \mathcal{M}' from two models in the same prediction space Y . How much better is one manifold in fitting *the* data than the other? How much better is one manifold in fitting *all* data than the other – that is, how different is their *expres-*

¹³Here it assumes that $\dim P < \dim Y$. With some extra work, it is possible to have the homotopy picture without this assumption.

¹⁴The morphing is unique for a given embedding, but there are infinitely many embeddings; eg, $g : p \mapsto (0, p)$ also workd.

siveness? How does increasing parameter complexity affect a manifold's expressiveness? Under what parameter conditions are they comparable? Relevant questions like these can all be readily translated into geometric inquiries. (Section 3.4.4 briefly discusses this again.)

Finally, geometrically model reduction, a kin task of model selection, becomes identifying manifold boundaries. Since typically an \mathcal{M} has a hierarchy of boundaries and its widths form a geometric sequence, approximating it by its boundaries, at least reasonably high-dimensional ones, should lose little of its expressiveness. An \mathcal{M} with the two features is sometimes likened to a ribbon, a three-dimensional object with a small width and an even smaller thickness, of which its two dimensional faces or one-dimensional edges along the long axis are good approximations. This, is the underlying idea of *manifold boundary approximation method* (MBAM) [106], one of the most exciting recent developments in information geometry. It provides a practical, elegant and tractable solution to the nagging problem looming over the exciting discoveries on "sloppiness" [17, 16, 45]: all evidence points to that the models are overparametrized, but nobody knows where the overparametrization lies or how to get rid of it. Like Google solves the "I know half of my advertising money is wasted but I don't know which half" dilemma, MBAM identifies the overparametrization and trims it away. More importantly, MBAM holds the promise of bridging microscopic details and systems behaviors, enabling us to finally understand, for example, the "relevant details" in that intracellular soup that makes a cell dance [107] (Section 1.4). Fig. 2.6 shows an application of MBAM using the Michaelis-Menten model.

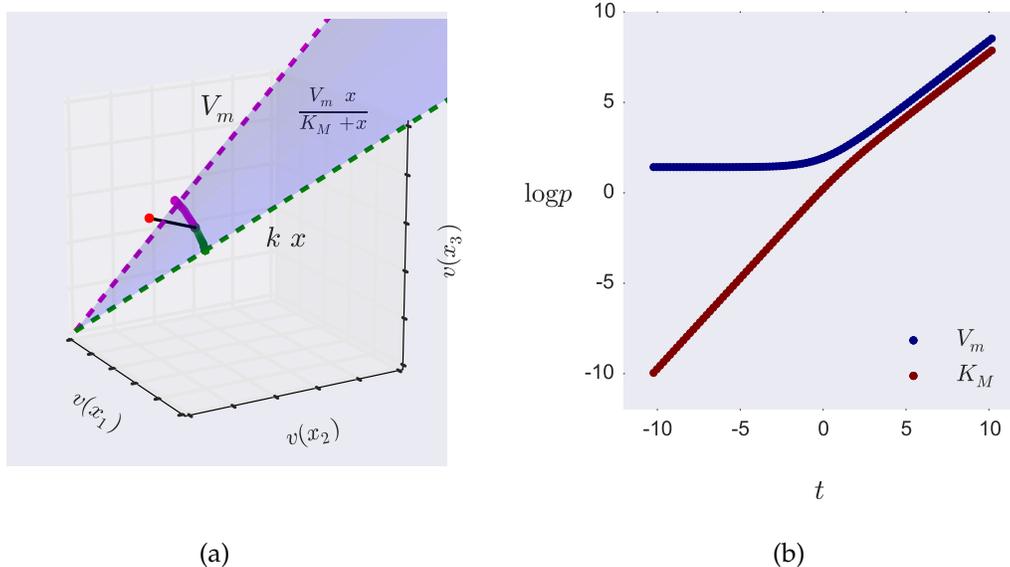


Figure 2.6: Geometry of parameter estimation and model reduction of the Michaelis-Menten model. (a) Using the sampling $X = \{x_1, x_2, x_3\}$ illustrated in Fig. 2.3, the prediction space Y is a three-dimensional Euclidean space with axes corresponding to $v(x_1)$, $v(x_2)$ and $v(x_3)$, and the model manifold \mathcal{M} is a two-dimensional surface embedded in Y (the blue surface). Geometrically, data with measurement noise corresponds to a point in Y with some deviation from the manifold (the red dot), and parameter estimation using the least-square method corresponds to finding the point on the manifold with the shortest distance to the data vector (the black line segment projects the data vector onto the manifold and finds the closest point). Geometrically, model reduction is translated by MBAM to traveling on the manifold towards boundaries (the magenta and green dashed rays) via geodesic motion (the magenta and green curves). (b) Parameter trajectories along the geodesics, which reveal parameter limits at the boundaries. One geodesic (the magenta curve) hits a boundary (the magenta dashed ray) where $K_M \rightarrow 0$ while $V_m \sim O(1)$ (the left half of the parameter trajectories; note that the parameters have been logarithmically transformed), meaning that the model $v = \frac{V_m x}{K_M + x}$ becomes $\lim_{K_M \rightarrow 0} v = V_m$; similarly, geodesic along the opposite direction (the green curve) hits the other boundary (the green dashed ray) where $V_m \rightarrow \infty$ and $K_M \rightarrow \infty$ while $k = V_m/K_M \sim O(1)$, meaning that the model becomes $\lim_{K_M \rightarrow \infty} \frac{k K_M x}{K_M + x} = k x$. In this way, MBAM finds the linear and saturate regimes semi-automatically, the two commonly-used simplifications of the Michaelis-Menten rate law.

There is one more thing to be said about MBAM's impact. It inspires a systematic examination of a manifold's boundary structure, and this leads to a new line of inquiry, namely *information topology* [105], inheriting some flavors from information geometry and at the same time having brewed many of its own.

One note on notation here. In this thesis, we often use symbol $p \rightarrow p_\partial$ to denote the limit of approaching the boundary of a manifold \mathcal{M} , denoted $\partial\mathcal{M}$. To represent the parameter limit of boundary $\partial\mathcal{M}$, we sometimes write $p_\partial(\partial\mathcal{M})$.

Starting in the next section, we will turn to information topology, on which much of this thesis is based.

2.2 From information geometry to information topology

There are two main motivations to move beyond information geometry.

The first motivation is brought about by MBAM: it is local in vision. Like a late-15th century world explorer who frantically bounds from shore to shore, it travels to one boundary after another. The sailing technology is good enough for it to travel along a straight line on earth and save curving into the wild blue yonder (that is, traveling along geodesics), and sirens sing whenever the waterscape looks ominous or unpromising (MBAM can, for example, detect if the direction it is heading into is an unbounded singular limits and it had better stop). Upon hitting a post it can enjoy a bit its local exotic culture (upon hitting a boundary MBAM

can usually decipher the boundary information and enjoys a bit its interesting behaviors), all the while under the guidance of the north star constellation (the data vector ensures MBAM exploration does not wander too far). Yet the whole expedition, with all its excitement and rewards, was marred by some deep-seated anxiety: Could I have gone to a better place than the one just visited? Is the current direction the right one? Where will it lead to? What will the place be like? *What is this world like?* Today, a 21st century world trekker perhaps does not feel far less excited in his exploration than his 15th century ancestor, but is saved from all these insecurities, thanks to one thing: a map of the world. He has one, and his ancestor did not. Just like human explorers develop all sorts of maps to guide their explorations and satisfy their curiosity, from genomes to the earth to the universe, an information geometer also needs to chart his manifold: he needs a map of its boundaries.

The second motivation comes from information geometry as a whole. Conceptually, information geometry often is concerned with the geometry of \mathcal{M} for a given fixed prediction map f ; it can say an awful lot of interesting things about a particular manifold, but any tiniest change to the manifold would require a restart of the whole undertaking. And as we have seen, a model \mathcal{M} is often able to generate a variety of B 's, and each B can admit various sampling X 's (amounting to various Y 's), perhaps realized by different experiments. That is a lot of parallel manifolds! Can we make the fruit of our effort more lasting?

It is important to recognize the tension between these two motivations. A map

has the info of both how things are globally connected and the distances between them. but the geometric info of distances would be irrelevant in a different manifold. Of course, it would be ideal to have a map for every manifold, but it may be too much to ask for under limited human and computational resources. A good balance is struck by a topological characterization, which is global and at the same time promises to be less sensitive to manifold perturbations.

Hence comes *information topology* [105], which aims to achieve the following two things:

1. Provide a topological characterization of the boundary structure of a manifold;
2. Understand how it depends on B and X .

We call such a topological characterization of a manifold's boundary structure the manifold's *topology*.

Two immediate benefits follow from the adoption of this new line of inquiry.. First, with a topological characterization of a manifold, we have *all* its reduced models. Second, since the topology of a manifold persists upon different samplings or experimental conditions, we can make progress unhindered by possible issues on data quantity or quality of the systems we study.

We take note of one practical matter: the boundary structure of a manifold is

usually described by a *Hasse diagram*, illustrated in Fig. 2.7 using the Michaelis-Menten model.

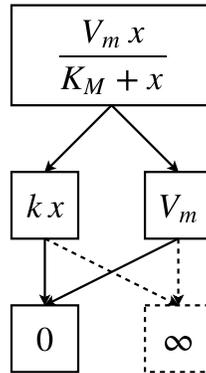


Figure 2.7: Hasse diagram of the Michaelis-Menten model. Fig. 2.6 shows that for the Michaelis-Menten model $v = \frac{V_m x}{K_M + x}$, there are two boundaries, $v = V_m$ and $v = k x$, both of which geometrically are rays emanating from the origin to infinity. Such information can be intuitively encoded in a hierarchical graph called Hasse diagram (dashed arrow and boxes represent unbounded singular limits, that is, infinity). High-dimensional model manifolds are hard to visualize, but their Hasse diagrams encoding the information of their boundary structures can be readily constructed (see Figs. 3.2, 3.4 and 3.9 for examples).

2.3 Model composition

How to get the topology of a manifold? So far there are three ways, but they either provide no guarantee of exhaustiveness or are not general.

- Parameter-space sampling
- Data-space sampling
- Models with U finite [104].

Even if we can exhaustively enumerate all the boundaries, what if the set is vast? There are hints that this is the case. Without some conceptual organization of the set, it could just be a jumbo mix. Finally, in one's geodesic explorations, certain types of simplification occur again and again, which suggests some simplicity and invites understanding.

To understand a model manifold's boundary structure, let us first ponder a bit on how models are typically developed in the first place. It is a broad question, and the following caricature is oversimplifying but useful.

First, there are models that are product of prolonged and intense investigation of some typically small systems. Examples include the exponential decay model, law of universal gravitation, Hodgkin-Huxley model and enzyme kinetics models (Table 2.1).

Second, some models in the first class are combined together to form new models of some typically large systems of which the small systems in the first class are considered components. Examples include the sum of two exponential decay model [103], three body dynamics, neural network and metabolic network models.

The first class of models can be thought as fruits, in one way or another, of centuries' *reductionism* program, and the second approach to modeling has been called "*constructionism*" [3]. While it is natural and nice to make use of the "reductionistic models" in the first class, the "constructionistic models" typically lose the elegance and sharpness of their reductionistic predecessors: they become bulky, unstable, hard to interpret, difficult to parametrize and inaccurate in predictions.¹⁵ For these reasons, constructionism has many detractors, but the ease of following it has it attract even more practitioners. It is the predominant approach in fields like systems biology; in fact, the parameter problem (Sec. 1.4), whose symptoms are reminiscent of the list above, can be thought as a systems biology manifestation of constructionism. Among the proposed solutions, we have followed information geometry and continued our journey beyond in information topology. Like MBAM, which starts from a constructionistic model and tries to meet system behaviors in the middle, here as a continuation of this quest, we similarly adopt the hierarchical dichotomy implicit in constructionistic models and articulate it in the following way.

System-component formulation

A system consists of interacting components. A system model consists of coupled component models.

¹⁵It should be apparent that different as the models they produce may seem, constructionism has considerable ideological kinship with a naive version of reductionism. They have similar faiths, just live in different eras.

The most important consequence of such an apparently self-evident formulation is that, for a certain class of system models, the formulation translates to a decomposition of their behavioral maps:

$$F = H \circ G, \tag{2.1}$$

where

- $F : p \mapsto z(u)$ is the behavioral map of a system model, which we rename *system behavioral map* in this context;
- $G : p \mapsto v(x)$ is the behavioral map of the direct product of component models, which we call the *component product behavioral map* (and we call the model *component product model*);
- $H : v(x) \mapsto z(u)$ is a map that sends component product behaviors to system behaviors, which we call the *coupling map*.

Additional definitions naturally flow out from here:

- The manifold of a system behavioral map F , and its interior, boundaries and topology are called *system manifold*, *system interior*, *system boundaries* and *system topology*, respectively.

- Likewise, for a component product behavioral map G we have *component product manifold*, *component product interior*, *component product boundaries* and *component product topology*.

When confusion is unlikely and there is no need to emphasize the product nature, we will shorten the names by skipping the word "product" and simply refer to them as component model, component behavioral map, component manifold, component interior, component boundaries and component topology.

What kind of system models admit such a decomposition? It is those system models whose parameters all lie in its component models. In this case, all parameters have to line up from the start waiting to be processed, first by G then by H , and no new parameters can join in between G and H ; geometrically, coupling simply further "morphs" the component manifold to the system manifold without adding extra complexity or dimensions. Models arising from reaction networks, typified by Eq. 1.1, are in this class: devils lie in individual reactions, and the coupling is a benign linear transformation N with fixed coefficients. Neural networks, on the other hand, can have parameters in both its component models (parameters of the activation function, if any) and its coupling (weights on the edges). There are system models where devils solely resides in the coupling, as exemplified by a common formulation of Ising model where the component models are no more than binary spins and it is the interaction parameters that parametrize system behaviors.

So far we have been frantically building abstract formalisms and creating pompous terminologies, without much of an apparent fruit; it may appear that instead of taming the “wild elephant” as promised, we have just added yet another “white elephant”. Starting in the next section, we will see how the effort here pays off handsomely, bringing us both conceptual understanding of system topology and practical recipes of dealing with structural nonidentifiability.

2.4 Features of system topology

Let us start by applying chain rule to Eq. 2.1:

$$DF(p) = DH(G(p))DG(p), \quad (2.2)$$

which, with $|\bullet|$ denoting the product of the singular values of a matrix, implies:

$$\text{If } |DG(p)| = 0, \text{ then } |DF(p)| = 0. \quad (2.3)$$

What does this statement mean? There are two possibilities, and we discuss them one by one in the next two sections.

2.4.1 Structural nonidentifiability

The first possibility is that $|DG(p)| = 0$ at a generic p . For example, consider the following component model:

$$M(u, p) = \frac{p_1 p_2 u}{p_3 + u}. \quad (2.4)$$

One can verify that at any point p , $\ker DG$ always has a nontrivial nullspace spanned by $w = \begin{pmatrix} p_2 \\ p_1 \\ 0 \end{pmatrix}$ making $|DG(p)| = 0$ for any p . Any system model F composed of it would inherit the nullspace: $DF w = 0$ for any p .

This marks our first encountering of what is known as *structural nonidentifiability*. Structural nonidentifiability¹⁶ (referred to as SN hereafter) is the symptom of a *behavioral map* where changing parameters infinitesimally in some directions leaves the model behavior unchanged (hence the parameters cannot be “identified” from model behaviors), and one mathematical manifestation of this is that the linearization of the behavioral map has a nontrivial nullspace or, equivalently, some of its singular values are zero, as shown in the example above.¹⁷ Another mathematical manifestation, which we will use extensively in the next section, is that one point on the model manifold now corresponds to a curve, a surface or an even higher-dimensional subset, depending on how many zero singular values

¹⁶It is also known as structural non-identifiability, structural unidentifiability, *a priori* nonidentifiability and algebraic degeneracy.

¹⁷Several types of structural nonidentifiability are usually distinguished, from local (any) to generic (almost all) to global (all); we focus on generic SN which we believe is the most relevant type.

there are, in parameter space; we will call such type of curves *isocurves*, such type of surfaces *isosurfaces*, etc.¹⁸ Mathematical statements aside, SN intuitively means that the behaviors that these parameters describe can in principle be faithfully described by a smaller set of parameters and there are some redundancy in our use of parameters to describe model behaviors.

In this case, statement 2.3 says that component SN corresponds to system SN. Mathematically, $|DG(p)| = 0$ for a generic $p \Rightarrow |DF(p)| = 0$ for a generic p .

It is intuitive to understand why, for the model in Eq. 2.4, moving parameters in some directions would leave the model behavior unchanged: p_1 and p_2 affect the model behavior as a product, and one can always increase p_1 while decrease p_2 to maintain their product constant, thereby leaving the model behavior unchanged; to free the model from the SN, one can simply replace p_1 and p_2 by their product. In fact, the example which is taken from enzyme kinetics, was treated just like that $\left(v = \frac{k_{cat}E_0x}{K_M + x} \text{ was changed to } v = \frac{V_m x}{K_M + x} \right)$. From this example, one may get the impression that SN is no more than a manifestation of modelers' occasional carelessness, and whenever it arises, some careful inspection should quickly fix it up. This impression is generally true for component models, whose size usually comes small, form interpretable, and inspection can go a long way. However, for system models, this impression is far from being true, owing to another way in which system SN can arise.

¹⁸Isocurves are also known as level curves, contour lines and isolines; isosurfaces are also known as level surfaces. They are one- or two-dimensional instances of the general concept *level set*.

A system model has SN when $|DF(p)| = 0$ for a generic p and Eq. 2.2 provides a factoring of DF : this suggests that aside from the possibility of $DG(p)$ having nontrivial nullspace, there is the other possibility of $DH(G(p))$ having nontrivial nullspace *and* part of the image of $DG(p)$ somehow falling onto it; mathematically, $\text{im } DG(p) \cap \ker DH(G(p)) \neq \emptyset$ for a generic p , or equivalently, with (\bullet, \bullet) denoting the matrix formed by joining two existing matrices row-wise, $\left| \left(DG(p), \ker DH(G(p)) \right) \right| = 0$ for a generic p .

This second type of system SN is far nastier than the first type. First, it is not existent in component models, hence one cannot trace it to the simple component models; it owes its whole existence to the constructionism program, under which the chosen G and H conspire to plot havoc on the resulting F . Second, this type of SN tends to be *global* in scope, involving multiple component models, which makes its removal especially challenging. For these two reasons, and the aforementioned reason that component models can be thought as largely immune from SN, we think that *SN is exclusively a system model's blight*. The appearance of SN is the *first* feature of system topology.

2.4.2 Compositionality

The second possibility to have $|DG(p)| = 0$ is at component boundaries:

$\lim_{p \rightarrow p_\partial} |DG(p)| = 0$. Then the statement 2.3 says that $|DF(p)|$ vanishes at component boundaries: $\lim_{p \rightarrow p_\partial} |DG(p)| = 0 \Rightarrow \lim_{p \rightarrow p_\partial} |DF(p)| = 0$. Its interpretation takes some care.

The previous discussion on SN tells us that if a system model has SN, an interior point on the system manifold would correspond to a curve or a higher-dimensional subset of the parameter space. If such a subset includes a component boundary, then the component boundary would correspond to the interior point on the system manifold.

On the other hand, if a system model does not have SN, then $|DF(p)| = 0$ implies that p is at its boundary, and $\lim_{p \rightarrow p_0} |DG(p)| = 0 \Rightarrow \lim_{p \rightarrow p_0} |DF(p)| = 0$ implies that component boundaries correspond to system boundaries.

To summarize the two cases, *component boundaries correspond to system boundaries in models without structural nonidentifiability, and correspond to either system boundaries or system interior in models with structural nonidentifiability.*

This unassuming statement has important implications. Most of the models out there do not have or have been cured of SN, so we can focus on the first part for a moment. It tells us the simple fact that when a component becomes simpler, so does the system, and describes what we call the **compositionality** feature of system topology: component boundaries become system boundaries through composition, and for the set of system boundaries that we want to get, a portion of it comes from components. This provides significant simplification of the game: component models are much simpler to study and we know them much better; at the same time, to reduce component models *is* to reduce system models.

As we shall see, for some system models, component boundaries dominate.

Also, certain non-component boundaries would only appear once the component models have been simplified to a certain level. These two reasons add further significance to component models and their separate topological characterizations, as we will do for metabolic models in Sec. 3.2

For the second part of the statement, it turns out important in our resolving SN, as we will see in Sec. 2.5.

2.4.3 Emergent boundaries

If all system boundaries are component boundaries, then "systems science" might as well be renamed "additive component science", and reductionism shall rule the world. From a manifold boundary point of view, the way a system model can have its own boundaries and be truly "larger than its parts" is through the following mechanism:

$$\text{As } p \rightarrow p_0, |DG(p)| \not\rightarrow 0 \text{ but } \left| (DG(p), \ker DH(G(p))) \right| \rightarrow 0.$$

In words, as p goes to some limit, the component map does not become singular, but "falls into the crack" where the "crack" is the nullspace of the coupling map. This reminds us of how SN comes into being: the component map has no SN, but falls into the crack introduced by the coupling map.

We call this type of system boundaries **emergent boundaries**. They arguably capture the spirit of systems science and should be its focus.

Curiously, an emergent boundary can be local, that is, involving only one component. The only class of examples that have been found so far are component unbounded singular limits (ie, some component behaviors go to infinity). It is conjectured here that this class constitute the only type of local emergent boundaries.

2.4.4 Combinatoriality

We call the fourth feature of system topology **combinatoriality**, which states that *myriad system boundaries correspond to combinatorial explosion of a few atomic reductions*. An example best illustrates this feature.

Fig. 2.8 shows the Hasse diagram of one-substrate-one-product Michaelis-Menten rate law. It is one of the simplest rate laws, yet it has 25 boundaries. Consider a system model for a pathway consisting of 10 such simple reactions. The component product topology of such a system model would have about 100 trillion (10^{14}) boundaries.

However, despite the astronomical number, they *all* can be described as combinations of *three* what we call *atomic reductions*, discernible from the diagram of a single reaction in Fig. 2.8:¹⁹

¹⁹The set of atomic reductions for Michaelis-Menten rate laws is well-defined enough that it is possible to automate its boundary enumeration process for a give rate law in arbitrary numbers of substrates or products.

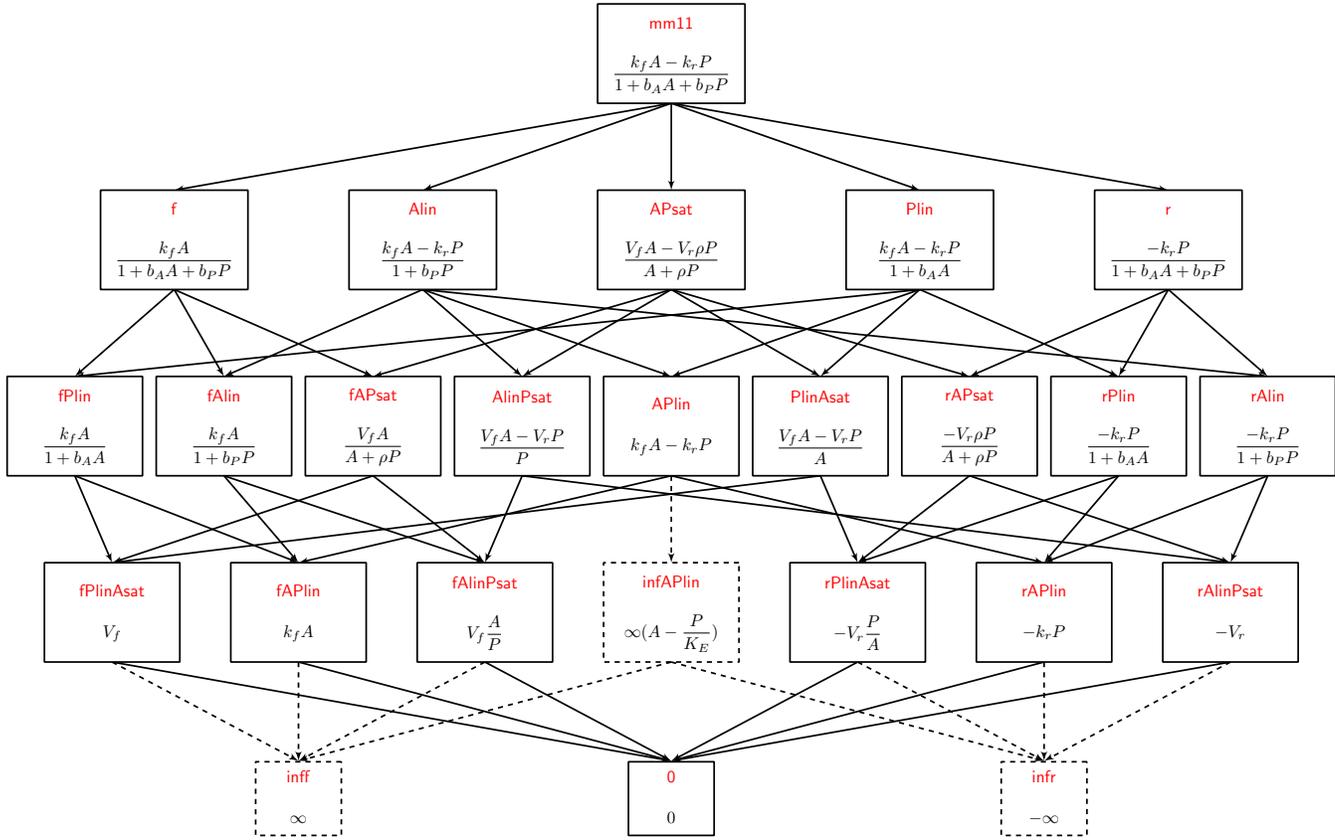


Figure 2.8: Hasse diagram of the topology of rate law $mm11$. The names are given in red which contain information about the atomic reductions involved: for example, f/r means forward/reverse unidirectionalization, $Alin$ means linearization of substrate A, and $APsat$ means saturation of both A and P. Functional forms are given in black and unbounded singular limits are boxed in dashed lines.

1. *Unidirectionalization*, in which the reaction becomes irreversible in one direction.
2. *Linearization*, in which one reactant goes to the linear regime of Michaelis-Menten kinetics.
3. *Multisaturation*, in which multiple reactants go to the saturation regime of Michaelis-Menten kinetics.

Moreover, many system models have considerable homogeneity in their component models, producing significant overlap in the set of "atomic reductions" between component models. This simplicity makes a big part of the vast set of system boundaries comprehensible, if not mundane.

2.4.5 Symmetry

Lastly, many models have symmetry, which can be used to simplify our understanding of their topology.

In [104], it shows that *p-symmetry likely leads to boundaries*.²⁰ Below is a re-derivation using the present notation of a model map ??.

p-symmetry means that a transformation of *p* into $\phi(p)$ does not change model

²⁰By "likely", we mean that it seems unlikely for a linearized symmetry transformation to have both its eigenvalues one. For a two-dimensional linear symmetry transformation, only a shear mapping does it.

predictions: $M(u, \phi(p)) = M(u, p)$. Denoting the fixed points of the transformation as p^* : $\phi(p^*) = p^*$, we have the following linearization of the model map at p^* :

$$D_p M(u, p^*) = D_p M(u, \phi(p^*)) = D_p M(u, p^*) D\phi(p^*).$$

Multiply both sides by an eigenvector of $D\phi(p^*)$, v_i , with an associated eigenvalue we denote λ_i :

$$D_p M(u, p^*) v_i = D_p M(u, p^*) D\phi(p^*) v_i = \lambda_i D_p M(u, p^*) v_i,$$

which implies that if $\lambda_i \neq 1$, then $D_p M(u, p^*) v_i = 0$. In words, at a fixed point of a symmetry transformation in p , any eigenvalue that is not equal to one would introduce a boundary with its associated vector pointing its direction.

One can identify a few more types of symmetry, and we describe two of them below.

Second, *p-antisymmetry leads to singularity*.

p-antisymmetry means that a transformation of p into $\phi(p)$ only changes the sign of model predictions: $M(u, \phi(p)) = -M(u, p)$. At fixed points of ϕ , p^* , we have:

$$M(u, \phi(p^*)) = M(u, p^*) = -M(u, p^*) \Rightarrow M(u, p^*) = 0$$

.

Given that the set of fixed points $\{p^* \mid \phi(p^*) = p^*\}$ generically constitutes a subset *in the interior* of P , $M(u, p^*) = 0$ means that M maps the whole set of fixed points to a single point, which, when combined with the continuity argument, represents a singularity. Geometrically, it "pinches" the manifold along a subset to a point.

Example: `diff2exp`

$M(u, p) = e^{-p_1 u} - e^{-p_2 u}$ has an antisymmetry in p : let $\phi(p) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} p$, and we have $M(u, \phi(p)) = -M(u, p)$; the set of fixed points $\{p \mid p_1 = p_2\}$ is mapped to the origin and forms a singularity.

Third, (p, u) -*(anti)symmetry leads to manifold symmetry*.

(p, u) -*(anti)symmetry* means that having symmetry transformations applied to both p and u does not change model prediction, up to a sign: $M(\psi(u), \phi(p)) = \pm M(u, p)$. We have:

$$\begin{aligned}
 & \lim_{p \rightarrow p_\partial} |D_p M(u, p)| = 0 \\
 \Rightarrow & \lim_{p \rightarrow p_\partial} |D_p M(\psi(u), \phi(p)) D\phi(p)| = 0 \\
 \Rightarrow & \lim_{p \rightarrow p_\partial} |D_p M(\psi(u), \phi(p))| |D\phi(p)| = 0, \\
 \Rightarrow & \lim_{p \rightarrow p_\partial} |D_p M(\psi(u), \phi(p))| = 0 \\
 \Rightarrow & \lim_{p \rightarrow \phi^{-1}(p_\partial)} |D_p M(u', p)| = 0
 \end{aligned}$$

where we have used the assumption that $\lim_{p \rightarrow p_\partial} |D\phi(p)| \neq 0$. The equations say that when $p \rightarrow p_\partial$ is a boundary, then by symmetry $p \rightarrow \phi^{-1}(p_\partial)$ is also a boundary.

Example: mm11

$M(u, p) = \frac{p_1 u_1 - p_2 u_2}{1 + p_3 u_1 + p_4 u_2}$ has an antisymmetry in (u, p) : let $\psi \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} u_2 \\ u_1 \end{pmatrix}$ and $\psi \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix} = \begin{pmatrix} p_2 \\ p_1 \\ p_4 \\ p_3 \end{pmatrix}$, and we have $M(\psi(u), \phi(p)) = -M(u, p)$. This symmetry explains the symmetric Hasse diagram in Fig. 2.8.

We have described five salient features of system topology: the appearance of structural nonidentifiability and emergent boundaries, compositionality, combinatoriality and symmetry. They interact: for example, symmetry in component models contributes to its combinatoriality, which is then manifested in system topology through compositionality, and emergent boundaries and symmetry can also lead to combinatoriality. A more accurate way to think about some of these features is that they describe different aspects of same regularity that is hidden in system topology, and they together help us organize our understanding of system topology. What about structural nonidentifiability? It is generally considered to be a major roadblock for any modeling practice, only the removal of which can allow various modeling machineries to pass and information to flow [30]. In the next section we are going for a long ride in the terrain of SN, marking a series of posts represented by examples from metabolism, and peaking at a method inspired by information topology that is in principle as general for resolving SN as MBAM for model reduction.

2.5 Information topology helps resolve structural nonidentifiability

2.5.1 Introduction

First off, a basic question: how to detect structural nonidentifiability? Sophisticated methods abound in the vast literature on SN [25]. We use the most naive method, for its simplicity, intuitiveness and generality: impose a sampling X on the behavioral space B and convert it to a finite-dimensional Euclidean space Y ; evaluate the jacobian of the resulting f at a sample of P , choose a typical p and plot out its *spectrum*, ie, the list of its singular values. This method usually, but not always, works well,²¹ and applying it to a few models produces the following figure.

Fig. 2.9 shows the general phenomenology of our method of detecting SN: there is a noticeable gap between some singular values close to either the "machine epsilon"²² or a preset numerical tolerance, and the rest. Such close-to-zero singular values are, because of the gap, likely to be actually zero. The number of zero singular values, then, represents the number of independent directions a

²¹When it does not, it is typically due to either *undersampling* that brings *practical nonidentifiability* "down" the spectrum or numerical noise that moves structural nonidentifiability "up" the spectrum. In this case, making the sampling more extensive or numerical tolerance more stringent often helps. Another remedy is to find a *surrogate map* f_1 that is known to have the same rank as f (eg, $f = f_2 \circ f_1$ with Df_2 always full rank), but whose jacobian can be evaluated with greater precision, and numerically investigate f_1 .

²²Around 10^{-16} for a double-precision, or 64-bit, representation of floating-point numbers.

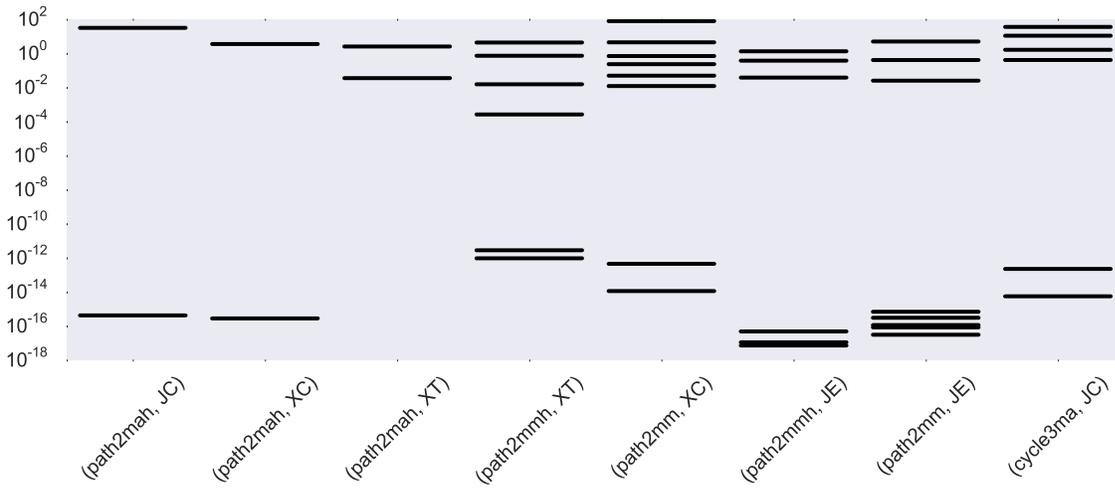


Figure 2.9: Singular value spectra of a few behavioral maps from metabolic modeling. All behavioral maps have structural non-identifiability, manifested by the gap in the spectra, except $(\text{path2mah}, \text{XT})$. $(\text{path2mmh}, \text{XT})$ has some of its singular values close not to “machine epsilon”, but to the preset numerical tolerance in integration (10^{-12}).

parameter can move without affecting model behaviors, a measure of the *degree* of SN; we denote it by d . When $d = 1$, there is one independent direction, and moving along that direction traces out a curve, namely, an isocurve; when $d = 2$, an isosurface, etc. With n denoting the number of parameters in a model, $n - d$ represents the actual degrees of freedom in model behaviors; we call it the *rank* of a prediction map f , and denote it r . For example, in the fifth column in Fig. 2.9, $(\text{path2mm}, \text{XC})$ has $(n, r, d) = (8, 6, 2)$.

2.5.2 Some examples

In this section we will examine a few examples from metabolism, which provides a glimpse of the kind of SN one often encounters and what one typically can do about it. For each of the first five examples, we outline its individual characteristics; major effort will be spent on the sixth one, which marks our take-off point towards a general framework.

Example 1: (path2mah, JC)

- $(n, r, d) = (2, 1, 1)$
- Inspecting the formula $J(C, k) = \frac{k_1 k_2}{k_1 + k_2}(C_1 - C_2)$ derived in Section 1.6 immediately suggests a reparametrization: $\tilde{k} = \frac{k_1 k_2}{k_1 + k_2}$, and removes the SN: $J(C, \tilde{k}) = \tilde{k}(C_1 - C_2)$.
- Our resolution of the SN is global in scope as it involves the parameters from both component models.

Example 2: (path2mah, XC)

- $(n, r, d) = (2, 1, 1)$
- Inspecting the formula $X(C, k) = \frac{k_1 C_1 + k_2 C_2}{k_1 + k_2}$ derived in Section 1.6 suggests that it is only the ratio between k_1 and k_2 that affects the model behavior: $r = \frac{k_2}{k_1}$, and reparametrization using the new ratio parameter removes the SN: $X(C, k) = \frac{C_1 + r C_2}{1 + r}$.

- Our resolution of the SN is again global in scope.
- Mathematically, replacing k_1 and k_2 with their ratio r can be interpreted in two ways. The first way is conventional: we combine two parameters into one; it is global in scope (we have to simultaneously think about both component models), and has the reparametrization formula $(k_1, k_2) \mapsto r$. The second way is less conventional but mathematically equivalent: we fix the value of k_1 to be 1; it is local in scope (we can think only about the first component model), and has the reparametrization formula $(k_1, k_2) \mapsto (1, k_2)$.²³ We call the second reparametrization *currying reparametrization*.²⁴ Note that both having global SN, $(\text{path2mah}, \text{XC})$ admits a currying reparametrization while $(\text{path2mah}, \text{JC})$ does not.

Example 3: $(\text{path2mah}, \text{XT})$

- $(n, r, d) = (2, 2, 0)$
- Inspecting the solution derived in Section 1.6, $X(t, k) = X(\infty) + (X(0) - X(\infty))e^{-(k_1+k_2)t}$ where $X(\infty) = \frac{k_1 C_1 + k_2 C_2}{k_1 + k_2}$ is the steady-state solution of X , we can understand the lack of SN in this case: $X(t, k)$ depends on both the ratio of k through $X(\infty)$ and the scale of k through the transient term $e^{-(k_1+k_2)t}$. While each of $(\text{path2mah}, \text{XC})$ and $(\text{path2mah}, \text{JC})$ probes only one

²³The two forms are only mathematically equivalent; physically, fixing the value of k_1 implies a change in meaning of k_2 to r .

²⁴The procedure of obtaining new functions by fixing some input values of a multi-input function is called currying.

aspect of the parameters and introduces one degree of SN, (`path2mah`, `XT`) probes both and leaves no room for SN.

Example 4: (`path2mmh`, `XT`)

- $(n, r, d) = (6, 4, 2)$
- After careful inspection, we find out that the SN arises from the fact that C_1 and C_2 are treated as constants in this model: $\frac{dX}{dt} = v_1 - v_2$, where $v_1 = \frac{k_1(C_1 - X)}{1 + b_1C_1 + b_2X}$ and $v_2 = \frac{k_2(X - C_2)}{1 + b_3X + b_4C_2}$. C_1 and C_2 can be absorbed into the parameters, which can then be rearranged to reduce the total number of parameters while maintaining the model equivalent: $v_1 = \frac{\frac{k_1}{1+b_1C_1}(C_1 - X)}{1 + \frac{b_2}{1+b_1C_1}X} = \frac{k'_1(C_1 - X)}{1 + b'_2X}$ and $v_2 = \frac{\frac{k_2}{1+b_4C_2}(X - C_2)}{1 + \frac{b_3}{1+b_4C_2}X} = \frac{k'_2(X - C_2)}{1 + b'_3X}$.
- Note that here, again, while it is conventional to have the reparametrization formula as $(k_1, b_1, b_2, k_2, b_3, b_4) \mapsto (k'_1, b'_2, k'_2, b'_3)$, a currying interpretation is also possible, thanks to the form of rate laws after parameter rearrangement: $(k_1, b_1, b_2, k_2, b_3, b_4) \mapsto (k_1, 0, b_2, k_2, b_3, 0)$. There is a difference between the currying reparametrization here and the one in (`path2mah`, `XC`): here $b_1 = 0$ lies in the boundary of the domain of b_1 , while in (`path2mah`, `XC`) $k_1 = 1$ lies in the interior of the domain of k_1 .

Example 5: (`cycle3ma`, `JC`)

For this case, $(n, r, d) = (6, 4, 2)$. It is an example where SN yields to serious mathematical effort but few alternative means.

The model is defined as:

$$\begin{cases} v_1 = v_2 \\ v_2 = v_3, \end{cases}$$

where $v_1 = k_1 C_1 X_1 - k_2 X_2^2$, $v_2 = k_3 C_2 X_2 - k_4 X_1$ and $v_3 = k_5 X_2 - k_6 C_3$. We can solve the two equations and get the model map:

$$J(C, k) = \frac{k_5}{2k_2 k_4} \left(k_1 k_3 C_1 C_2 - k_1 k_5 C_1 - k_4 k_5 + \sqrt{(k_1 k_3 C_1 C_2 - k_1 k_5 C_1 - k_4 k_5)^2 + 4k_2 k_4 (k_1 k_6 C_1 C_3 + k_4 k_6 C_3)} \right) - k_6 C_3 \quad (2.5)$$

Inspection tells us that it is the following parameter combinations that enter the map: $\left(\frac{135}{24}, \frac{155}{24}, \frac{455}{24}, \frac{124556}{2424}, \frac{244556}{2424}, 6 \right)$, where 1 stands for k_1 , 2 for k_2 , etc. Upon simplification and further inspection we can single out the following four parameter combinations, which we note is a global reparametrization:

$$q = \begin{pmatrix} \frac{k_3}{k_5} \\ \frac{k_1}{k_4} \\ \frac{k_4}{k_5} \\ \frac{k_2}{k_6} \end{pmatrix}. \quad (2.6)$$

So far, we have sorted out the SN in all models we examine, through a combination of mathematical derivation, insight and inspection. But these ad hoc meth-

ods typically do not go very far: mathematics quickly gets inscrutable or simply intractable, insight mostly comes from past experience with solved cases which we often lack and inspection is far too a fortuitous business to count on. The next example illustrates such a situation.

Example 6: (path2mm, XC)

As we have noted before, in this case $(n, r, d) = (8, 6, 2)$.

It is natural to start by attempting to solve for X like in (path2mah, XC), which yields the following quadratic equation:

$$\begin{aligned} (k_2b_3 + k_3b_2)X^2 + ((k_3b_1 - k_1b_3)C_1 + (k_2b_4 - k_4b_2)C_2 + (k_2 + k_3))X \\ - ((k_1b_4 + k_4b_1)C_1C_2 + k_1C_1 + k_4C_2) = 0 \end{aligned} \quad (2.7)$$

Inspection proves unhelpful, so we resort to other means. We note that Eq. 2.7 is a quadratic equation thanks to that the rate laws are rational functions. In general, rational rate laws always give rise to some polynomial equations of which steady-state metabolite concentrations X are the roots. This reasoning inspires us to look at the model in the following light:

- Eq. 2.7 represents a *nested* polynomial in the following sense: first, it is a polynomial in X , with coefficients c ; second, c are polynomials themselves, in C , with coefficients q which are functions of p .

- In the reverse direction, p determines model behavior $X(C)$ in the following way: first, it is mapped to q ; second, q and C are mapped to c ; third, c is mapped to X .

Symbolically, we have decomposed the model map $(p, C) \mapsto X$ into three parts:
 $p \mapsto q$, $(q, C) \mapsto c$ and $c \mapsto X$.

Casting Eq. 2.7 in such a formalism should make things less abstract:

$$1. \quad p = \begin{pmatrix} k_1 \\ k_2 \\ b_1 \\ b_2 \\ k_3 \\ k_4 \\ b_3 \\ b_4 \end{pmatrix} \text{ is mapped to } q = \begin{pmatrix} k_2 b_3 + k_3 b_2 \\ k_3 b_1 - k_1 b_3 \\ k_2 b_4 - k_4 b_2 \\ k_2 + k_3 \\ k_1 b_4 + k_4 b_1 \\ k_1 \\ k_4 \end{pmatrix},$$

the coefficients of C_1 and C_2 in the nested polynomials. Note that p is eight-dimensional while q is seven.

2. (q, C) , where $C = \begin{pmatrix} C_1 \\ C_2 \end{pmatrix}$, is mapped to

$$c = \begin{pmatrix} k_2 b_3 + k_3 b_2 \\ (k_3 b_1 - k_1 b_3)C_1 + (k_2 b_4 - k_4 b_2)C_2 + (k_2 + k_3) \\ (k_1 b_4 + k_4 b_1)C_1 C_2 + k_1 C_1 + k_4 C_2 \end{pmatrix},$$

the coefficients of X in Eq. 2.7.

3. c is mapped to X through root-solving Eq. 2.7.

What is the point of having all these symbolic jugglings? In this specific case,

the symbolic jugglings help us locate the SN through factoring its action: anything p does has to be done through q , and with p eight-dimensional and q seven, q encodes one degree of SN. In fact, we can do better and find out the other degree as well: we can normalize Eq. 2.7 by its leading coefficient $k_1b_3 + k_2b_2$ and reduce the dimension of q down to six:

$$q = \begin{pmatrix} (k_3b_1 - k_1b_3)/D \\ (k_2b_4 - k_4b_2)/D \\ (k_2 + k_3)/D \\ (k_1b_4 + k_4b_1)/D \\ k_1/D \\ k_4/D \end{pmatrix}, \quad (2.8)$$

where $D = k_2b_3 + k_3b_2$. In this way we resolve the SN and identify a reparametrization. More broadly speaking, the kind of symbolic jugglings represent a divide-and-conquer way of thinking and decompose an impenetrable behemoth into more comprehensible pieces. We have done it here, in the system-component formulation in Sec. 2.3, and we will do it again in the next chapter.

But there is something unsatisfying about our resolutions of SN here and in Example 5: what do those knobs of q mean? With the original parametrization, problematic as it is, at least we know the parameters well; the new parameters, on the other hand, look rather unfamiliar, with ratios upon sums upon products of the original ones. We now risk getting caught in a Faustian limbo where we exercise the SN at the cost of knowing where we are. One key reason for such a lack of interpretability in these two q 's is that they are global, mixing up the parameters from different component models. Yet the mathematics of reparametriza-

tion suggests that there is nothing to be self-pitying about here: in the space of all reparametrizations, local ones are more of the exceptions than the norms, for *given any local reparametrization, one can always apply a global map on it to mix things up and turn it into a global one*. The real question, then, should be: does there exist a local reparametrization? Let us try again.

The example of $(\text{path2mah}, \text{XC})$ provides the first hint of how to wreak a local crack on the present case: ²⁵ the scale of the two reaction rates does not affect steady-state concentration and only their ratio matters. Holding to our hope of staying local, we apply the currying reparametrization on the model, and set $k_1 = 1$.

For the remaining degree of SN, we try one thing: since $d = 1$ and the existence of isocurves provide well-defined trajectories of motion in the parameter space, let us follow them and see if there is any pattern. There are no *a priori* reasons to believe that there should be a pattern, as the genericity of global reparametrization suggested above should have a typical isocurve involve all parameters, making its interpretation just as hard as the q reparametrization above. However, in this case, we do have a pleasant surprise.

Fig. 2.10 plots such an isocurve, and it does have a pattern, a strikingly simple one: the isocurve goes to component boundaries in the limit, in this case, $b_3 \rightarrow 0$ and $b_4 \rightarrow 0$. What does this mean? We know that a whole isocurve is mapped to an interior point on the system manifold due to SN, so as far as representing

²⁵Again, experience is a major source of insight.

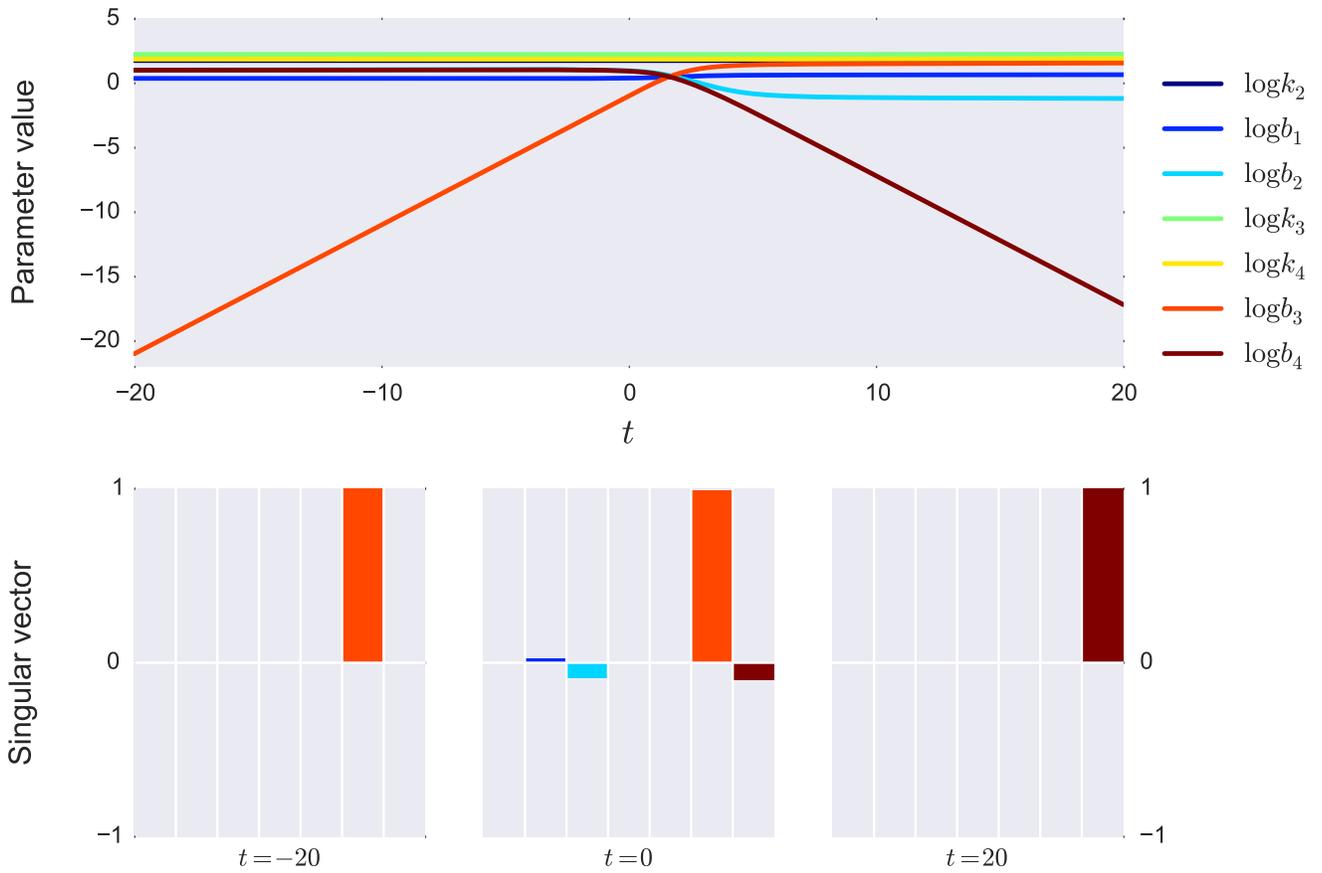


Figure 2.10: An isocurve with its singular vector signature of $(\text{path2mah}, \text{XC})$ with $k_1 = 1$. The time information along the x-axis comes from the fact that we trace out the isocurve using geodesic motion with a constant speed in parameter space.

that point is concerned, any point along the isocurve is *sufficient and equivalent*. That the isocurve goes to component boundaries in the limit means that, although one cannot identify a point on *component manifold* by a point on system manifold, which is the very meaning of structural nonidentifiability (assuming the lack of SN for component models), one *can* identify a point on *component boundaries* by a point on system manifold. In other words, as far as composing a system model is concerned, some "boundary component models" are *just as expressive* as the component models themselves; we have transformed system SN resolution to component model reduction. This is remarkable. As we have seen, system SN is often challenging to resolve and its resolution, if global, can be difficult to interpret; on the other hand, as we have also seen, component models are usually easy to reduce and their reduction, by definition local, is straightforward to interpret: the simplification brought by the solution is enormous. Of course, there is no guarantee that this type of solution exists; what we have here is, when it does, a way to find it out.

The specific solution suggests two general recipes:

1. When $d = 1$, plot isocurves and see if they exhibit simple behaviors.
2. When there is a characterization of the reduced models of component models, construct the system model using the reduced component models and see if the jacobian of the resulting system model f_r, Df_r , has the appropriate rank.

The first recipe works even when we know little about reduced component models, but is confined to $d = 1$, as when $d > 1$ the level set in parameter space of a point on the system manifold would be an isosurface or higher-dimensional analogue, where we cease having a well-defined trajectory to follow and a typical curve lying in that level set represents the joint interaction of more than one SN, making any interpretation difficult. The second recipe works for any d , but requires some knowledge of reduced component models.

To complete our resolution of SN for the case of $(\text{path2mm}, \text{XC})$, we choose a reduced component model; since both $b_3 \rightarrow 0$ and $b_4 \rightarrow 0$ lie on the same curve, it does not matter which one to choose. We choose $b_4 \rightarrow 0$, and now have a system model, composed of an mm11 with $k_1 = 1$ and an mm11 with $b_4 = 0$, which is free of SN: $|Df_r| \neq 0$. But is that it?

Mathematically, $|Df_r| \neq 0$ implies local *injectivity*: two different nearby p 's are sent to different y 's. Having $|Df_r| \neq 0$ for a generic p implies generic injectivity, which geometrically means that a k -dimensional parameter space morphs to a k -dimensional model manifold without being flattened and losing dimension: a square morphs to a sphere; a cube morphs to a ball, but not a sphere. It is exactly this lack of injectivity that defines SN: locally, the same y corresponds to a set of different p 's and cannot identify a single one; generically, a cube can first be flattened to a square, which then morphs to a sphere. What $|Df_r| \neq 0$ does not imply is *surjectivity*: the image of f_r covers the whole manifold one starts with. When we resolve SN, injectivity is what we lack and strive for, and surjectivity is

what we need to ensure. What if in an SN resolution, we identify a sphere with half of a square and turn it to a hemisphere? Yes, it is now identifiable, but we lose half of the elephant.

2.5.3 Surjectivity analysis

We can apply three different tests of surjectivity to our $(k_1 = 1, b_4 = 0)$ resolution of $(\text{path2mm}, \text{XC})$.

The first method, which we call *sampling-fitting*, is a direct application of our intuitive reasoning above: if a reduced map f_r is surjective and covers the whole image of f , \mathcal{M} , then for any point y sampled on \mathcal{M} , f_r should be able to be fit to y with zero cost; if f_r is not surjective and misses part of \mathcal{M} , then any y sampled in the missed part would result in a positive cost. The method is simple and general, and its idea applicable to a variety of settings; although it depends on stochastic sampling and as such sometimes takes considerable computation while does not guarantee a definitive answer, in most cases it does constitute a quick and easy way to glean some information.

Applying the method to our case yields Fig. 2.11, which shows three histograms. The first one (a) plots the cost distribution of f_r with respect to our sampling of y on \mathcal{M} , and it has a notable feature: the distribution falls into two groups with a wild gap in between; the group on the left take on exceedingly small values (on the order of 10^{-25}), which, given the finite precision a computer

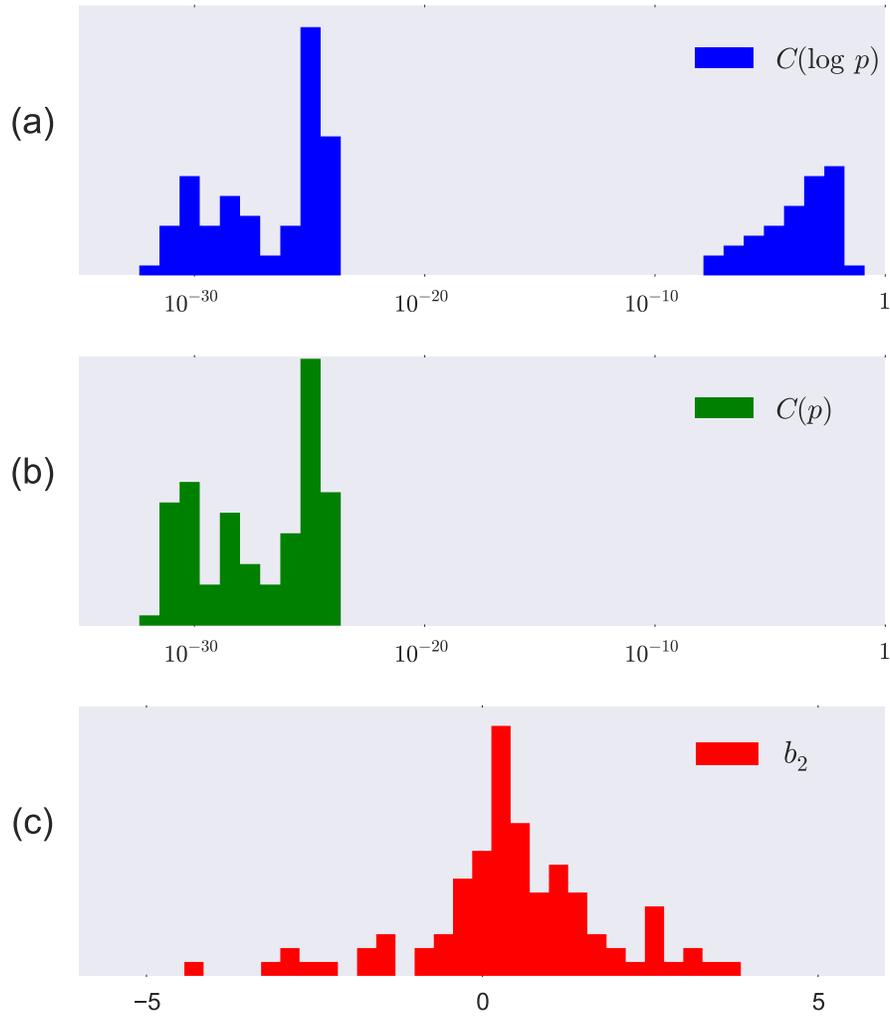


Figure 2.11: Applying sampling-fitting to the $(k_1 = 1, b_4 = 0)$ resolution of $(\text{path2mm}, \text{XC})$. In a sample of 100 points, the blue histogram (a) shows the cost distribution when the fitting is done in log parameters, thereby enforcing positivity; the green histogram (b) shows the cost distribution when the fitting is done in bare parameters without log -transformation, thereby allowing parameters to take on negative values; the red histogram (c) shows the distribution of the fitted value of parameter b_2 , which, upon inspection, is found to be the only parameter ever taking negative values in the fit.

can represent floating-point numbers, can be taken as zero; the group on the right, on the other hand, take on values much larger than both the first group and machine precision, signs that they are not zero. This is the first evidence that in our resolution, indeed we have lost part of the elephant.

Further experimentation shows that, curiously, enlarging the domain of f_r , from its usual positive real numbers to all real numbers, salvages the situation: now the cost is all zero (histogram (b)). Inspecting the fitted parameter values further shows that only one parameter, b_2 , is needed to take negative values to bring the cost to zero (histogram (c)).

This is interesting: certainly the domain of f is only the positive orthant of \mathbb{R}^n ; having a different map f_r acting on part of that domain loses part of the image; the lost part is exactly the same as the image under f_r of another orthant of \mathbb{R}^n that allows negative values along one dimension. It so happens that it is possible to derive the reparametrization formula in this case, which sheds light on the curious facts here.

As we noted in Eq. 2.8, p comes into play through q . Now that we have changed f to f_r (by setting $b_4 = 0$), we have also changed p to p_r and q to q_r . In order for f_r to mimic the behaviors of f , q_r has to be able to mimic the behaviors of q ; that is, for whatever value q takes by varying p , q_r has to be able to take the value as well by varying p_r . Remaining p_r to p' and q_r to q' for a moment here for readability, we have $q(p) = q'(p')$, which translates to the following system of equations in p' :

$$\frac{1}{D} \begin{pmatrix} k_3 b_1 - b_3 \\ k_2 b_4 - k_4 b_2 \\ k_2 + k_3 \\ b_4 + k_4 b_1 \\ 1 \\ k_4 \end{pmatrix} = \frac{1}{D'} \begin{pmatrix} k'_3 b'_1 - b'_3 \\ -k'_4 b'_2 \\ k'_2 + k'_3 \\ k'_4 b'_1 \\ 1 \\ k'_4 \end{pmatrix}, \quad (2.9)$$

where $D = k_2 b_3 + k_3 b_2$ and $D' = k'_2 b'_3 + k'_3 b'_2$.

With six equations and six unknowns $(k'_2, k'_3, k'_4, b'_1, b'_2, b'_3)$, one can solve the system and the solution reads:

$$\begin{pmatrix} k'_2 \\ k'_3 \\ k'_4 \\ b'_1 \\ b'_2 \\ b'_3 \end{pmatrix} = \begin{pmatrix} k_2 \\ k_3 \\ k_4 \\ b_1 + b_4 \frac{1}{k_4} \\ b_2 - b_4 \frac{k_2}{k_4} \\ b_3 + b_4 \frac{k_3}{k_4} \end{pmatrix} \quad (2.10)$$

The reparametrization formula in Eq. 2.10 explains why enlarging the domain of f_r to include negative values works and why only b_2 needs such treatment: b_2 now needs to take the value of $b_2 - b_4 \frac{k_2}{k_4}$ before, which can be negative, in order to compensate for the fact that b_4 has been set to zero. In general, having a reparametrization formula allows one to inspect whether the reparametrization is bijective under their presumed domains, thereby testing surjectivity.

Here two key facts make applying this second method of *deriving reparametrization formula* possible: the number of parameters is small (six) and the functional form describing how p determines model behaviors is friendly (simple ratio-

nal functions, thanks to our use of q), which makes equations like Eq. 2.9 solvable. In general, however, models do not enjoy these two facts and deriving reparametrization formula can be difficult.

The third method, which we call *isocurve enumeration*, is based on the following idea: if f_r misses part of \mathcal{M} , then the isocurve corresponding to a point in the missed part would not lead to f_r , but to some other f_r' instead; upon sampling on \mathcal{M} and plotting the isocurves, we can attempt an enumeration of all classes of isocurves and their associated f_r' 's; with such an enumeration, we can somehow piece these different f_r' 's together to form a surjective map to \mathcal{M} . Applying this method to our case yields the following figure.

Fig. 2.13 shows all classes of isocurves we have found, four in total (subplot (a)). Two prominent features stand out. First, there are indeed more than one class of isocurves, and two classes ($b_1 \leftrightarrow b_2$ and $b_2 \leftrightarrow b_3$) have no intersection with the $b_4 = 0$ boundary chosen to construct f_r , which explains our loss in coverage of part of \mathcal{M} . Second, all isocurve classes end in component boundaries, which means that *every point on \mathcal{M} is identified by some component boundary, and a complete coverage of \mathcal{M} through a collection of component boundaries is possible.*

How to get the collection of component boundaries that give a complete coverage of \mathcal{M} ? In our present case, it is easy and we can eyeball out the answer: since all isocurve classes intersect with component boundary $b_1 \rightarrow 0$ or $b_3 \rightarrow 0$, collection $\{b_1 \rightarrow 0, b_3 \rightarrow 0\}$ works; the choice is not unique, collection $\{b_2 \rightarrow 0, b_4 \rightarrow 0\}$ also works for a similar reason. This represents our solution for the case so far:

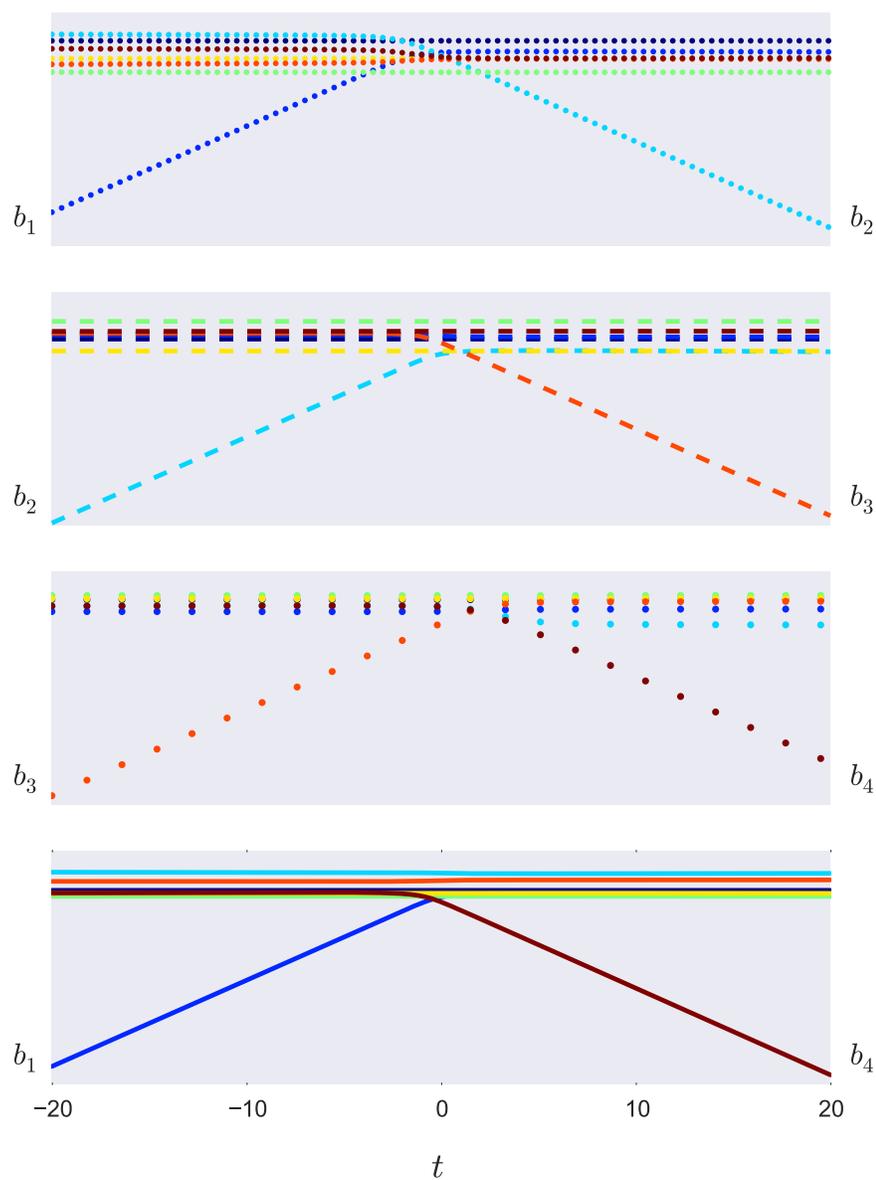


Figure 2.12: Enumerating all four classes of isocurves for $(\text{path2mm}_{k_1=1}, \text{XC})$

representing \mathcal{M} through a collection of two models corresponding to two component boundaries resolves the SN and ensures surjectivity.

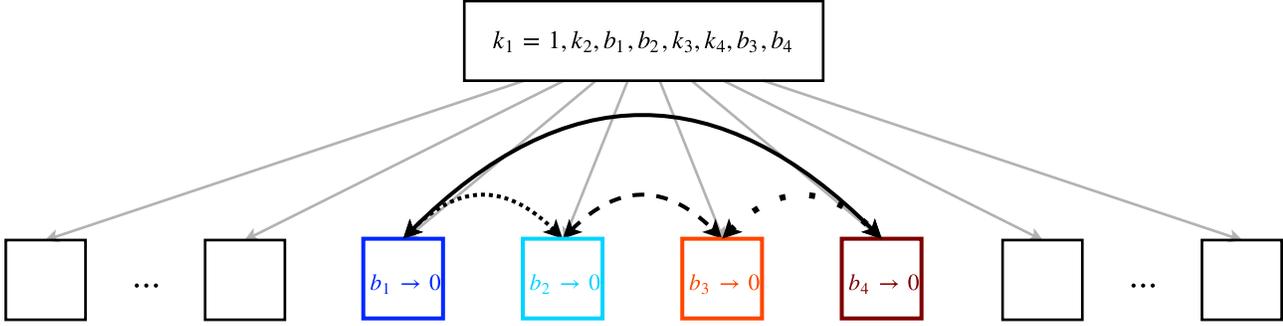


Figure 2.13: "Connectome" of isocurve classes overlaid on Hasse diagram. The colors and line styles of the two subplots match each other.

However, even after such a long quest, two sets of unsettling questions remain:

1. Why the isocurve class pattern? Specifically, why both $\{b_1 \rightarrow 0, b_3 \rightarrow 0\}$ and $\{b_2 \rightarrow 0, b_4 \rightarrow 0\}$ work? Why do the isocurves classes consist of $b_1 \leftrightarrow b_2, b_2 \leftrightarrow b_3, b_3 \leftrightarrow b_4$ and $b_1 \leftrightarrow b_4$, but not $b_1 \leftrightarrow b_3$ or $b_2 \leftrightarrow b_4$? Is there a connection between the presence of, say, b_1 and b_3 , in the collection of models, and the absence of them in the isocurve classes? It would be nice to have a better understanding on the conceptual questions.
2. What does it mean by having a model represented by a collection of two models? Specifically, how to perform model prediction, parameter estimation and model reduction in this case? How is the topology of one model represented by two in terms of Hasse diagram? It is important to have a solution to the practical issues.

To understand these puzzles, we will need to rework our mental picture, starting in the next section.

2.5.4 Augmentation

We started out thinking about structural nonidentifiability mostly algebraically: its signature involves the rank of a linearized map, its resolution involves reparametrizations, etc. Is there a geometric perspective on SN? In Sec. 2.1, we described one way different behavioral spaces can be related geometrically: B can morph to P through a homotopy and then to B' through another homotopy. There is another way to relate two behavioral spaces, which we call B and \tilde{B} here, geometrically when they have a nested relationship, that is, \tilde{B} describes not only B but also some other behaviors \bar{B} : $\tilde{B} = B \times \bar{B}$. This way turns out particularly pertinent for understanding SN.

First, when $\tilde{B} = B \times \bar{B}$, \mathcal{M} can be thought as a *projection* of $\tilde{\mathcal{M}}$ from \tilde{B} to B . Second, just like the first way where a geometric interpretation of the *end points* (manifolds) leads to a geometric interpretation of the *process* (homotopy), the two end points of the projection, \mathcal{M} and $\tilde{\mathcal{M}}$, also admit a geometric interpretation of the projection process. To see this, let us make a new manifold $\tilde{\mathcal{M}}_\sigma$ in the following way: we represent a point on it as $\tilde{y}_\sigma = \begin{pmatrix} y \\ \bar{y}/\sigma \end{pmatrix}$, where $y \in \mathcal{M}$, $\bar{y} \in \bar{\mathcal{M}}$, and σ is a positive real number. When $\sigma = 1$, $\tilde{\mathcal{M}}_\sigma$ is the same as $\tilde{\mathcal{M}}$; however, as we increase σ from 1 to ∞ , the dimensions corresponding to \bar{B} get flatter and flatter,

and eventually vanish: $\begin{pmatrix} y \\ \bar{y}/\sigma \end{pmatrix} \rightarrow \begin{pmatrix} y \\ 0 \end{pmatrix}$ as $\sigma \rightarrow \infty$. Interpreting $\begin{pmatrix} y \\ 0 \end{pmatrix}$ as a point on an embedding of \mathcal{M} in \tilde{B} , we have $\tilde{\mathcal{M}}_\sigma \rightarrow \mathcal{M}$ as $\sigma \rightarrow \infty$, and therefore a homotopy from $\tilde{\mathcal{M}}$ to \mathcal{M} as σ goes from 1 to ∞ (replacing σ with $\lambda = 1/\sigma$ would make the notation standard).

This second way to relate two behavioral spaces geometrically suggests the following perspective on SN. Imagine that we probe both behavioral spaces B and \bar{B} and the resulting behavioral map has no SN: $\dim \tilde{\mathcal{M}} = \dim P$. However, the equipment for measuring behaviors \bar{B} , but not B , deteriorates quickly, manifested in larger and larger uncertainties in its measurements, and one day it stops functioning altogether: this whole “uncertainty blowup” process can be formalized as the $\sigma \rightarrow \infty$ process which morphs $\tilde{\mathcal{M}}$ to \mathcal{M} as described above. If the resulting behavioral map that maps to y only has SN, then $\tilde{\mathcal{M}}$ has *collapsed* to a lower-dimensional \mathcal{M} .²⁶

This perspective on SN as manifold collapse due to uncertainty blowup is particularly enticing. We can forget for a moment all the algebraic subtleties of SN, and enjoy the imagination of how a round muffin gets flattened into a pancake. In addition to its geometric intuitiveness, this perspective also emphasizes the dependence of SN on behavioral spaces and highlights the fact it is not an intrinsic property of the model. Moreover, the continuous flattening to the ultimate collapse provides a natural bridge to *practical nonidentifiability*, which happens when σ is large, but not yet infinite. Lastly and most importantly, it suggests that a

²⁶The difference between collapse and projection is that collapse necessarily implies loss of dimension, while projection does not; collapse is a projection that is lossy in dimension.

process reverse to the uncertainty blowup would *resurrect* the collapsed manifold back to the full one; if we just look at the two end points of such a reverse process, it is as if we pull some new data out of the hat and *augment* it to the existing one, and voilà, the flat pancake turns back into a round muffin again.

This motivates the procedure of **augmentation**: given a model and a behavioral space B with SN ($|DF| = 0$), find an *augmenting behavioral space* \tilde{B} , such that the augmented behavioral map $\tilde{F} = \begin{pmatrix} F \\ \tilde{F} \end{pmatrix}$ is free of SN: $|D\tilde{F}| \neq 0$. Geometrically, the collapsed \mathcal{M} is resurrected to $\tilde{\mathcal{M}}$.

What is the point of having a resurrected $\tilde{\mathcal{M}}$? Three thoughts are in order. First, \tilde{f} has no SN so $\dim \tilde{\mathcal{M}} = \dim P$ and the boundaries of $\tilde{\mathcal{M}}$, $\partial\tilde{\mathcal{M}}$'s, then, have dimensions lower than P . Second, upon manifold collapse, some of those boundaries of $\tilde{\mathcal{M}}$ would be mapped to a nonzero volume of the interior of \mathcal{M} ; in other words, the collapse establishes a correspondence between \mathcal{M} and the boundaries of $\tilde{\mathcal{M}}$. Third, that a boundary of $\tilde{\mathcal{M}}$ is mapped to a nonzero volume of \mathcal{M} interior means that the boundary is able to describe the behaviors that represented by the volume in \mathcal{M} . Now let us combine these three thoughts: if some boundaries of $\tilde{\mathcal{M}}$ are able to describe whatever \mathcal{M} describe and at the same time have fewer parameters, doesn't it mean we have a way to find an SN-free representation of the model? It is also easy to convince ourselves that such boundaries have to have exactly the same number of parameters as the dimension of \mathcal{M} in order to be considered a candidate for such an SN-free representation (if $\dim \partial\tilde{\mathcal{M}} > \dim \mathcal{M}$, then $\partial\tilde{\mathcal{M}}$ is still overparametrized; if $\dim \partial\tilde{\mathcal{M}} < \dim \mathcal{M}$, then it must be mapped to a zero volume

in \mathcal{M}).

Let us look at an example.

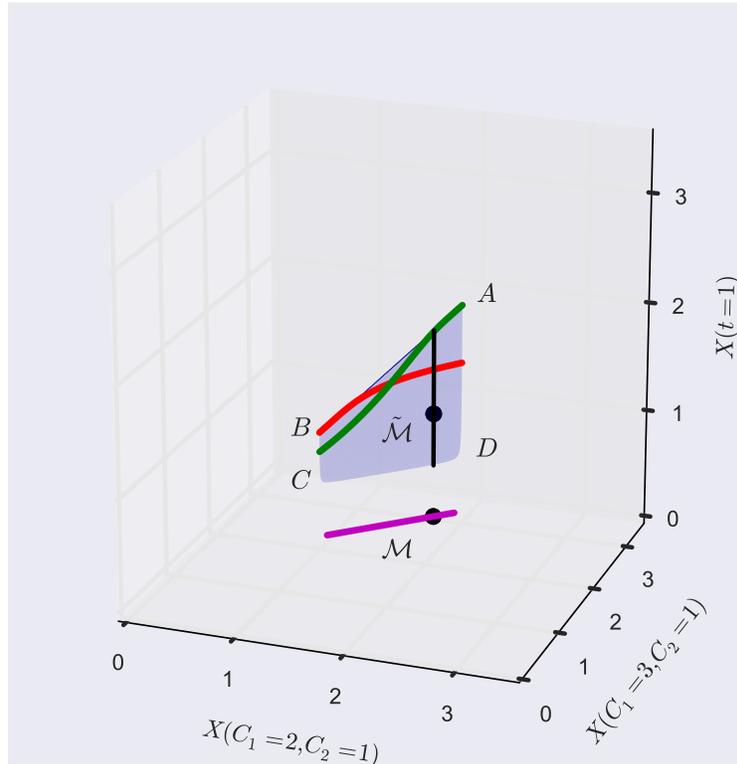


Figure 2.14: Manifold resurrection for $(\text{path2mah}, \text{XC})$ with XT as the augmenting behavioral space. $(\text{path2mah}, \text{XC})$ has $(n, r, d) = (2, 1, 1)$, so its manifold \mathcal{M} is a collapsed curve represented by the magenta line segment. Using XT as the augmenting behavioral space resurrects the manifold to $\tilde{\mathcal{M}}$ represented by the blue trapezoid. The red and green curves represent the currying reparametrizations of $(\text{path2mah}, \text{XC})$ with k_1 and k_2 set to 1 respectively. The black curve represents an isocurve on $\tilde{\mathcal{M}}$, with its starting point and its projection on \mathcal{M} represented by the two solid dots.

Figure 2.14 shows an example of manifold resurrection. For model `path2mah`,

the original behavioral space $B = X^C$ has $(n, r, d) = (2, 1, 1)$, and as such, its manifold \mathcal{M} is a one-dimensional curve, here represented by the magenta line segment. We choose XT as the augmenting behavioral space, add one prediction $\bar{y} = X(t = 1)$ and resurrect \mathcal{M} to $\tilde{\mathcal{M}}$, now a two-dimensional surface, here represented by the blue trapezoid. Among the four one-dimensional boundaries of $\tilde{\mathcal{M}}$, AB (corresponding to $k_1, k_2 \rightarrow \infty$) and CD (corresponding to $k_1, k_2 \rightarrow 0$) are mapped to nonzero volumes in \mathcal{M} , in fact, the whole of \mathcal{M} , upon collapse, which we interpret as meaning that AB and CD can describe the behaviors encoded in \mathcal{M} as well as \mathcal{M} can. Since AB and CD have only parameter, using them to describe the behavioral space B resolves the SU. Also shown in the figure are the two submanifolds of $\tilde{\mathcal{M}}$ corresponding to the currying reparametrizations of $(\text{path2mah}, \text{XC})$ (Sec. 2.5.2): $k_1 = 1$ or $k_2 = 1$. That the two submanifolds, upon collapse, are also mapped to the whole of \mathcal{M} explains why the reparametrization works.

Both the abstract reasonings so far and the concrete example above suggest that we have found a general method to resolve SN: given a model and a behavioral space B with SN, we use augmentation to resurrect the collapsed manifold \mathcal{M} to $\tilde{\mathcal{M}}$, and look for a set of boundaries of $\tilde{\mathcal{M}}$ that satisfies the following requirements:

- Each boundary $\partial\tilde{\mathcal{M}}$ in the set has the same dimension as \mathcal{M} .
- Each boundary $\partial\tilde{\mathcal{M}}$ in the set, upon collapse, is mapped to a nonzero volume in \mathcal{M} ; symbolically, one can write this requirement as $|DF'| \neq 0$ where $F' =$

$$\lim_{p \rightarrow p_\partial(\partial\tilde{\mathcal{M}})} F.$$

- The set covers the whole \mathcal{M} upon collapse.

The first two requirements ensure that a boundary $\partial\tilde{\mathcal{M}}$ can *identify* at least part of \mathcal{M} , or mathematically, injective; a boundary that satisfies both requirements is called an *identifying boundary* and a boundary that does not satisfy the second requirement is called a *nonidentifying boundary*. The third requirement ensures that the set as a whole is covering, or mathematically, surjective; a set of identifying boundaries that satisfies this requirement is called a *covering set of identifying boundaries*.

This is remarkable. It was not very long ago when, confronted by SN and having limited gizmos in our bag, we often felt inadequate and insecure. If luck strikes we could crack some small cases open, otherwise the situation would appear hopeless. The first approach of some yet limited generality appeared when we discovered that for cases with $d = 1$, traveling along the isocurves to the limit can reveal boundary reductions, which, when coupled with careful surjectivity analysis, could resolve the one lingering degree of SN. What we discover here, however, is more general: with the geometric perspective on SN, we have transformed the SN resolution problem to a manifold boundary identification problem, which we feel more familiar with and works for SN in any degree.

This reformulation of the problem to one involving manifold boundary identification suggests the following way to go about: given an \mathcal{M} and an $\tilde{\mathcal{M}}$, enumerate the boundaries of $\tilde{\mathcal{M}}$ and search within for a covering identifying set of bound-

aries that satisfy the three requirements above. However, it turns out that we can incorporate the reasoning underlying isocurve traversal for cases where $d = 1$ to make it more efficient to find a covering identifying set of boundaries. To see this, let us imagine the trajectory of an isocurve on $\tilde{\mathcal{M}}$. An isocurve by definition is a curve that is mapped to a point on \mathcal{M} , and since \mathcal{M} can be thought as $\tilde{\mathcal{M}}$ in \tilde{B} (or an associated finite-dimensional prediction space \tilde{Y}) projected to B (or Y), the isocurve has to be perpendicular to B (or Y). Fig. 2.10 shows such an isocurve: starting at the point represented by the solid dot on $\tilde{\mathcal{M}}$, the isocurve has two directions to follow, either shooting vertically up to AB or vertically down to CD in the figure; in this way the isocurve on $\tilde{\mathcal{M}}$ would be projected to the corresponding solid point on \mathcal{M} .

How to generalize this to $d > 1$, where a well-defined isocurve ceases to exist? It lies on the following mathematical statement: *If a geodesic on $\tilde{\mathcal{M}}$ has initial velocity vector $(D\tilde{F})v_0$ where v_0 is any vector in $\ker DF$, then it does not intersect with nonidentifying boundaries.* To see this, let us first write $D\tilde{F} = \begin{pmatrix} DF \\ D\tilde{F} \end{pmatrix}$ and hence $(D\tilde{F})v_0 = \begin{pmatrix} (DF)v_0 \\ (D\tilde{F})v_0 \end{pmatrix} = \begin{pmatrix} 0 \\ (D\tilde{F})v_0 \end{pmatrix}$. That is, the initial velocity of the geodesic is perpendicular to \mathcal{M} . Suppose the geodesic starts at point $\tilde{y}(0)$ with its projection $y(0)$ on \mathcal{M} : any geodesics starting at $\tilde{y}(0)$ with initial velocity perpendicular to \mathcal{M} would also have its whole curve projected to $y(0)$, that is, a zero volume on \mathcal{M} . Since nonidentifying boundaries are also projected to zero volumes on \mathcal{M} , the projections do not intersect on \mathcal{M} generically, which shows that the geodesic and the nonidentifying boundaries do not intersect on $\tilde{\mathcal{M}}$ generically, for if they do on $\tilde{\mathcal{M}}$, their projections also do on \mathcal{M} .

In essence, we have generalized the notion of isocurves: any such a geodesic would be projected to a single point on \mathcal{M} yet they require manifold resurrection and are well-defined even when $d > 1$. We call such geodesics *generalized isocurves*, or, almost always, simply isocurves. The use of generalized isocurves greatly improves the efficiency in our search for a covering set of identifying boundaries: now we do not need to waste time enumerating boundaries of $\tilde{\mathcal{M}}$ that are bound to be useless in resolving SN (ie, the nonidentifying boundaries); instead, we focus on the potentially useful ones by going only orthogonally with respect to \mathcal{M} (or B). Each isocurve has its ends either going unbounded, or connected to some boundaries; we can think of both cases as going to some generalized boundaries. An *isocurve class* is a class of isocurves connected to the same boundaries; as an *equivalent class* on the set of isocurves, we often simply say "an isocurve" to mean the whole class of which the isocurve is a representative. Following our convention in Sec. 2.5.3, we call such enumeration of all isocurve classes **isocurve enumeration**.

There is one more property to be desired for a covering set of identifying boundaries: it would be nice if the set covers \mathcal{M} in a *disjoint* fashion, each covering a distinct part of it without any double counting of \mathcal{M} ; we call such a set a **disjoint covering set of identifying boundaries**, or simply disjoint covering set. There is no guarantee that such a set exists, as illustrated in Fig. 2.15. However, the geometry of most model manifolds from science is usually not so perverse and, to our satisfaction, a disjoint covering set can usually be found.²⁷ As we will see in the next section, the property of disjointness confers much convenience when

²⁷A covering set of identifying boundaries, however, must exist by the projection argument.

we try to *stitch* the set of boundaries together. A (disjoint) covering set of identifying boundaries is also not unique, as shown in the examples of $(\text{path2mm}, \text{XC})$ (Sec. 2.5.3) and $(\text{path2mah}, \text{XC})$ (Fig. 2.14).

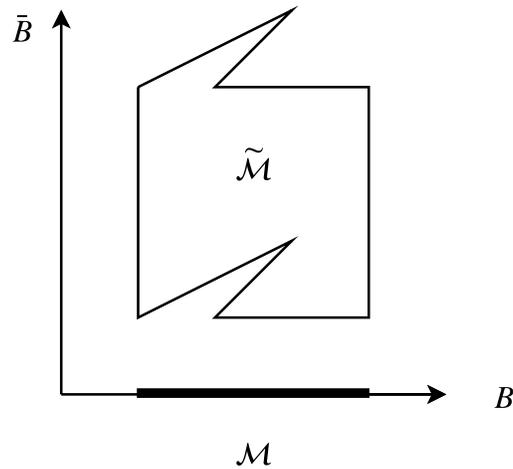


Figure 2.15: Illustration of a resurrected manifold $\tilde{\mathcal{M}}$ without a disjoint covering set of identifying boundaries for \mathcal{M} .

2.5.5 Hasse-isocurve diagram and stitchwork

In both examples of $(\text{path2mm}, \text{XC})$ (Sec. 2.5.3) and $(\text{path2mah}, \text{XC})$ (Fig. 2.14), $d = 1$. This makes searching for a disjoint covering set of identifying boundaries straightforward. Things become more complicated when $d > 1$, where the set of identifying boundaries are hyperedges, hypercorners, etc. while the enumerated isocurves first connect hyperfaces. More work needs to be done in this case.

Imagine a model with $(n, r, d) = (3, 1, 2)$. Its manifold \mathcal{M} would be a curve, to be resurrected to a solid $\tilde{\mathcal{M}}$. How to get a disjoint covering set of identifying boundaries, that is, a set of edges of the solid?

Fig. 2.16 illustrates such a situation. Not from an actual model, the collapsed \mathcal{M} is assumed to be a line segment along the y_1 axis; upon augmentation along the \bar{y}_2 and \bar{y}_3 axes, the manifold is assumed to have been resurrected to a tetrahedron $\tilde{\mathcal{M}}$. The first round of orthogonal explorations would return isocurve classes connecting pairs of faces, but what do they represent? Remember that what we want is a set of edges.

To know how to proceed in this case, we need to understand the following two things: that in our first round isocurves connect only faces but not edges means that we need to descend *twice*, first from tetrahedron interior to faces, then from faces to edges; since we need to make sure that the set of edges we ultimately find cover the whole line segment \mathcal{M} upon collapse, in each iteration of descension we need to make sure that we have not lost any coverage of \mathcal{M} . To achieve this successive descension without loss of coverage, we adopt the following approach: in each iteration we overlay the isocurve classes on top of the Hasse diagram, just like in Fig. 2.13, and find a set of boundaries on the diagram that have all isocurves connected to at least one of them. We name such a Hasse diagram with isocurves connecting nodes within the same rank as edges a **Hasse-isocurve diagram**.

In our tetrahedron example, some pairs of faces would constitute a covering set, but not a single one would suffice. We get this conclusion through eyeballing

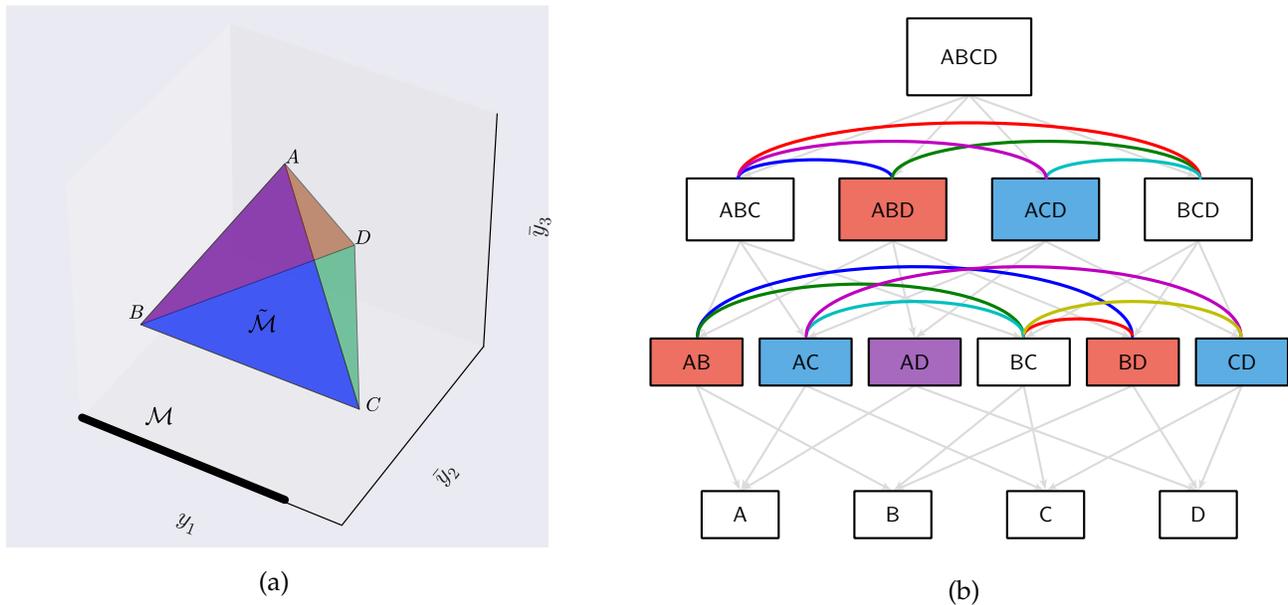


Figure 2.16: Manifold resurrection and Hasse-isocurve diagram for a hypothetical model with $d = 2$. (a) Resurrection of a collapsed line segment to a tetrahedron. The solid black line segment along the y_1 axis represents the collapse manifold \mathcal{M} , and the tetrahedron represents the resurrected manifold $\tilde{\mathcal{M}}$. (b) The associated Hasse-isocurve diagram. To get a covering set of identifying boundaries, one needs to travel two ranks down, first from the interior of the tetrahedron $ABCD$ to its faces, then from the faces to the edges. Enumerating isocurves in $ABCD$ gives the isocurves and the faces they connect, as shown in the upper set of arching edges connecting nodes of corank one; the colored ABD and ACD nodes represent the chosen faces for descension which form a covering set of \mathcal{M} . Further enumerating isocurves in ABD and ACD gives the lower set of arching edges connecting nodes of corank two (AD is shared by ABD and ACD , hence in a different color), from which a covering set of identifying boundaries can be found: any of $\{BD, CD\}$, $\{AB, AC\}$ or $\{BC\}$ works.

the tetrahedron or its Hasse-isocurve diagram, but as we can imagine, the connection pattern gets complicated quickly with dimension, and an automated algorithm is needed. Formally, what we want is called a *vertex cover* of the the nodes within a rank: a subset of the nodes that have every edge incident to some of them; the problem is a classic NP-hard problem, but many good algorithms exist, and what is really relevant here is that we can automate the search for a vertex cover, even if brute-forcefully, as typically the preceding task of isocurve enumeration is the computational bottleneck. Given a graph typically there are many vertex covers, and obviously the smaller the better; together with our desire for disjointness, we search for a *minimal disjoint vertex cover* within a rank, where disjointness here translates to no vertices in the cover are joined by an edge.²⁸

Let us pick a minimal disjoint vertex cover for the nodes with corank one (ie, the faces), say $\{ABD, ACD\}$. Now let us descend to each of them, and enumerate the isocurves again. On ABD , isocurve enumeration gives us (AB, BD) and (AD, BD) and on ACD gives us (AD, CD) and (AC, CD) . Now for these four edges, we pick our minimal vertex disjoint vertex cover for the second time, say, $\{BD, CD\}$. The important claim here is: $\{BD, CD\}$ is a covering set of identifying boundaries for \mathcal{M} which also happens to be disjoint. To see why, we can interpret the fact that $\{BD, CD\}$ is a vertex cover of the four isocurve classes as meaning that the set $\{BD, CD\}$ is as good as its parental faces $\{ABD, ACD\}$ when representing \mathcal{M} is concerned, for *any* isocurve of \mathcal{M} on $\{ABD, ACD\}$ hits either BD or CD ; applying this reasoning one more time to $\{ABD, ACD\}$ as a simpler replacement of the whole

²⁸Disjointness does not imply minimality.

tetrahedron $ABCD$ for representing \mathcal{M} completes our argument.

Looking at the tetrahedron, we see that $\{BD, CD\}$ constituting a covering set makes intuitive sense: their projections exactly cover the whole \mathcal{M} . Similarly, $\{AB, AC\}$ also works, which is the other minimal disjoint vertex cover of the four isocurve classes of the two faces. Remarkably, $\{BC\}$ alone should work: indeed, if we choose $\{ABC, BCD\}$ as the covering set of faces, we would have $\{BC\}$, which is truly the minimal covering set of identifying boundaries that we are looking for. This teaches us a lesson: going for disjoint covering set in earlier rounds may not optimize our final outcome. Unfortunately, there is no algorithm so far that guarantees a globally minimal covering set; all we can do so far is to explore different paths and choose the best one we get.

Lastly, let us think about what we do if we settle down to $\{BD, CD\}$ as a covering set. We have two models here, corresponding to $\{BD, CD\}$, but originally we have only one, corresponding to \mathcal{M} . What does it mean by having two models represent one?

Geometrically, this is easy to understand: \mathcal{M} is partitioned into different parts with different functional representations, like a piecewise function; if each part has only one functional representation, then the covering set is also disjoint. Therefore, if we fit such a "piecewise model" to data, for example, we will have to fit two models and choose the better one.

When it comes constructing model topology, there is one important operation

for a disjoint covering set $\{\partial\tilde{\mathcal{M}}\}$. Because of the disjointness, different $\partial\tilde{\mathcal{M}}$'s share borders on \mathcal{M} but do not overlap. To the $\partial\tilde{\mathcal{M}}$'s, the borders look like their boundaries (that is, the boundaries of $\partial\tilde{\mathcal{M}}$'s), but to \mathcal{M} , the borders represent its interior. Therefore, when we construct the Hasse diagram for \mathcal{M} using a disjoint covering set, we need to remove its shared boundaries, a procedure we call **stitchwork** and illustrated in Fig. 2.17.

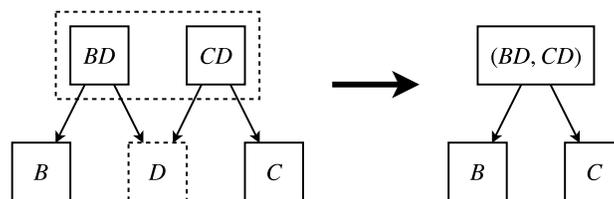


Figure 2.17: A simple example of stitchwork. Using $\{BD, CD\}$ to represent \mathcal{M} in our tetrahedron example, one needs to remove the boundaries shared between BD and CD , that is D . Hence the resulting topology is a line segment, like \mathcal{M} .

2.5.6 Summary

To summarize the details in the previous sections, we have developed a general method for resolving structural nonidentifiability. We call the method **manifold boundary identification method** (MBIM), and it has the following steps:

1. Augmentation:

Given a model of n parameters and a behavioral space B with SN of de-

gree d , find another behavioral space \bar{B} such that the augmented behavioral map $\tilde{F} = \begin{pmatrix} F \\ \bar{F} \end{pmatrix}$ has no SN; geometrically this step resurrects the collapsed $(n - k)$ -dimensional manifold \mathcal{M} to an n -dimensional manifold $\tilde{\mathcal{M}}$.

2. Iterative isocurve enumeration followed by vertex covering:

Set $\mathcal{S}_0 = \{\tilde{\mathcal{M}}\}$.

For $k = 1, 2, \dots, d$:

- (a) Perform isocurve enumeration on each element in \mathcal{S}_{k-1} .
- (b) Collect the isocurve classes from all elements in \mathcal{S}_{k-1} into \mathcal{E}_k , representable by a Hasse-isocurve diagram on nodes of corank k .
- (c) Choose a minimal vertex cover C_k on \mathcal{E}_k .
- (d) Set $\mathcal{S}_k = C_k$.

3. Choose a covering set of identifying boundaries:

If \mathcal{E}_k has a minimal disjoint vertex cover C_k^d , choose it as C ; otherwise choose $C = C_k$. The claim is that C is a covering set of identifying boundaries for \mathcal{M} and resolves the SN in B .

If C is disjoint and used to construct the topology of \mathcal{M} , construct the topology of each element in C and perform stitchwork.

Some general remarks are in order. First, since compositionality suggests that many boundaries are component boundaries, a manifold boundary resolution of SN found by MBIM would often be local and hence interpretable. Second,

counterexamples exist. For example, for $(\text{cycle3ma}, \text{JC})$ (Sec. 2.5.2), applying MBIM yields the same global reparametrization as found by the derivation. Third, the existence of a local reparametrization should depend on when the manifold collapse takes place: for a composition of a component model and a coupling to form a system model, if the collapse takes place before the coupling, then in principle one can find a local reparametrization; otherwise only global reparametrizations exist.

This concludes our journey on the land of structural nonidentifiability, starting out shakily but ending on a triumphant note. Looking about at this vantage point, we note that there is a poetic correspondence between MBAM which mainly uses geometric information and helps resolve practical nonidentifiability and MBIM which mainly uses topological information and helps resolve structural nonidentifiability. Yet even MBIM is geometric in inspiration, as it starts from the manifold collapse perspective on SN. We have seen again and again the awesome power of geometric perspective. Here it helps us redefine the flavor of the problem, from chasing elusive, often-global and impenetrable reparametrizations, to identifying less-complicated yet as-expressive models at manifold boundaries. Aside from the intuitiveness and conduciveness to algorithms, geometry also offers generality: it is oblivious to whether the model is a dynamical system or a neural network; "Come what may, I will morph it to something comfortable for you." it says.

But there is something to be said on the other side: some algebraic insight can be lost in the geometric translation. For example, for $(\text{path2mah}, \text{JC})$, algebra

tells us that the behavioral map J^C has SN for this model because it is the harmonic mean of the two rate constants $\frac{2k_1k_2}{k_1+k_2}$ that matters (Section 1.6). Like a delicate message from a nurturing teacher. When the problem is handed over to geometry, however, it somehow acts like a peevish manager, pointing k_1 to infinity and thereby sending it to nihility, followed by a smug message "I knew what can be done by one does not need two."

So perhaps the balance is more subtle than the triumph suggests.

2.6 Construction of system topology

Using system-component formulation and the features of system topology that have been characterized so far, we naturally arrive at an algorithm on constructing the topology of a system model. The algorithm as it is now is provisional and not general, as there are situations that would impede its execution, but is helpful in our limited experience of using it. In this section, we provide a brief description of the algorithm, and show its application to some concrete system models from metabolism in the next chapter.

First, given a system model composed of a set of component models, if the system model has structural nonidentifiability, one applies MBIM to remove it (Section 2.5.6). Those in the identifying set of boundaries returned by MBIM are either component boundaries or emergent boundaries: for component boundaries,

MBIM preserves the “localness” of the model specification, that is, the system model is *still* composed of a set of component models after the removal of SN; for emergent boundaries, one tries to convert them to a currying reparametrization so that the model specification stays “local”. If the system model has no SN, then nothing needs to be done in this first step. The key point here is to try as hard as one can to preserve the system-component formulation of a system model.

The benefits of such a formulation of a system model can be seen at this second step: Given a system model composed of a set of component models, one then constructs the *component product topology*, that is, the cartesian product of all boundaries of all component models, representable by a Hasse diagram where each node corresponds to a reduced system model constructed using a reduced component model from each of its components. This step would not be possible if the system model is not formulated as a composition of component models. Reduced system models arising this way can have SN for the same reason as our original full system model can have SN. We remove them one by one using MBIM and again try as hard as we can to preserve the system-component formulation of the system models.

Upon removal of all SN, such a component product topology immediately gives us many system boundaries thanks to the feature of compositionality (Section 2.4.2). Moreover, they serve as a “scaffold” for our searching for emergent boundaries: as a third step, we explore the boundary structure at each of these system models constructed using a set of reduced component models. Note that,

thanks to symmetry (Section 2.4.5), sometimes we do not need to explore all of them: a topological characterization of one part of the system manifold can be translated into those of other parts.

For a given node on such a Hasse diagram, if no emergent boundaries are found on it, one moves on exploring other nodes. If some emergent boundaries are found, one incorporates them into the Hasse diagram, and continues exploring the boundary structures of them down the ranks (ie, dimensions of the models) of the Hasse diagram, until one reaches the zero-dimensional vertices on the system manifold, thus completing its topological characterization.

The abstract description of the algorithm above will be complemented by two examples in the next chapter (Sections 3.3 and 3.4.3). Here we conclude this section with a few remarks on the algorithm.

The algorithm highlights the utility of the system-component formulation of a system model. First, it guarantees interpretability: parameters, and hence boundaries, of system models are often hard to interpret if they have no direct connections with parameters of component models (eg, Section 2.5.2). Second, it gives us many system boundaries at almost no cost. Third, it gives us anchoring points in our search for emergent boundaries. Last, it allows for *information reuse*: a topological characterization of a component model can be used in the studies of any behaviors of any system models as long as the construction of the system model involves that component model.

2.7 Appendix: Implementation

The codes used for the computation in this chapter have been deposited to <http://github.com/leihuang/infotopo>. Here we provide a brief description of the code structure and some simple examples. Note that the codes have not been seriously tested and can seem brittle to new users.

The codes reside in a Python package which we call `infotopo`, and the package revolves around a few core classes whose relationships are summarized by the following diagram.

Most upstream of the whole pipeline is class `Model`, which represents whatever system one studies. We study reaction networks, so our `Model` class is called `Network`, essentially a wrapper of SloppyCell's `ReactionNetwork` class [76]. Anyone who studies, for example, neural networks, can write a similar wrapper over some popular implementations of neural network models. The key point is that, *the codes mean to mimic how information geometry and information topology abstract various mathematical models: different models may have vastly different representations, behaviors or implementations, but starting from the prediction map f / the `Predict` class, analysis / computation converge.*

In the specific model family of reaction networks, to facilitate studying its system-component formulation, we define a `RateLaw` class, instances of which can be composed to form a `Network` object. Studying key behaviors of a reaction network (including all those listed in Table 1.1) boil down to the following

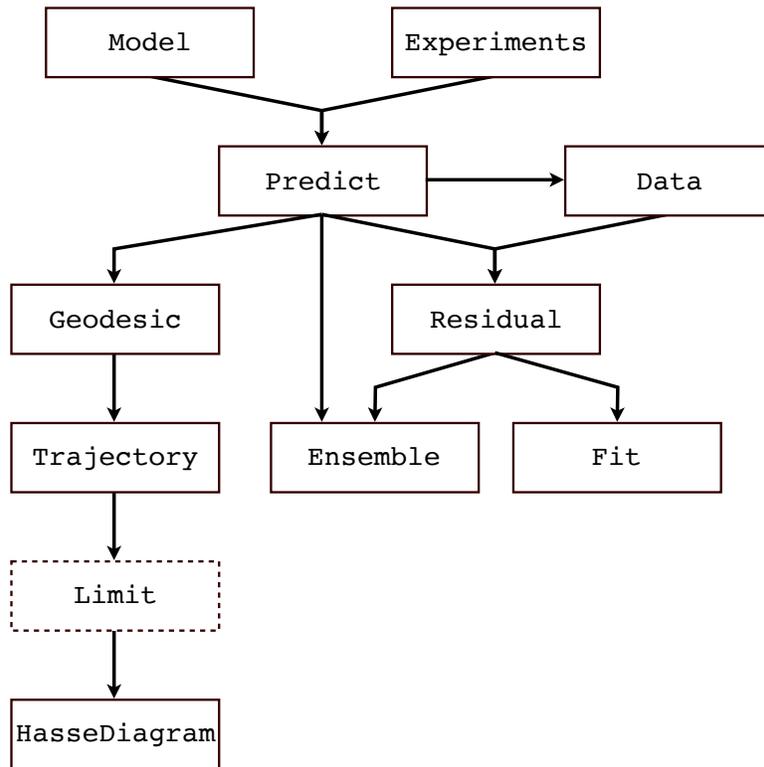


Figure 2.18: Relationships between the core classes in the package `infotopo`. The central class is `Predict` (ie, a prediction map f), derived from a `Model` object (specifying a mathematical model \mathcal{M}) and an `Experiments` object (specifying a behavioral space B and a sampling X). With a `Predict` object, one can perform a number of standard modeling tasks: generating predictions, fitting it to data (by creating a `Residual` object and computing a `Fit` object), and sampling the parameter space using different combinations of priors and posteriors (represented by `Ensemble` objects). A `Predict` object is also key to some information geometric and information topological analysis: it can generate a `Geodesic` object, which can be integrated to form a `Trajectory` object and the limiting behaviors can be examined in a `Limit` object (only partially implemented so far; hence the dashed box) constructed from an integrated geodesic; sampling many limiting behaviors starts to accumulate global and topological information, stored in a `HasseDiagram` object.

computations: inspecting the structure of a network (eg, stoichiometry matrix and steady-state flux matrix), integration to get its dynamics, rootfinding to get its steady states and MCA computations. This is summarized in the following diagram.

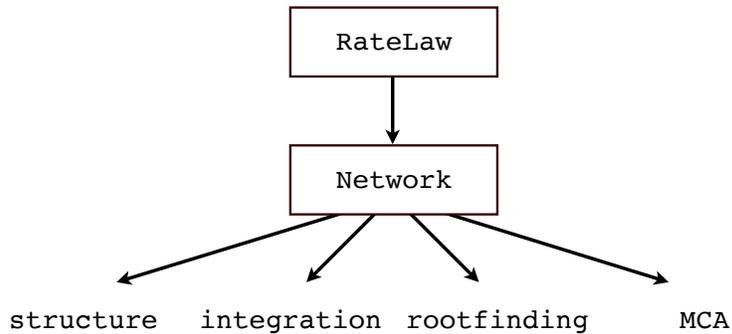


Figure 2.19: The structure and functions of the codes for reaction network models.

Example: The following codes create a network which is a pathway of two reactions with mass-action-Haldane rate laws, ie, `path2mah` model used in previous analysis.

```
>>> from infotopo.models.rxnnet import model, ratelaw
>>> net = model.make_path([ratelaw.mah11]*2, cids=['C1', 'C2', 'KE1', 'KE2',
           ], C1=2)
>>> type(net)
<class 'infotopo.models.rxnnet.model.Network'>
```

The most important method of any `Model` object is `get_predict`, which takes an object from its associated `Experiments` class and outputs a `Predict` object.

In the parlance of the present chapter, an `Experiments` object defines the behavioral space B and its sampling X for a model \mathcal{M} , and the resulting `Predict` object represents the prediction map f . In terms of implementation details, a `Predict` object essentially stores two things: f and Df ²⁹, therefore what `get_predict` does is to infer f (and Df) from the given `Experiments` object.

Example: The following codes create an `Experiments` object for `path2mah`, which consists of the steady-state concentration of the sole metabolite X at two external conditions: $C_1 = 2, C_2 = 1$ and $C_1 = 3, C_2 = 2$ (that is, XC behaviors). Imposing it on the network results in a `Predict` object.

```
>>> from infotopo.models.rxnet import experiments
>>> expts = experiments.get_experiments(zids=net.xids, uids=['C1', 'C2'],
    us=[[2,1],[3,2]])
>>> type(expts)
<class 'infotopo.models.rxnet.experiments.Experiments'>
>>> pred = net.get_predict(expts, tol=1e-13)
>>> type(pred)
<class 'infotopo.predict.Predict'>
```

Most of the analysis revolves around a given `Predict` object. One common task is to compute its spectrum, the list of singular values of the linearization of f at a particular parameter point.

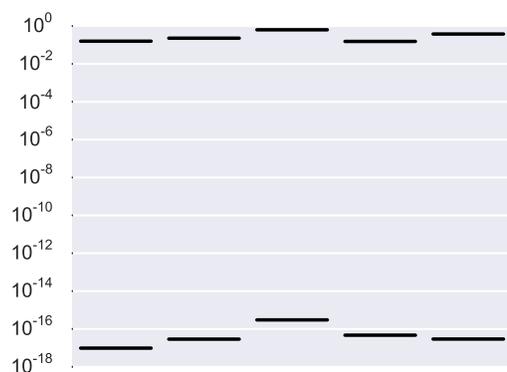
Example: The following codes sample five points in the parameter space and

²⁹Higher-order derivatives are sometimes needed (eg, in geodesic integration) but for now they are computed using finite difference.

demonstrates that the prediction map created above has a structural nonidentifiability of degree $d = 1$ (Section 2.5.2).

```
>>> pred.pids # two parameters
['k1f', 'k2f']
>>> pred.yids # two data points
['(C1=2, C2=1), X, inf', '(C1=3, C2=2), X, inf']
>>> pred.plot_spectra([pred.p0.randomize(seed=i) for i in range(5)],
                      figsize=(4,3))
```

The resulting plot looks like the following.³⁰



To perform the augmentation step in MBIM, one defines some other `Experiments` objects and pad them after the original one plagued by SN.

Example: The following codes create another `Experiments` object for `path2mah`, which consists of the metabolite concentration at time $t = 1$ (that is,

³⁰Plots in this section are given without captions to mimic the IPython notebook environment.

XT behaviors). Upon getting the resulting new `Predict` object, we append it to the old one to get an augmented `Predict` object which is now free of SN.

```
>>> expts2 = experiments.get_experiments(net.xids, uids=['t'], us=[1])
>>> pred2 = net.get_predict(expts2, tol=1e-13)
>>> pred_aug = pred + pred2
>>> pred_aug.yids
['(C1=2, C2=1), X, inf', '(C1=3, C2=2), X, inf', '(), X, 1']
>>> pred_aug.get_spectrum(pred_aug.p0.randomize(seed=1))
array([ 0.27814677,  0.00297135])
```

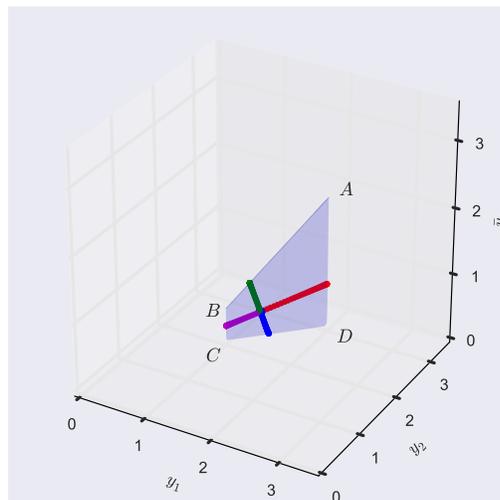
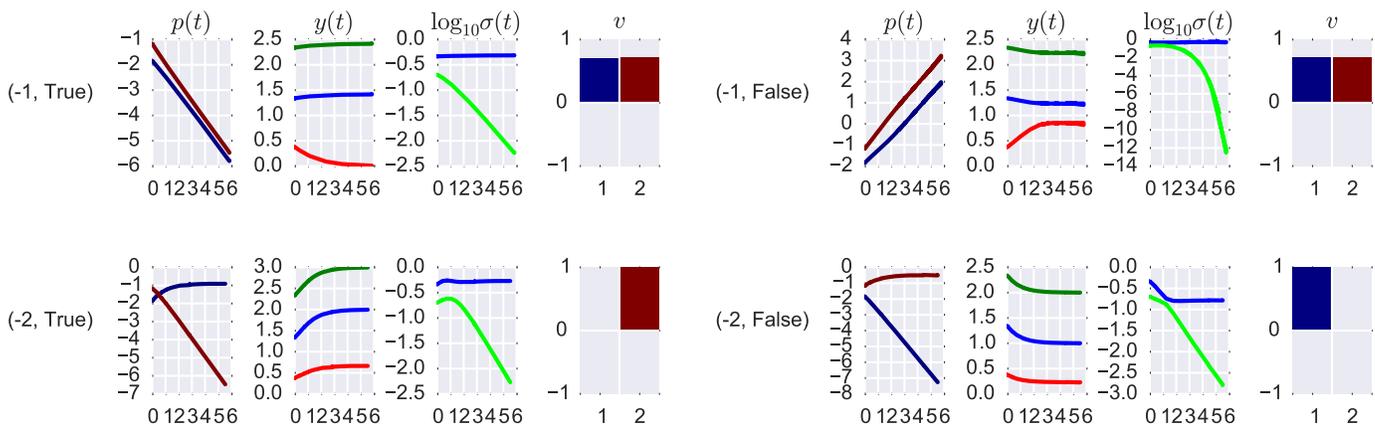
One major information geometric/topological analysis is to travel along geodesics on a model manifold, which is the image of a prediction map f represented by a `Predict` object.

Example: The following codes illustrate the creation of a bunch of `Geodesic` objects (contained in a `Geodesics` object), their integration and visualization. The standard inspection is done through visualizing the parameter trajectories, prediction trajectories, singular value trajectories and the "sloppiest" singular vectors at the terminal time point. We make and show such a plot below (an index of (-1, True) means that the initial velocity of the geodesic is along its sloppiest singular vector in the forward direction, and (-2, False) means along the second sloppiest direction in the reverse direction). Since our augmented prediction map \tilde{f} takes in two parameters and outputs three predictions, we can also visualize its image, ie, the resurrected manifold $\tilde{\mathcal{M}}$, as a surface in \mathbb{R}^3 and overlay the geodesics on it.

```

>>> p0 = pred_aug.p0.randomize(seed=60, sigma=2)
>>> gdss = pred_aug.get_geodesics(p0=p0)
>>> gdss.integrate(tmax=5, dt=0.1)
>>> gdss.plot(figsize=(10,3))
>>> from util import butil
>>> ys = butil.flatten(gdss.apply(lambda gds: gds.ytraj.values), 1)
>>> cs = butil.flatten([[c]*len(gds._ts) for gds, c in zip(gdss, list('
    bgrm'))], 1)
>>> import numpy as np
>>> ks = np.logspace(-2,2,11)
>>> pred.plot_image(ks, ks, alpha=0.2, pts=ys, cs=cs, xyzlims
    =[[0,3.5]]*3, linewidth=0, xyzticks=[range(4)]*3, xyzlabels=[r'$y_1$
    ',r'$y_2$',r'$\bar{y}_3$'])

```

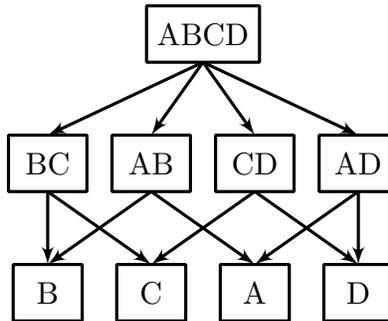


For a manifold, its topological characterization is encoded by a Hasse diagram. We demonstrate manual creation of such a diagram below. For typical system models, their topologies are usually too large to be created manually, and the algorithm described in Section 2.6 is intended to help accomplish this task.

```

>>> from infotopo import hasse
>>> hd = hasse.HasseDiagram(rank=2)
>>> hd.add_node('ABCD', rank=2) # the manifold interior
>>> hd.add_node('BC', rank=1) #  $k_1 \rightarrow 0$ , (-2, False), magenta
>>> hd.add_node('AB', rank=1) #  $k_1, k_2 \rightarrow \text{inf}$ , (-1, False), green
>>> hd.add_node('CD', rank=1) #  $k_1, k_2 \rightarrow 0$ , (-1, True), blue
>>> hd.add_node('AD', rank=1) #  $k_2 \rightarrow 0$ , (-2, True), red
>>> hd.add_node('B', rank=0) #  $k_1, k_2 \rightarrow \text{inf}$ , and  $k_1/k_2 \rightarrow 0$ 
>>> hd.add_node('C', rank=0) #  $k_1, k_2 \rightarrow 0$ , and  $k_1/k_2 \rightarrow 0$ 
>>> hd.add_node('A', rank=0) #  $k_1, k_2 \rightarrow \text{inf}$ , and  $k_2/k_1 \rightarrow 0$ 
>>> hd.add_node('D', rank=0) #  $k_1, k_2 \rightarrow 0$ , and  $k_2/k_1 \rightarrow 0$ 
>>> hd.add_edge('ABCD', 'AB')
>>> hd.add_edge('ABCD', 'BC')
>>> hd.add_edge('ABCD', 'AD')
>>> hd.add_edge('ABCD', 'CD')
>>> hd.add_edge('AB', 'A')
>>> hd.add_edge('AB', 'B')
>>> hd.add_edge('BC', 'B')
>>> hd.add_edge('BC', 'C')
>>> hd.add_edge('AD', 'A')
>>> hd.add_edge('AD', 'D')
>>> hd.add_edge('CD', 'C')
>>> hd.add_edge('CD', 'D')
>>> hd.draw(width=10, height=10, rank2size={2:(0.5,0.25), 1:(0.35,0.25)
, 0:(0.3,0.25)}, filepath='hd.pdf')

```



Some boundaries of the augmented manifold $\tilde{\mathcal{M}}$ are identifying in the sense of MBIM and some are not. So we would not need to enumerate all boundaries of $\tilde{\mathcal{M}}$; rather, *isocurve enumeration* have geodesics travel penpendicularly to the collapse manifold and never hit non-identifying boundaries. In the current example, $\tilde{\mathcal{M}}$ would collapse to a line segment in the (y_1, y_2) plane, hence boundaries AB and CD are identifying and boundaries BC and AD are not.

Example: The following codes illustrate how to do isocurve enumeration. Argument `v0idxs` provides the indices of "sloppy directions" and argument `yidxs` provides the indices of the original prediction vector, and together they ensure that our geodesics always travel perpendicularly to the collapsed manifold. Repeating such geodesic motions for different points on $\tilde{\mathcal{M}}$ would approach exhaustive enumeration of all isocurve classes.

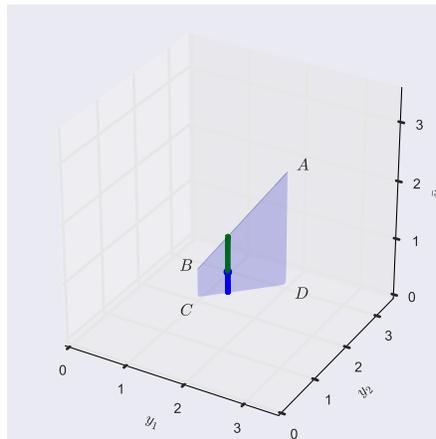
```

>>> gdss = pred.get_geodesics(p0=p0, v0idxs=[(-1,True),(-1,False)],
    yidxs=slice(2))
>>> gdss.integrate(tmax=5, dt=0.1)
>>> ys = butil.flatten(gdss.apply(lambda gds: gds.ytraj.values), 1)
  
```

```

>>> cs = butil.flatten([[c]*len(gds._ts) for gds, c in zip(gdss, list('
    bg'))], 1)
>>> pred.plot_image(ks, ks, alpha=0.2, pts=ys, cs=cs, xyzlims
    =[[0,3.5]]*3, linewidth=0, xyzticks=[range(4)]*3, xyzlabels=[r'$y_1$
    ',r'$y_2$',r'$\bar{y}_3$'])

```



Since it is clear that in our current example there is only one isocurve class, the one that connects boundaries AB and CD , one can imagine the Hasse-isocurve diagram the same as the Hasse diagram above but with an extra arching edge connecting AB and CD . In this case it is trivial to select "a disjoint covering set of identifying boundaries": either AB or CD would work. For more realistic examples, we have codes that do the tasks.

Example: The following codes select a covering, and disjoint if any, set of identifying boundaries upon isocurve enumeration.

```
>>> edges = [('AB', 'CD')]
>>> covers = hasse.get_vertex_covers(edges)
>>> covers
[('AB', ), ('CD', ), ('AB', 'CD')]
>>> [hasse.is_disjoint(cover, edges) for cover in covers]
[True, True, False]
```

CHAPTER 3

INFORMATION TOPOLOGY OF KINETIC MODELS OF METABOLISM

3.1 Introduction

In Chapter 2, we glanced through many examples from metabolism, always consisting of a toy model and some type of system behaviors (XT , XC , XE , JC or JE), and treated them with equal weight in our quest for a general understanding of system topology. Thus, one may have the impression that these behavioral spaces are of equal value for metabolic studies. This is true when one's focus is mathematical in nature: all behavioral spaces are created equal in the kingdom of mathematics. However, as soon as one ventures out of the kingdom into the real world and starts modeling data collected from expensive experiments probing whatever system behaviors that strike one's fancy, one realizes that there is a "caste system" on the behavioral spaces, based on their presumed "value" to us humans.

A general way to assess the value of something is through the intrinsic vs. extrinsic lens.¹ There are things of *intrinsic* value, that is, they are important to us and make our lives more exciting. And there are things of *extrinsic* value, that is, they are useful to us and make our lives easier by bringing the important things closer. A given thing, then, falls along these two dimensions: it can be important but not useful, useful but not important, both or neither.

¹It goes all the way back to Plato.

What does it look like if we cast a behavioral space along the two dimensions? Useful behavioral spaces typically are those we can easily probe experimentally, generating data with information, while important behavioral spaces typically are those we need to understand conceptually, producing knowledge with insight.

In metabolism, we have outlined in Chapter 1 the two dichotomies, steady state vs. transient and flux vs. concentration, and concluded that, for both conceptual and practical reasons, steady-state flux is the major phenotype. When it comes to experimental ease, one more dichotomy appears: *external* vs. *internal* perturbation, where perturbing quantities external to the system such as C is easier to perform than perturbing quantities internal to the system such as E .² For example, to change the extracellular concentration of glucose in a tissue culture, all one needs to do is to pour extra glucose into it, which is a high-school student project and a successful run barely makes it to the outreach newsletter; on the other hand, to change intracellular enzyme concentrations, one needs to spend months constructing transgenic lines, which is a PhD project and a positive result promises a paper in an academic journal.

Thus, along these dichotomies, we can attempt to describe some of the structures of "the space of behavioral spaces" *imposed by our human values and limitations*. As a consequence of its artificial nature, such a description is bound to be oversimplifying or even controversial, non-absolute or even misguided. But

²External and internal perturbations are sometimes also known as environmental and genetic perturbations, respectively. Not all internal perturbations are genetic, such as adding enzyme inhibitors to a metabolic network, hence our choice on terminology.

better to amicably provoke with reason and hopefully make progress than to be timidly overburdened by exactness and retreat to inaction.

The three dichotomies and the stance of metabolism research along them translate to the following conclusion. A behavioral space Z^U of a metabolic model is:

- Important when $Z = J$ (steady-state flux)
- Unimportant when $Z = X$ (steady-state concentration) or $U = T$ (dynamic behavior)
- Useful when $U = C$ (external perturbation)
- Unuseful when $U = E$ (internal perturbation) or $U = T$ (dynamic behavior)

Table 3.1 summarizes these value judgments.

Table 3.1: A partition of the set of behavioral spaces in Table 1.1 along importance and usefulness in metabolism.

		Intrinsic	
		Important	Unimportant
Extrinsic	Useful	J^C	X^C
	Unuseful	J^E	X^T, X^E

How does such a conceptual partition inform our practice? It is simple: importance determines our goals, and usefulness dictates our means. We harness the power provided by useful behaviors to tame important behaviors. This idea introduces a notion of *directed pairing* among behavioral spaces and suggests what

we call the *information flow perspective*: information flows from a useful behavioral space to an important one.³

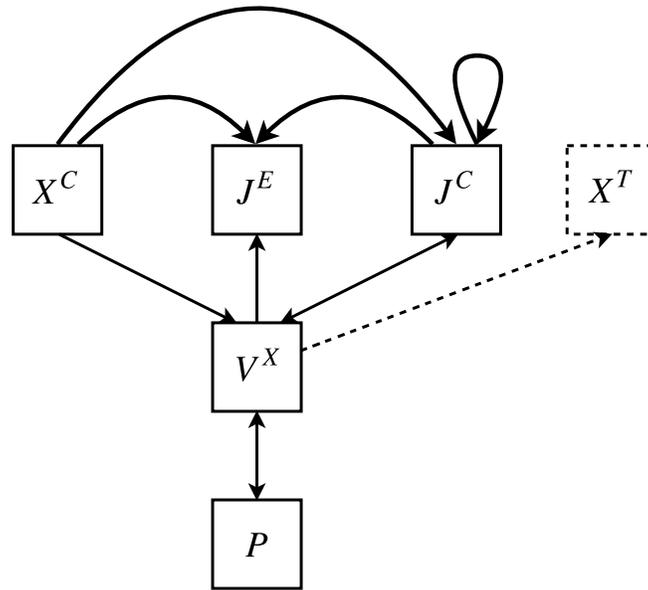


Figure 3.1: Information flows between a directed pair of behavioral spaces. Judgements on importance and usefulness of behavioral spaces impose directionality on the edges between parameter space P and behavioral spaces B .

Fig. 3.1 explains the notion and perspective. First, all behavioral spaces are connected to parameter space P through their associated behavioral maps, and information can flow both ways: from P to B in the form of prediction and from B to P in the form of parameter estimation (Fig. 2.4 (a)). Second, our judgements on the importance and usefulness of them impose an *intended direction of information flow*: information flows from useful behavioral spaces to P , and from P

³It is in this process where information transforms into the wanted insight.

to important behavioral spaces, as denoted by the directed edges in the figure; for behavioral spaces both important and useful, the flow is bidirectional and the edge is double-arrowed; behavioral spaces both unuseful and unimportant have no edges with P and do not need to make it to the figure. Third, the direction of information flow from a useful behavioral space B , through parameter space P , to an important behavioral space B' induces a directed pairing relation among the behavioral spaces, represented by the arching edges in the figure; we call such (B, B') a *directed pair*.^{4 5}

Now we can interpret some familiar modeling practice in systems biology from the perspective: traditional kinetic modeling of signal transduction networks and gene regulatory networks, where system dynamics are of primary concern, represents an information flow from VX to XT ; traditional kinetic modeling of metabolic networks, where flux controls are of primary concern, represents an information flow from VX to JE ; using metabolomic data that are typically collected by varying C to understand other important system behaviors represents information flows from XC to JE or JC . In fact, the perspective describes our entire research program: any modeling practice that has ever been done or will ever

⁴Formally, one may say that a directed bipartite graph between P and the set of behavioral spaces $\{B\}$ induces a directed graph on $\{B\}$ (possibly cyclic).

⁵What is the "information" that is flowing here? Roughly speaking, a probability distribution in P or B measuring the uncertainty of our understanding. How does it "flow"? Measurements in one B makes its distribution sharper, which in turn makes the distributions in other spaces sharper through the maps connecting them. Hence, while the word "flow" implies some process continuous in time, typically information only gets passed among spaces in discrete installment as measurements are discrete in time. If we zoom out, however, and look at human endeavors at a grand temporal scale, we can in fact speak of an information flow, say, from VX to XT at some number of bits per year.

be done represents some information flow from a useful B to an important B' . The status of a behavior space in its importance or usefulness may change: technological breakthroughs may replace old useful behavioral spaces with new ones, and scientific discoveries may refine important behavioral spaces from time to time. Whichever useful and behavioral spaces get chosen, the nature of information flow between them in any modeling does not change.

The perspective also guides our current research program. For the four important or useful behavioral spaces we currently have for metabolism, we need to do two things: to understand their individual relations with P (that is, the behavioral maps), and to understand how they pair (that is, the compositions of the behavioral maps). This sounds straightforward enough, but count them and we have four behavioral maps and six possible compositions. We are only beginning to think about them so it is hard to attempt them all at once. How to choose to start? First, the component behavioral map $G : P \mapsto V^X$ is the simplest, and a characterization of it should help any investigation of system models due to compositionality of system topology. Second, J^C behavioral space is special in that it is the only useful and important space, and it turns out that J^C harbors some mathematical richness due to some of its properties. Lastly, X^T behavioral space, while not important or useful for metabolism, has been studied much elsewhere in systems biology, and it admits some further decomposition that may help its boundary enumeration. For these reasons, we will focus on these three behavioral spaces.

3.2 Rate law topology

Our discussions in Chapter 2 have highlighted the importance and usefulness of topological characterizations of the component models in achieving an information topological understanding of any system models constructed using the component models. In our studies of metabolic models, the task amounts to characterizing the rate laws, which we have already done and discussed much in our previous general discussions on information topology. Here we recapitulate some of the key points and note a few more.

First, as component models, rate laws are usually simple and comprehensible, free of structural nonidentifiability, and a full topological characterization relatively easy to achieve.

Second, the boundary structure of rate laws in the Michaelis-Menten family harbors a remarkable degree of regularity: applying MBAM to them involves only three atomic reductions (unidirectionalization, linearization and multisaturation) and combinations of the three reductions lead to various reduced rate laws. Understanding such regularity enables us to automate the boundary enumeration for any Michaelis-Menten rate laws, facilitating a full information topological understanding of the entire rate law family.

Third, Michaelis-Menten rate laws, in their reversible form, are by construction symmetric with respect to the substrates and products, and this induces a corresponding symmetry in the manifold which can be formalized using the model

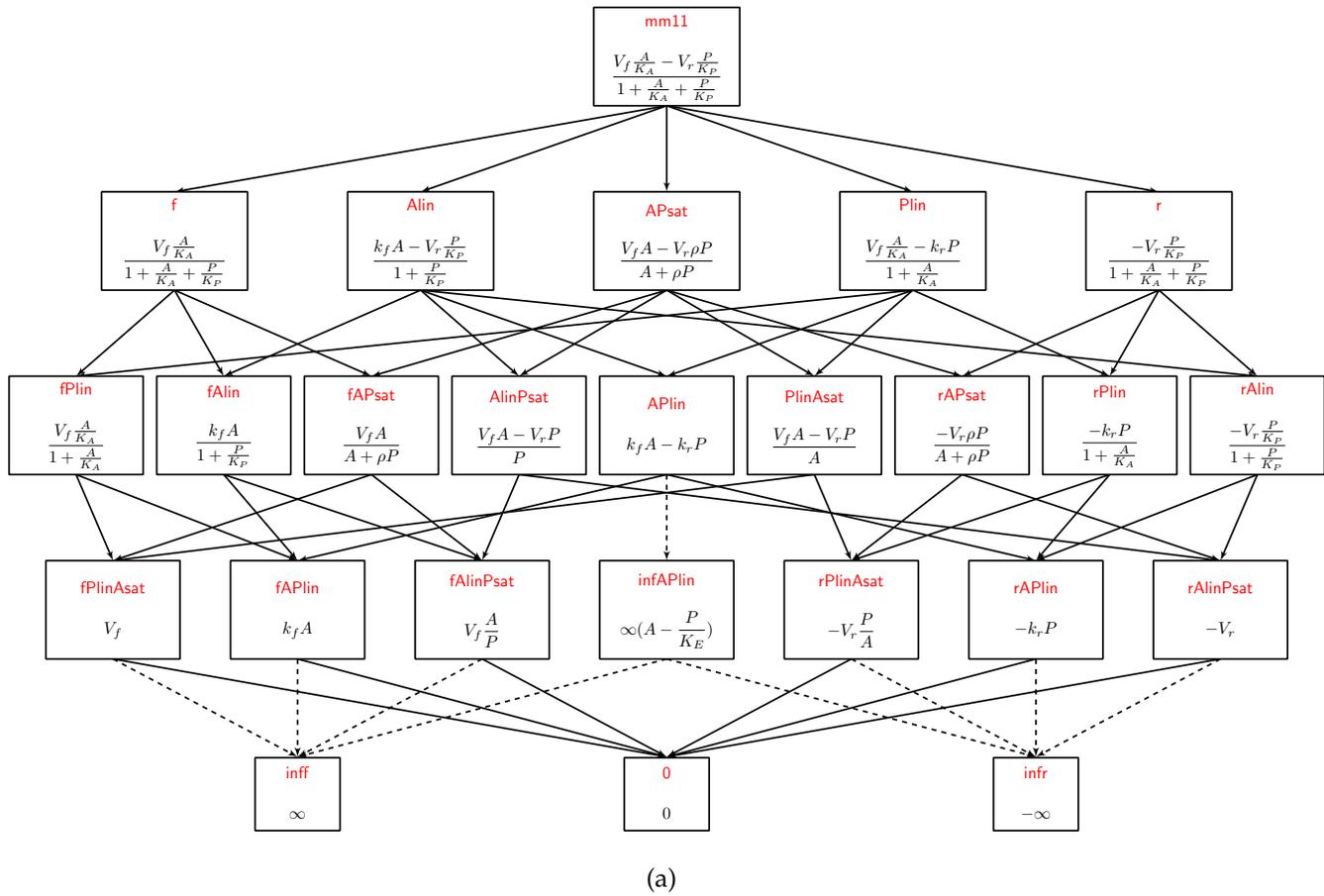
map M (Section 2.4.5).

Fourth, manifolds of Michaelis-Menten rate laws are unbounded in directions that correspond to their maximal velocities V , and we suggest viewing them as a type of generalized boundaries and treating them similarly in the system-component formulation, as they often get *compactified* by the coupling map and turn into boundaries of system models (Sections 2.1, 2.6, 3.4.3).

Fifth, all Michaelis-Menten rate laws (including those with the Haldane constraint) have the 0 rate law as one of their boundaries. This simple fact has the implication that in studying any networks, removing branches and reactions from the system (for example, removing the connection to TCA cycle in studying glycolysis) is a boundary approximation and a model of the resulting simplified system is connected to a model of the full system through boundary approximations (eg, they are path-connected on a Hasse diagram).

Lastly, occasions often arise where we have to use the *interiors*, rather than boundaries, of a rate law to represent a model. Unlike taking a manifold's boundary which shares all its own boundaries (that is, the boundaries of the boundary) with the manifold, slicing a manifold through its interior can disrupt its boundary structure to varying degrees, depending on the way of slicing. For example, a Michaelis-Menten-Haldane rate law, which uses the equilibrium constant between substrates and products of a reaction to introduce a thermodynamic constraint into the rate law (Section 1.3), amounts to taking a slice of the Michaelis-Menten manifold through *fixing the ratios* of certain forward and reverse kinetic

parameters, and its topology shares *no* boundaries with its full Michaelis-Menten counterpart excepts the 0 rate law. This is shown in Fig. 3.2.



However, when we resolve structural nonidentifiability and are required to use a currying reparametrization (that is, *fixing the values* of some parameters) to preserve the localness of the parametrization, the resulting slice of the manifold shares some boundaries with the manifold itself, as is shown in Fig. 3.3. It plots one currying reparametrization of the full Michaelis-Menten rate law in Fig. 3.2 which fixes the forward maximal velocity to one.

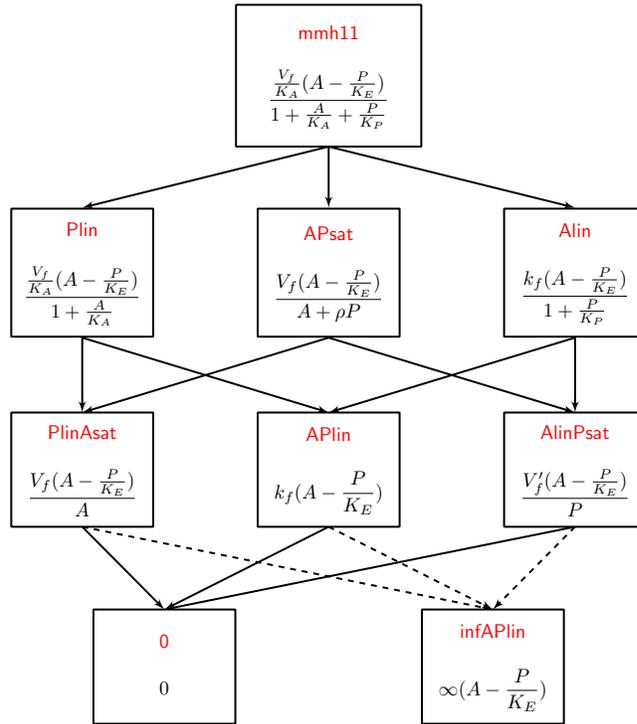


Figure 3.2: Hasse diagrams of a Michaelis-Menten rate law and the same one with the Haldane constraint. (a) The diagram of Michaelis-Menten rate law, a re-plot of the one in Fig. 2.8 using VK parametrization to facilitate interpretation. (b) The diagram of Michaelis-Menten-Haldane rate law.

Such situations of interior-slicing of a rate law manifold to construct a system model are expected to happen quite often, since any parameter enjoying high confidence in its value can be treated this way. We also recognize that the interior-slicing by a Michaelis-Menten-Haldane rate law is of a different origin: theories tell us that there are some interdependencies between parameters, which translate to some algebraic constraints in the parameter space (in this case, the ratios of

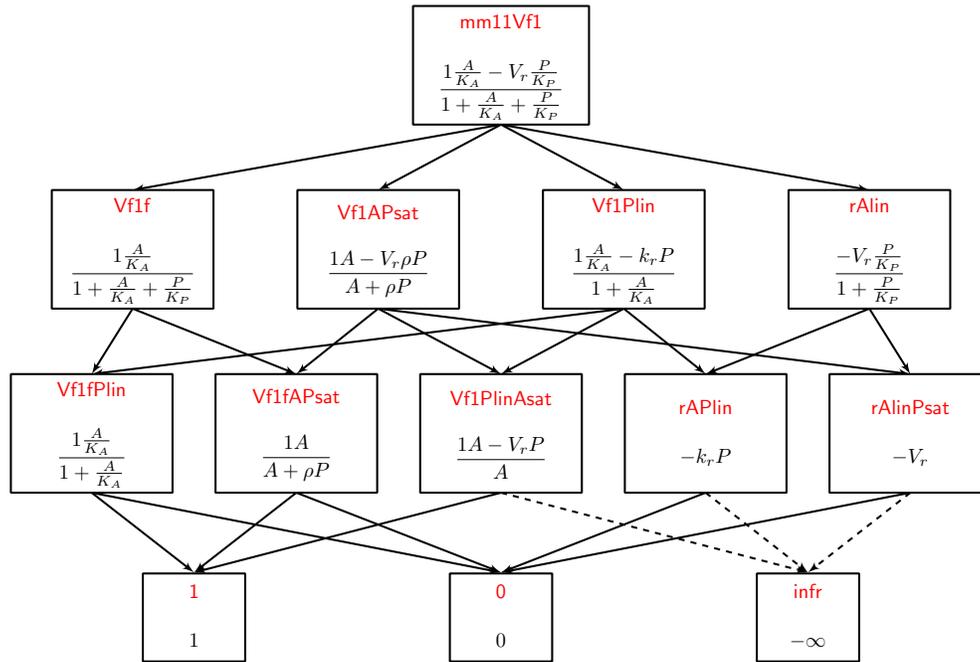


Figure 3.3: Hasse diagram of the same Michaelis-Menten rate law in Fig. 3.2 with $V_f = 1$.

some parameters should be fixed). We speculate that, in general, sliced submanifolds resulting from value fixing share more of their topological structures with their parental full manifolds than sliced submanifolds resulting from algebraic constraints. In either case, however, topological characterizations of the sliced submanifolds need to be started anew (which, as we argued, should not be too much of a hassle due to their nature as component models).

3.3 Kinetic behaviors

As we have discussed in Section 2.3, the system-component formulation entails a decomposition of a system behavioral map F into two parts, one from the component models and one from their coupling. In the context of reaction network models, the component models are invariably the reaction rate laws and the coupling map depends on the kind of system behaviors one studies (eg, Section 1.5). If one studies the kinetic behaviors of a reaction network, then the coupling map admits a further decomposition, which can shed further light on our understanding of the behaviors and their boundary approximations.

To see this, let us look at the explicit formula for the kinetic behaviors, $x(t)$, of a reaction network:

$$x(t) = x(0) + \int_0^t Nv(x(\tau), p) d\tau$$

The equation tells how $x(t)$ comes into being: first we have the reaction rate laws $v(x, p)$ which are our component models, then we form a vector field $Nv(x, p)$ for x by multiplying the stoichiometry matrix N with the rate laws, and lastly we get the trajectories $x(t)$ by integrating the vector field. Symbolically, we have:

$$p \mapsto v(x) \mapsto Nv(x) \mapsto x(t).$$

It is the second and third steps that together constitute our coupling map, or conversely, we can *decompose the coupling map into the vector field formation from reaction rate laws and its integration*.

A natural implication of this further decomposition, inline with our previous thinking, is that emergent boundaries now fall into two classes: those that arise in the vector field formation but not integration and those that arise only in the integration. We call boundaries of $Nv(x, p)$ **vector field boundaries** and boundaries of $x(t)$ **trajectory boundaries**. The first class of emergent boundaries, then, correspond to vector field boundaries that are not rate law boundaries, and we can call them *emergent vector field boundaries*; the second class correspond to trajectory boundaries that are not vector field boundaries, and we can call them *emergent trajectory boundaries*. An explanation of the origin of emergent vector field boundaries that is similar to the one for emergent boundaries in general (Section 2.4.3) can be given: Dv falls into the null space of N along the parameter limit for an emergent vector field boundary.

This further decomposition also brings practical benefits in computation. Vector field boundaries can be explored without integration and the fact that vector fields are *rational functions* in this case (N is a linear transformation and $v(x, p)$ as members of the Michaelis-Menten rate law family are rational functions) makes their boundary enumeration especially straightforward. Boundaries specific to trajectories, on the other hand, by definition would only emerge upon integration, hence finding them usually involves numerical integration which drastically

increases the computational load.

Below we give one example for each of the two classes of emergent boundaries.

Example 1: An emergent vector field boundary

Given a pathway of two reactions, the sole metabolite X has its dynamics described by:

$$\frac{dX}{dt} = v_1 - v_2 = (k_1 C_1 - k_2 X) - \left(\frac{VX}{K + X} \right),$$

where $k_1 C_1 - k_2 X$ is the rate law of the first reaction (mass action) and $\frac{VX}{K + X}$ rate law of the second (irreversible Michaelis-Menten). Without integration, one can find the following boundary for the vector field:

$$\frac{dX}{dt} = \frac{k_1 C_1 (K + X) - VX}{K + X} - k_2 X = \frac{C_1 k_1 K + (C_1 k_1 - V)X}{K + X} - k_2 X \rightarrow \frac{q_1 + q_2 X}{X} - k_2 X$$

as $K \rightarrow 0$, and $k_1, V \rightarrow \infty$.

The parameters go to the limit in such a way that $k_1 K \sim O(1)$ (meaning the product remains on the order of one) when k_1 diverges and K vanishes, and similarly $C_1 k_1 - V \sim O(1)$ when both k_1 and K diverge (C_1 is a constant on the order of one). Since the limit involves parameters from both reactions, it is an emergent boundary.

Example 2: An emergent trajectory boundary

In the same setting of a two-reaction pathway as in the previous example, we keep the first mass-action rate law and change the second one to the zero rate law which amounts to inactivating the second reaction. Now the ODE simply reads:

$$\frac{dX}{dt} = v_1 - v_2 = k_1 C_1 - k_2 X.$$

Enumerating the vector field boundaries uncovers two: $k_1 \rightarrow 0$ and $k_2 \rightarrow 0$. However, enumerating the trajectory boundaries uncovers one more: $k_1, k_2 \rightarrow \infty$ with $\frac{k_1}{k_2} \sim O(1)$, which is an emergent trajectory boundary.

The emergent trajectory boundary in the example represents a quasi-equilibrium approximation (QEA) (Section 1.3). Mathematically, what this approximation does to our ODE is to change it to an algebraic equation: now $X(t)$ is described by $X(t) = \frac{k_1 C_1}{k_2} = K_E C_1$. In general, applying QEA to a system of ODEs turns part of the equations to algebraic ones, which now constitutes a system of differential-algebraic equations (DAEs). This observation leads to the following **conjecture**: *emergent trajectory boundaries involve changing the system of ODEs to a system of DAEs*. We speculate that if a system of ODEs stay as ODEs upon a boundary reduction, then the boundary should already be present in its vector field which preserves the ODE nature of the equations. If such a conjecture is correct, then all emergent trajectory boundaries involve *singular perturbations* to a system of ODEs. In chemistry, two types of singular perturbations of ODEs are commonly studied: QEA and quasi-steady-state approximation. Are there more types? We do not know. If there are not, then enumerating such singular pertur-

bations boils down to applying the two well-understood approximations, which would give us all the emergent trajectory boundaries.

We conclude this section with an example of constructing part of the system topology for a two-reaction pathway model with Michaelis-Menten rate laws describing kinetic behaviors, which provides some context to the above two example boundaries. We follow the algorithm described in Section 2.6 for constructing the system topology, so the following concrete model also illustrates some aspects of the algorithm execution. The ODE reads:

$$\frac{dX}{dt} = v_1 - v_2 = \frac{V_1 \frac{C_1}{K_1} - V_2 \frac{X}{K_2}}{1 + \frac{C_1}{K_1} + \frac{X}{K_2}} - \frac{V_3 \frac{X}{K_3} - V_4 \frac{C_2}{K_4}}{1 + \frac{X}{K_3} + \frac{C_2}{K_4}}.$$

As the first step of the algorithm, we test the model for structural nonidentifiability: the behavioral map for the model has a three-degree SN. One natural approach is to inspect the vector field, which reveals a reparametrization that is free of SN:

$$\frac{dX}{dt} = \frac{V_1 \frac{C_1}{K_1} - V_2 \frac{X}{K_2}}{1 + \frac{C_1}{K_1} + \frac{X}{K_2}} - \frac{V_3 \frac{X}{K_3} - V_4 \frac{C_2}{K_4}}{1 + \frac{X}{K_3} + \frac{C_2}{K_4}} = \frac{a_1 X^2 + b_1 + c_1}{a_2 X^2 + b_2 + 1},$$

where the five new parameters a_1 through b_2 are some combinations of the original parameters. However, we resist the temptation of following this approach for two main reasons: the reparametrization is global and hard to interpret (eg, $a_1 = \frac{V_2}{K_2 K_3} + \frac{V_3}{K_2 K_3}$ and $c_1 = -\frac{C_1 V_1 K_4 + C_2 V_4 K_1 + C_1 C_2 (V_1 + V_4)}{K_1 K_4}$!), and all our

knowledge of the component rate laws would become irrelevant upon adopting the reparametrization. Instead, we follow closely the guidelines of the algorithm and strive to maintain the system-component formulation of the model. To get rid of the SN, we first learn from our previous experience of resolving SN in similar cases (Section 2.5.2, Example 4) and spot that, since C_1 and C_2 are constant, both rate laws can be reparametrized that effectively become their own boundary approximations and remove two degrees of SN simultaneously; eg, for the first rate law, $\frac{V_1 \frac{C_1}{K_1} - V_2 \frac{X}{K_2}}{1 + \frac{C_1}{K_1} + \frac{X}{K_2}}$ becomes $\frac{k_1 C_1 - V_2 \frac{X}{K_2}}{1 + \frac{X}{K_2}}$ upon dividing all terms by $1 + \frac{C_1}{K_1}$, which is equivalent to its boundary approximation assuming C_1 to be in the linear regime of the Michaelis-Menten rate law. For the remaining degree of SN, we apply MBIM. Fig. 3.4 shows the Hasse-isocurve diagram for the isocurve enumeration; it shows that the degenerate system model can be identified by two disjoint component boundaries: either the two red nodes (each has one reaction going irreversibly forward) or the two white nodes (each has one reaction going irreversibly backward) works. Upon choosing the red nodes, we have effectively also chosen all their descendants down the Hasse diagram (also marked red in the figure). The stitchwork step in MBIM corresponds to removing the shared boundary between the two chosen identifying boundaries (the black node). This completes the MBIM resolution of model SN as the first step in the algorithm to construct model topology. For the rest of the steps in the algorithm, we defer their illustrations to Section 3.4.3, where we construct a system topology in full.

The two example emergent boundaries in this section are highlighted in

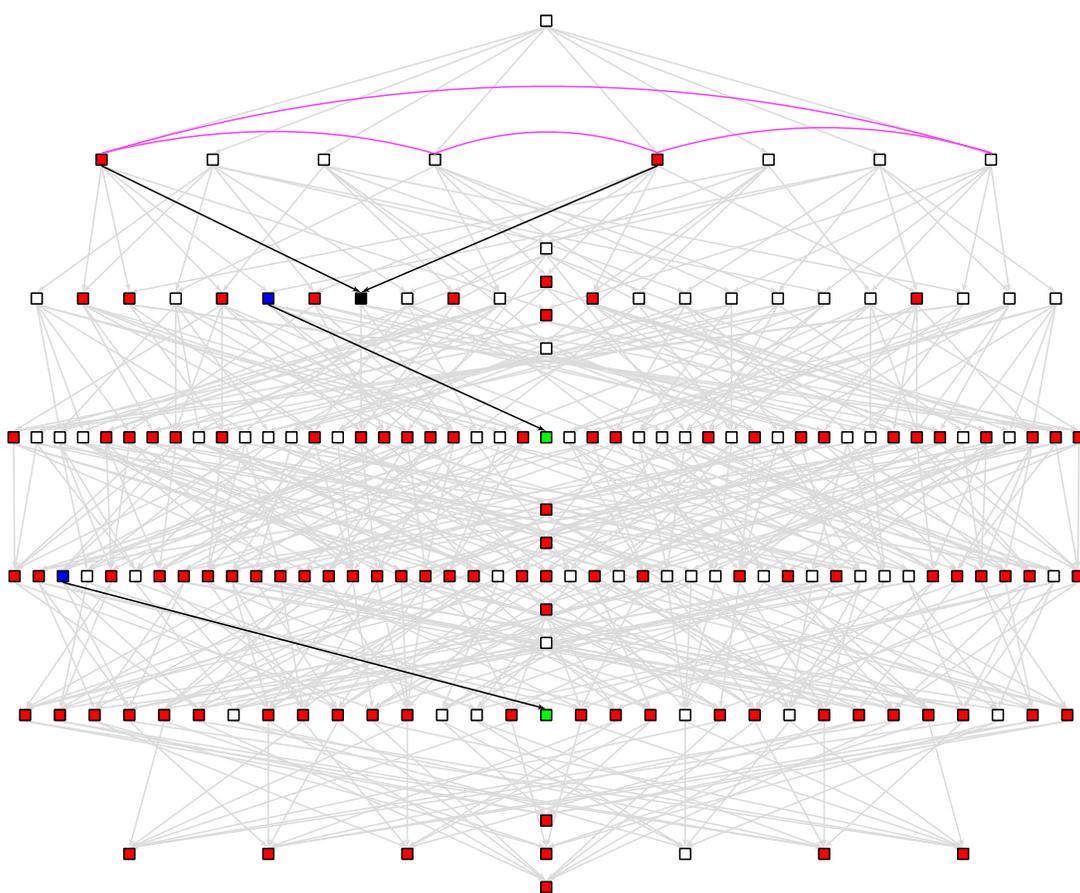


Figure 3.4: Partial construction of the topology for model $(\text{path2mm}, \text{XT})$, the kinetic behaviors of a two-reaction pathway with Michaelis-Menten rate laws. The Magenta curves correspond to the isocurve edges in a Hasse-isocurve diagram, and encode the resolution of SN by MBIM. All the red nodes represent the component product topology upon selecting an identifying set of boundaries in MBIM. The black node represents the shared boundary between the two disjoint identifying boundaries which corresponds to system manifold interior and is removed by the stitchwork step in MBIM. The two pairs of blue and green nodes represent the two emergent boundaries discussed as examples in the main text: the upper pair represents the emergent vector field boundary in Example 1, and the lower pair represents the emergent trajectory boundary in Example 2. Some nodes are stacked in the middle to highlight the bilateral symmetry of the model.

Fig. 3.4: Example 1 of an emergent vector field boundary corresponds to the upper pair of blue and green nodes, and Example 2 of an emergent trajectory boundary corresponds to the lower pair. As emergent boundaries, the two nodes are not present in the component product topology, and have to be added to the Hasse diagram and subsequently explored independently.

3.4 J^C behaviors

When we established the J^C behavioral space as an important and useful one, we based ourselves on the current consensus that steady-state flux J is the metabolic phenotype and C is easy to perturb. Whatever changes the future may bring to these two recognitions, there are two more properties of J^C that will likely make it everlasting in the study of metabolism, as these two properties are mathematical in nature: the system J^C behavioral space is **similar** to component V^X behavioral space, and the system behavioral map $F : P \rightarrow J^C$ admits a **modular** decomposition. We discuss them in turn.

3.4.1 Similarity

To understand similarity, let us look at a simple example (`path2mah`, `JC`). It has two component models, $v_1 = k_1(C_1 - X)$ and $v_2 = k_2(X - C_2)$, which describe the rate laws of the two reactions, that is, how their rates depend on their reactant

concentrations. The system model, as we have seen in Sec. 1.6, can be solved to be $J = \frac{k_1 k_2}{k_1 + k_2}(C_1 - C_2)$, which describes the same type of behaviors: how the rate of the network depends on the concentrations of the two relevant reactants, C_1 and C_2 . In fact, the interpretation of such a form of $J(C)$ mentioned in Section 1.6 is most suggestive: the whole pathway is equivalent to a single reaction with the harmonic mean of k_1 and k_2 as the rate constant and the average chemical potential between C_1 and C_2 as the driving force. Therefore, the component models and the system model simply describe the same type of behaviors at two different scales, and we call them similar, as a general notion of two mathematical objects sharing some scale-independent properties.⁶

In general, JC behaviors of any metabolic network has this property: it describes rate-law-type behaviors at a system scale. This gives rise to the following interpretation: a metabolic network acts as a **generalized reaction**, with $J(C)$ describing its **generalized rate law**. This interpretation is satisfying, as it assigns a simple and familiar meaning to a complicated and unfamiliar context. Also, as is described in Sec. 1.3, one common way of deriving rate laws is to apply quasi-steady-state approximation (QSSA), which sets some reactants in a network to steady state; JC behaviors, then, can be interpreted as applying QSSA to a large network to get its rate law, just like applying the approximation to a small network of an enzyme-catalyzed reaction to get its rate law. Moreover, this interpre-

⁶The functional form of the two behavioral maps happens to be the same in the given example, but this is due to the linearity of one-substrate one-product mass action rate laws. In general, rate laws are nonlinear and the functional form of the behavioral map does not stay invariant across scales. In fact, understanding the connections in functional form between scales is an important open challenge (Section 3.4.3).

tation accords with the essential functions of metabolic networks as we described in Sec. 1.1: conversion of food molecules to harvest energy and produce biomass, where food molecules, energy sink and terminal biomass can be thought as C and the conversion efficiency is described by the generalized rate laws. Admittedly, it misses the other fundamental aspect of metabolic networks, that is their regulation: J depends not only on C , but also on factors that are under active cellular regulation, such as enzyme concentrations E (which gives rise to JE behaviors). However, as a first approximation of metabolic behaviors, this interpretation enjoys some elegant simplicity.

3.4.2 Modularity

To understand modularity, let us look at the example of a three-reaction pathway (path3) in Fig. 3.5. We have a model for the pathway in which the three reactions (R_1 , R_2 and R_3) have rate laws $v_1(C_1, X_1)$, $v_2(X_1, X_2)$ and $v_3(X_2, C_2)$, respectively. Interested in $J(C_1, C_2)$, we can solve the system of equations $v_1(C_1, X_1) = v_2(X_1, X_2)$, $v_2(X_1, X_2) = v_3(X_2, C_2)$ for X_1 and X_2 *simultaneously*, and plug them back in to get J . Alternatively, we can also solve X_1 and X_2 *sequentially*: first, we solve a single equation $v_1(C_1, X_1) = v_2(X_1, X_2)$ for X_1 in terms of C_1 and X_2 ; second, plugging the solved X_1 in v_2 turns v_2 into a function of C_1 and X_2 , and since now v_2 results from solving both v_1 and v_2 , we rename it to \tilde{v}_{12} , $\tilde{v}_{12} = \tilde{v}_{12}(C_1, X_2)$; last, we can solve the remaining equation with the new form of v_2 , $\tilde{v}_{12}(C_1, X_2) = v_3(X_2, C_2)$, for X_2 , and plug the solved X_2 in either rate law to get J . More concretely, suppose

all three reactions have mass-action-Haldane rate laws: $v(A, P) = k(A - P/K_E)$. We know from previous analysis that in this case, the standard simultaneous solving procedure yields $J = \frac{k_1 k_2 k_3}{k_1 k_2 + k_1 k_3 + k_2 k_3} (C_1 - C_2)$ (Section 1.6). Now one can verify that in the sequential solving procedure, $\frac{k_1 k_2}{k_1 + k_2} (C_1 - X_2) = k_3 (X_2 - C_2)$ yields the same J .

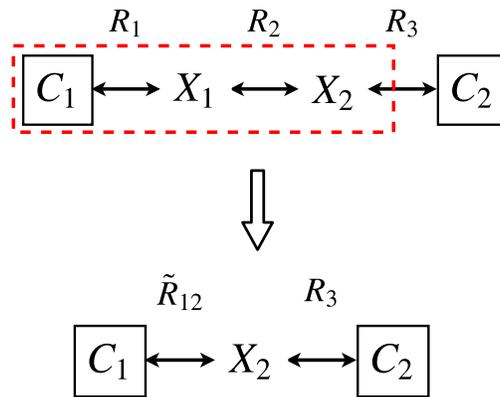


Figure 3.5: Modularization of a three-reaction pathway `path3`. The first two reactions in the red dashed box form a module, and can be conceptually thought as a single reaction \tilde{R}_{12} , thereby effectively coarse-graining the `path3` (top) to a two-reaction pathway `path2` (bottom).

What is the implication of the equivalence of such a sequential solving procedure? The key lies in an interpretation of \tilde{v}_{12} : it comes from a small network of two reactions (reactions R_1 and R_2) with the internal species X_1 in the small network eliminated; echoing our network-as-reaction notion discussed in similarity, we can interpret the small network of R_1 and R_2 as a generalized reaction \tilde{R}_{12} with \tilde{v}_{12} its generalized rate law. Since the scale of our generalized reaction consisting of

part of the network lies between that of a single reaction and the whole network, we call it a **module**, and its generalized rate law **module rate law**.

It is trivial to understand the modular structure for the three-reaction pathway: R_1 and R_2 form a module, and so do R_2 and R_3 . For a realistic network of decent size and complicated wiring, it is less straightforward; to develop a definition and a module-searching algorithm, we need to distill the essence of a module. The first essential feature already present in our example is that module \tilde{R}_{12} has its internal species X_1 *eliminated*. This is the very essence of *coarse-graining* ([97]; eg, p.131) and what makes modular thinking powerful: now thinking about a network as consisting of modules rather than reactions, we have fewer complexities to keep track of. In fact, what we consider as "reactions" are usually modules: Section 1.3 shows how an enzyme-catalyzed reaction with a simple net stoichiometry (eg, $A + B \leftrightarrow P + Q$) often involves a complicated network consisting of many elementary steps and a "reaction rate law" is derived by coarse-graining out its internal complexities using systematic means.⁷ When the systematic means is QSSA, then it is the same coarse-graining machinery operating at all three scales.

The second essential feature is *flexibility* of the scale of a module: it can eliminate one species, as in our example, or more species. Imagine a pathway of not only three, but many, reactions: upon lumping R_1 and R_2 into a module \tilde{R}_{12} , one can keep lumping the module with R_3 to get a new module \tilde{R}_{123} which now has

⁷It can be thought, then, that a reaction rate law, a module rate law and a network rate law have the complexities of, respectively, enzyme-reactant complexes, internal species in a module and all species in a network, coarse-grained and eliminated.

further eliminated X_2 , and so on. By spanning the whole range of scale between a single reaction and the whole network, it allows for successive coarse-grainings of a network model.

The third essential feature is a sense of *connectedness* between the species it eliminates. A module consisting of, say, R_1, R_2, R_4 and R_5 in a many-reaction pathway, would have eliminated multiple species, that is, X_1 and X_4 , but somehow we are prone to think of it as two disjoint modules, each consisting of two reactions, rather than a single module, because the eliminated X_1 and X_4 are not connected. What sense of connectedness are we using here? Intuitively, two species are connected if they participate in the same reaction, or formally, they are incident to the same reaction node in the species-reaction bipartite graph.

Now we can combine all three essential features to form a definition of a module.

Definitions:

Given a reaction network, a *connected set of species* is a subset of species in which any two species are connected by a reaction path, and a **module** is a subset of reactions which, when taken as a network in itself and solved for steady states, would eliminate a connected set of species. By convention, we also regard an individual reaction to be a module in itself.

The following algorithm puts the notions on a technically solid ground.

Module-searching algorithm:

Given a reaction network, its bipartite graph representation \mathcal{G} with X and R denoting its species and reaction sets induces an undirected graph \mathcal{G}' on the species set X : species X_i and X_j are connected by an edge in \mathcal{G}' if they are both connected to a reaction in \mathcal{G} . A subset of species $\alpha \subset X$ is called connected if α forms a connected subgraph in \mathcal{G}' .

A given connected subset of species $\alpha \subset X$ has an associated subset of reactions $\beta \subset R$: all species in α are connected to some reactions in β , and vice versa. The claim is that such a β constitutes a module. To see why, one just needs to realize that β represents the set of all reactions that α participate in, and solving β to steady states to get the module rate laws represents the first steps in the sequential solving procedure illustrated above; since α is by definition connected, β is a module.

A simple example illustrates the algorithm. Consider the network in Fig. 3.6(a). One can easily verify that there are seven connected sets of species, listed below together with their associated modules.

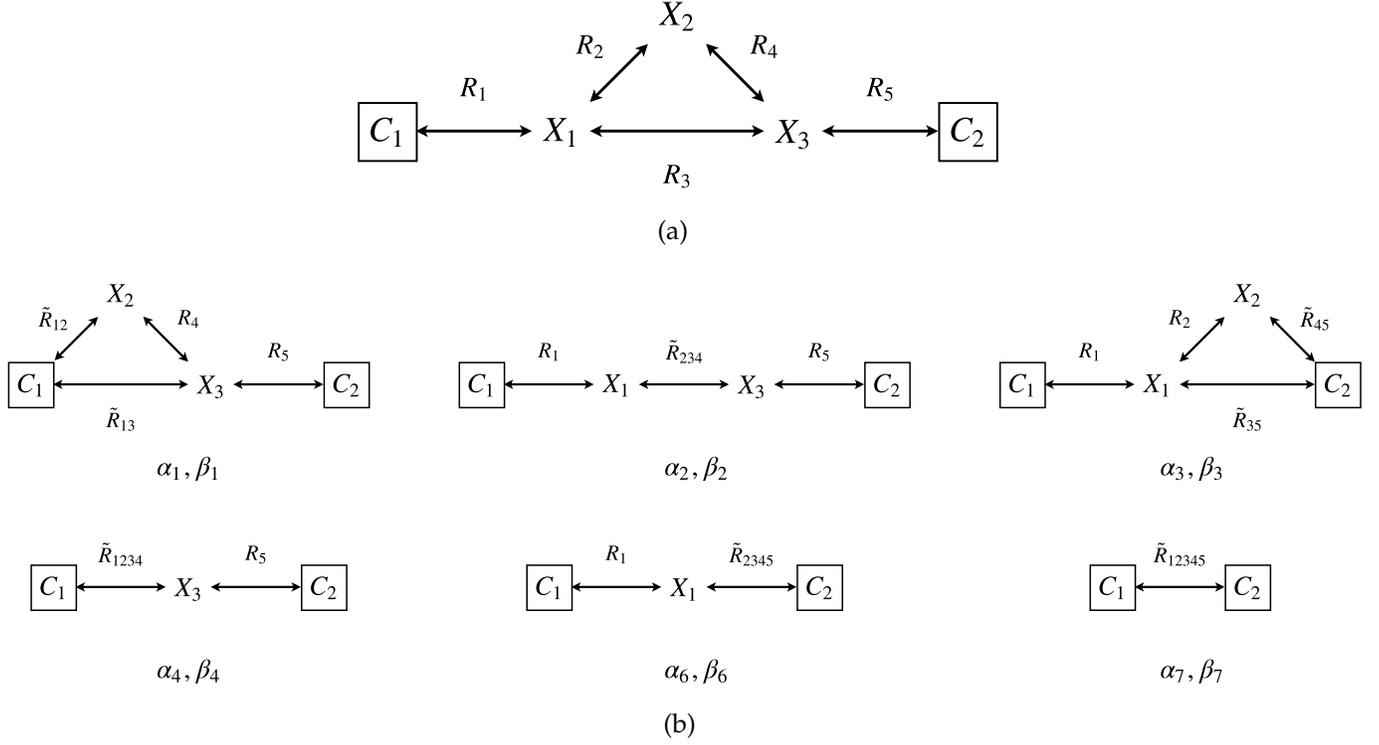


Figure 3.6: Modularization of an example network. (a) The example network. (b) Applying module-searching algorithm to the example network yields the six modules.

$$\begin{aligned}
 \alpha_1 = \{X_1\} &\Rightarrow \beta_1 = \{R_1, R_2, R_3\} \\
 \alpha_2 = \{X_2\} &\Rightarrow \beta_2 = \{R_2, R_4\} \\
 \alpha_3 = \{X_3\} &\Rightarrow \beta_3 = \{R_3, R_4, R_5\} \\
 \alpha_4 = \{X_1, X_2\} &\Rightarrow \beta_4 = \{R_1, R_2, R_3, R_4\} \\
 \alpha_5 = \{X_1, X_3\} &\Rightarrow \beta_5 = \{R_1, R_2, R_3, R_4, R_5\} \\
 \alpha_6 = \{X_2, X_3\} &\Rightarrow \beta_6 = \{R_2, R_3, R_4, R_5\} \\
 \alpha_7 = \{X_1, X_2, X_3\} &\Rightarrow \beta_7 = \{R_1, R_2, R_3, R_4, R_5\}
 \end{aligned}$$

Note that α_5 and α_7 give the same module. It is easy to understand why from Fig. 3.6(a): in order to have X_1 and X_3 reach steady states, X_2 has to as well. For networks of realistic size, this degeneracy can be great and many connected sets of species can give rise to the same modules. Fig. 3.6(b) shows the modularized networks resulting from the six distinct modules.

Having pinned down module definition and detection, we need to carefully characterize its properties. We have emphasized its coarse-graining power, which is the motivation for us to get this far in the first place. Coarse-graining is most visible in network topology: for example, the network topology in Fig. 3.6(a) is simplified successively in Fig. 3.6(b). What about system behaviors, in this case, JC behaviors? Modules would not be useful if JC behaviors of a network change greatly upon modularization. One would expect that the way we define modules preserves the network fluxes in some ways, and we have hinted at this possibility in the `path3` example in Fig. 3.5. This notion of the preservation of network fluxes upon modularization turns out to be basically correct, with one subtlety. The precise mathematical statement is the following: given a reaction network with a steady-state concentration vector x , modularize it and eliminate some species; concentrations in x of the remaining species constitute *a locally stable fixed point* of the modularized network. This means that *the steady states of a reaction network persist upon modularization only locally*. That they persist is easy to understand: we can first have a network reach steady states for different C and at those steady states construct the module rate laws *a posteriori*, which by construction have the same steady states; in the `path3` example in Fig. 3.5, different C_1 and C_2 give different

J and X , which induce a module rate law for \tilde{R}_{12} , $J(C_1, X_2)$. That the persistence holds only locally needs some explanation. As an example of the lack of global persistence, if we *integrate* the dynamics for both the original and the modularized networks, they may end up in different steady states, depending on how the initial conditions are chosen. This is explained in Fig. 3.7.

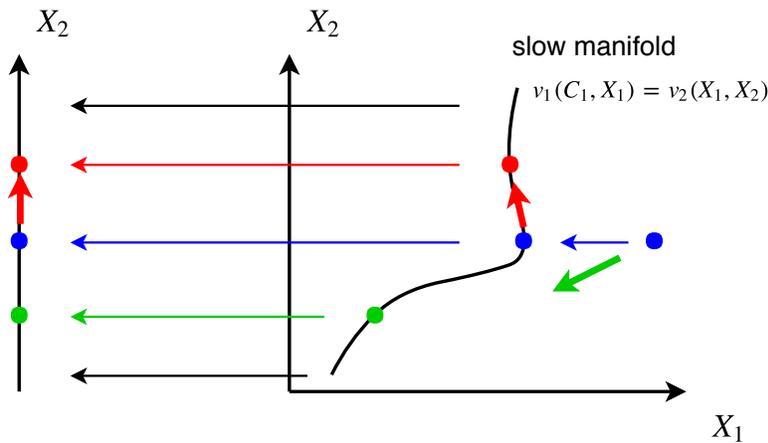


Figure 3.7: The steady state reached through integration may change upon modularizing path3 to path2 in Fig. 3.5. The dynamics of path2 in its reduced state space (the X_2 line on the left) is equivalent to the projection (the horizontal arrows pointing left) of the dynamics on the slow manifold (the black curve) of the original network in the full state space (the X_1 - X_2 plane on the right), defined by the equations that give rise to the module rate law, onto the reduced state space. As long as the initial condition and its projection onto the slow manifold (the two blue dots in the X_1 - X_2 plane) belong to different basins of attraction, they would reach different steady states (the green and red dots in the X_1 - X_2 plane), thereby having path3 and path2 reach different steady states.

For example, consider path3 in Fig. 3.5 again. The original network has two

dynamic variables X_1 and X_2 , and settling to a steady state can be visualized as a flow from an initial condition to a stable fixed point in its two-dimensional state space (represented by the green arrow that points the rightmost blue dot to the green dot in the X_1 - X_2 plane in the figure). The modularized network has only one dynamic variable X_2 , so its state space is a one-dimensional line; one way to represent the dynamics of the original network in the modularized network is to project the original initial condition to the X_2 line (the blue dot on the line; projections are represented by horizontal arrows), and evolve it on the line. The claim is that the dynamics on the X_2 line may end up in a different steady state. To see why, we need to realize that the dynamics on the X_2 line is equivalent to the dynamics on a *slow manifold* on the X_1 - X_2 plane: by using the module rate law in the X_2 dynamics, we are effectively assuming that X_1 reaches steady states much faster than X_2 and tracks the dynamics of X_2 (it is missing in the dynamics because it is always at steady state hence having no degree of freedom); the slow manifold corresponds exactly to the equation from which we derive the module rate law ($v_1(C_1, X_1) = v_2(X_1, X_2)$; represented by the black curve in the figure). Now suppose there are two stable fixed points on the X_1 - X_2 plane (the green and red dots) and the initial condition projected onto the slow manifold (the blue dot on the curve) falls in the basin of attraction of the other fixed point (the red dot), the one different from the one originally reached. Correspondingly, the dynamics along the X_2 line would end up in a different fixed point as well (the red dot on the line)

There are controversies in metabolism on multistationarity. For example, in the

studies of photosynthesis, some researchers posit the existence and/or biological relevance of multiple stable steady states [85, 38, 90], while some others dispute them [57, 122]. In any case, there are reasons to believe that the basin of attraction of a metabolic steady state should be large: otherwise the system would be easily tossed around between steady states. In this sense, one can simply interpret modularization to be preserving of JC behaviors; the coarse-graining in network topology is lossless in information.

The second property of modularity we want to emphasize lies in its interplay with information topology. As is discussed in Section 2.6, system-component formulation suggests constructing the topology of a system model from the product topology of its component models. For a reaction network model, its components are reactions, and model reduction using manifold boundaries boil down to either using simpler reaction rate laws (that is, component boundaries) or identifying simplifications involving multiple reactions (that is, emergent boundaries). Now with modularity, we add one more twist to such model reduction procedures: we can choose modules as our component models, and translate model reduction to *either simplifying module rate laws or identifying inter-module simplifications*. This new twist is satisfying: often the number of reactions and their interactions are too numerous to keep track of, too complicated to understand and the level of description they provide too microscopic to reveal important information on biological functions; modules, on the other hand, having coarse-grained out many drowning microscopic details, are steps closer towards the network behaviors which directly underlie biological functions. For example, as we will see in the exam-

ple of Calvin cycle (Section 3.4.4), modularity allows us to sharpen our focus on a level of system description that exactly corresponds to common understanding on the functional parts of the system.

There is an important implication of this property. Just like system-component formulation allows us to *reuse* topological information of component models (eg, Sections 2.6 and 3.2), modularity also allows us to reuse topological information of modules. In other words, modular model composition leads to modular model characterization.

We have a good topological characterization of common rate laws. What are the common modules? Having not systematically enumerated modular structures of common metabolic networks,⁸ we do not know. Experience suggests a few candidates (Fig. 3.8).

In all four candidate modules in Fig. 3.8, the dynamic species X can be coarse-grained out to give a net equation involving only the C 's. Pathway module is the simplest among them: any pathway can be considered successive compositions of such modules; reactions R_1 and R_2 form module \tilde{R}_{12} , module \tilde{R}_{12} and reaction R_3 form module \tilde{R}_{123} , etc. Pathway module is also the most fundamental one: the type of modularization it represents can be applied to the linear parts of any networks, shrinking them down to their nonlinear cores, which are exemplified by the other three modules in Fig. 3.8. Energy module involves two cofactors

⁸The flavor of this line of inquiry tastes much like the studies of network motif in systems biology: motif detection through enrichment analysis followed by functional investigations [73].

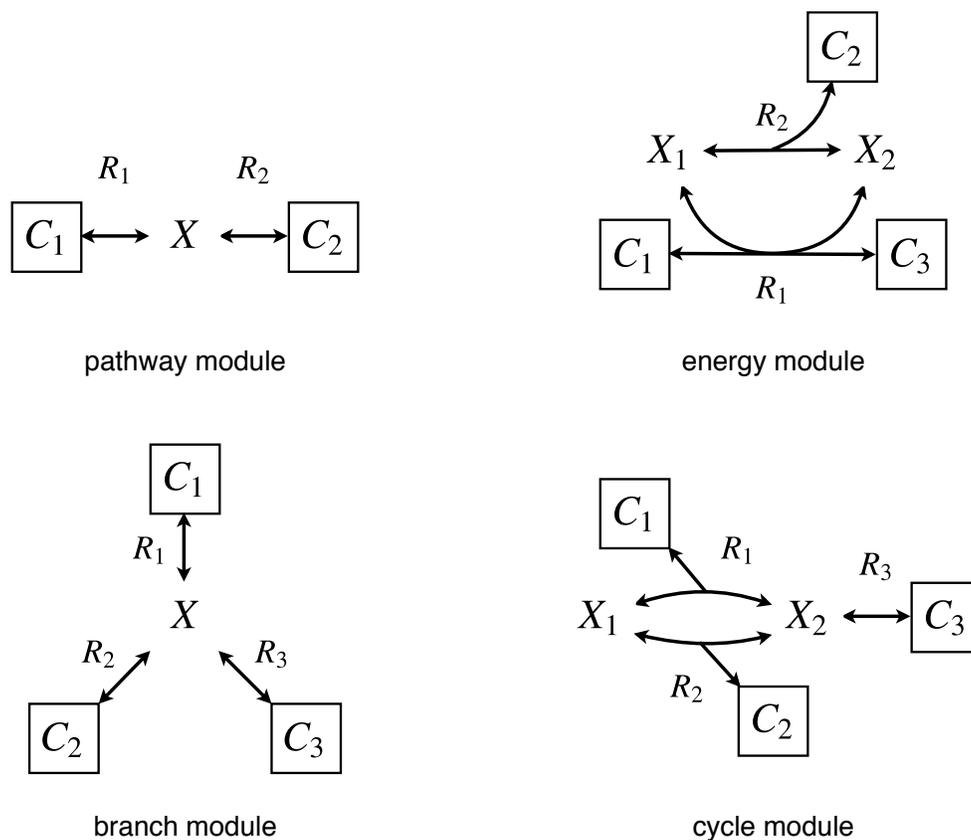


Figure 3.8: A collection of common modules of metabolic networks.

X_1 and X_2 (usually some energy currency molecules such as ATP and ADP) providing/absorbing energy to an otherwise endergonic/exergonic reaction between C_1 and C_3 , and the regeneration of X_1 from X_2 is supported by an external energy source/sink C_2 . Branch module involves metabolic decisions and determines how the flux should be channeled. Lastly, cycle module provides the basic architecture for a metabolic cycle, with an input C_1 , an output C_3 and an energy source C_2 whose interconversions are facilitated by cycle internal species X_1 and X_2 . As we

will see in Section 3.4.4, Calvin cycle can be considered essentially a cycle module, embellished with a few pathway, energy and branch modules through module composition. Another potential perspective on these basic modules is that they represent some sort of “topological canonical forms” of metabolic networks: upon modularization and inessential details removed, any metabolic network takes one of these skeletal architectures (and perhaps a few others).

Because of the fundamental importance and simplicity of pathway module, we attempt an information topological characterization of it in the next section. Such a characterization should provide the same type of utility as rate law characterizations (Section 3.2), as it can be reused in any context where this module appears.

3.4.3 Pathway module

To better understand the rate law of a pathway module, we construct its topology using the algorithm described in Section 2.6. As the simplest reversible rate law in the Michaelis-Menten family, Michaelis-Menten-Haldane rate law is chosen to be our component model (its Hasse diagram is shown in Fig. 3.2).

First, we verify that the system model for JC behaviors constructed using two Michaelis-Menten-Haldane rate laws is free of structural nonidentifiability. Therefore, we can proceed with the second step of constructing the component product topology, which has 81 boundaries since a Michaelis-Menten-Haldane rate law

has nine boundaries (Fig. 3.9). Third, we recognize the symmetry in the model: its model map M has (p, u) -antisymmetry as reversing the order of C_1 and C_2 and the order of two reactions' parameters effectively reverses the whole pathway; note that this symmetry is similar to the one in the example given in Section 2.4.5.⁹ This symmetry in model map is manifested by the bilateral symmetry of the manifold, visible from its Hasse diagram (Fig. 3.9; the stacked nodes in the middle represent singleton equivalence classes). Thanks to this symmetry, we only need to investigate half of the Hasse diagram (the nodes with colored solid boxes in Fig. 3.9); any understanding derived in this half can be translated to the other half.

Fourth, we check the system model at each node for SN, which reveals two classes of SN-plagued models. The first class (colored in green in the figure) have the 0 rate law, which represent a pathway with one reaction inactivated and hence no flux going through: this class of models map the whole parameter space to the origin in the behavioral space J^C , and no SN resolution is possible or needed. The second class (colored in magenta and red) do not have the 0 rate law, so the SN arise through more standard means. One example in this class is the model with two mass-action-Haldane rate laws, which, as we have seen in Sections 2.5.2 and 3.4.1, behaves like a single reaction with a renormalized rate constant (the harmonic mean of the two component rate constants). MBIM resolutions of the models in this class are satisfying: each of them can be identified by a *single com-*

⁹The symmetry in Section 2.4.5 is about a single reaction while here it is the whole pathway. It is the network stoichiometry, which represents the linear pathway, that is compatible with the reaction symmetry and hence preserves it to the pathway scale. Embedding the reaction into a branch module, for example, would lose the symmetry.

ponent boundary, and since they are already connected to their boundaries in the Hasse diagram, the MBIM resolutions translate to simply *removing* them from the diagram.

Lastly, we search in each of the remaining models for emergent boundaries. Upon extensive sampling, only one type of emergent boundaries are identified (colored in blue): a reaction has its rate go to infinity, amounting to a quasi-equilibrium approximation. Yet this is a simple type of emergent boundaries: they are local (involving only one reaction), and correspond to unbounded singular limits of the rate laws (Sections 2.1 and 3.2); if we extend our definitions of boundaries to include them (that is, generalized boundaries), they would be component boundaries and cease being emergent. Since they are included in the first place when constructing the component product topology (Fig. 3.2), such type of emergent boundaries create no new edges. Note that the emergent boundaries in this case would not further lead to more emergent boundaries, as all the remaining parameters reside in one remaining reaction; hence we have no need to further investigate them and can rest assured that all their topologies have been described by the component product topology already.

This concludes our construction of the system topology for the model (`path2mmh`, JC). A brief summary with numbers:

- The component product topology: 81 nodes (all)
- Halving by symmetry: 45 nodes (with colored solid boxes)

- Models with a 0 rate law: 8 nodes (in green)
- Models with nontrivial SN: 3 nodes (in magenta and red)
- Emergent boundaries: 9 nodes (in blue)
- Other component boundaries: 25 nodes (in cyan)

3.4.4 Calvin cycle

Now we apply the machineries we have developed to Calvin cycle, the metabolic network presented in our very first Figure 1.1.

First, we do some basic trimmings of the network, by removing branches leading to photorespiration, starch synthesis, etc., in order to manage the model complexity and highlight the core structure underlying the cycle. Note that such trimmings are boundary approximations: we are effectively driving some rate parameters in those removed reactions to zero. Another way of viewing it is that all Michaelis-Menten rate laws have the 0 rate law as one of their boundaries (Section 3.2). This means that we are still operating under the framework of information topology, traceable to the original full model through a series of boundary approximations. The resulting cycle after the trimming is shown in Fig. 3.10.

Next, we apply modularization to it. Without any modularization, a naive system-component formulation and its resulting algorithm of system topology construction would entail exploring on the order of 10^{15} boundaries (ie, the size

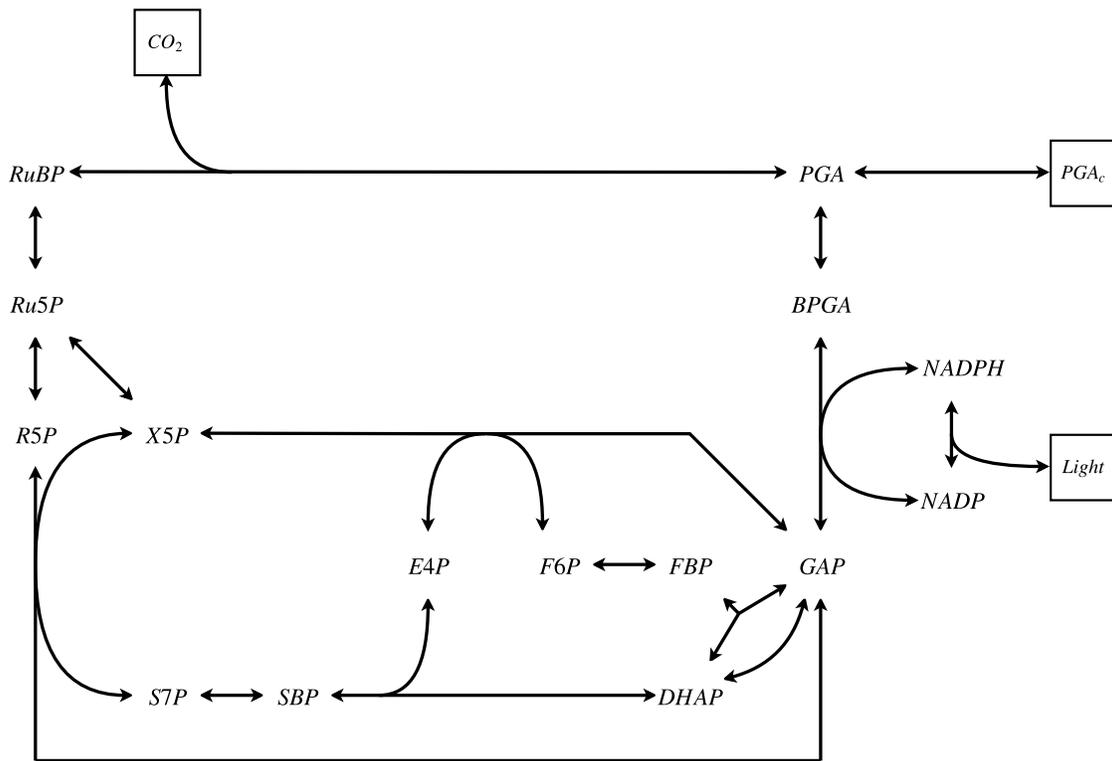


Figure 3.10: The core structure of Calvin cycle.

of the component product topology; Section 2.6). Modularization handles that complexity through divide and conquer, without losing our biological focus, as we will see next.

For modularization, the first thing to note is that, now having three environmental species CO_2 , Light and PGA_c , the whole network can be taken as a module, whose net equation $\text{CO}_2 + \text{Light} \rightarrow \text{PGA}_c$ corresponds to the chemical equation of photosynthesis commonly seen in biochemistry and popular science textbooks (PGA_c is a triose, to be further synthesized to glucose, starch, etc.).

Yet such a coarse description of the network might miss too many details. Let us start from the ground up and stop at some intermediate levels before we reach our final network-module. The network has 15 species and 15 reactions. Within the total 32768 subsets of species, 9989 are connected (the set of α 's). Applying our module-searching algorithm to them yields 516 modules (the set of β 's).¹⁰ The size distribution of the modules thus obtained is plotted in Fig. 3.11.

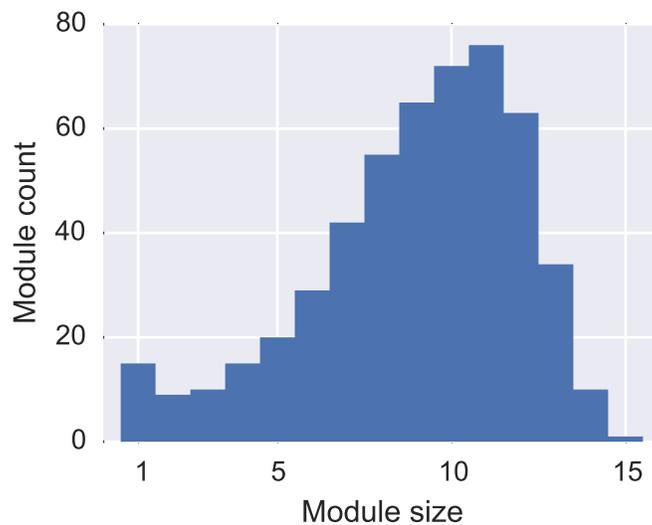


Figure 3.11: The size distribution of all modules in the Calvin cycle in Fig. 3.10.

Most of these modules eliminate some peculiar combinations of species that are hard to interpret, possibly as a result of the high nonlinearity of the network topology (for example, all but GAP, DHAP, E4P and SBP). We can make our for-

¹⁰On average every 20 distinct connected sets of species are mapped to one module, giving a 20-fold degeneracy.

mal notion of modules more relevant by following the guidance of common biochemical understanding [77, 46, 122], which considers the Calvin cycle to be decomposable to four functional parts: fixation of CO₂ by the reaction catalyzed by RuBisCO, regeneration of RuBP which is the substrate of RuBisCO, energization of the cycle by light-dependent reactions, and transport of the sugar molecules synthesized by the cycle out for cellular storage and use. Satisfyingly, we can find within our 516 modules those that correspond exactly to the functional parts in the understanding. The following plots show such a correspondence, and the associated successive coarse-grainings of the cycle.

The first module to be identified is the *regeneration module* (the magenta box in Fig. 3.12), which regenerates pentose RuBP from triose GAP. The module can be decomposed into a collection of pathway modules, branch modules and other types (Fig. 3.8); the different ways of decomposition partly explains the great numbers of modules for the whole cycle (Fig. 3.11). Now with this part of the cycle as a module, we can represent all the reactions therein by a single reaction. This is both satisfying and a bit anticlimactic: most popular descriptions of Calvin cycle and many mathematical models of the cycle in literature (eg, [90]) do not contain the complexities within the regeneration module; rather they mostly replace the complexities by a single regeneration reaction. Our definition and characterization of modules confirm the validity of such a popular simplification and put it on a solid footing, yet not without some serious effort we have put in so far.

Having done the replacement, the network now looks a lot more benign and

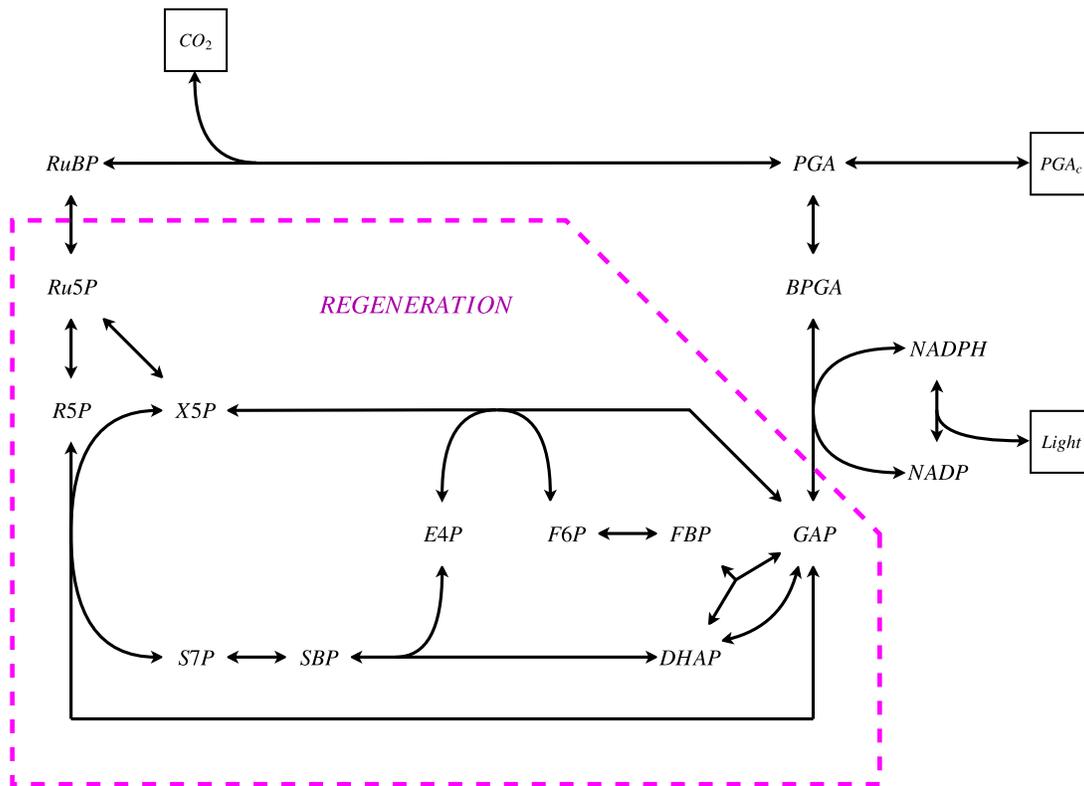


Figure 3.12: The regeneration module of Calvin cycle. The magenta box contains all the species and reactions in the module, effectively regenerating RuBP from GAP.

comprehensible. The next module we immediately recognize in the cycle is our energy module in Fig. 3.8, which in this context we rename the *energization module* because of the unidirectionality of the cycle operation (plants presumably cannot generate light using their sugars). Such a modularization is shown in Fig. 3.13.

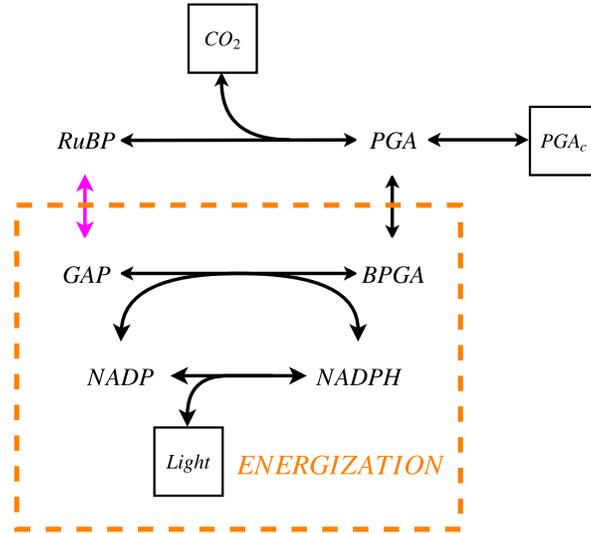


Figure 3.13: The energization module of Calvin cycle. The orange box contains all the species and reactions in the module, effectively harvesting the energy from light to energize the cycle. The magenta reaction represents the coarse-grained regeneration module.

Now having done two modularizations, and if we take the RuBisCO reaction as a *fixation module* in itself and the PGA export reaction as an *export module*, we have four modules in total, and they correspond exactly to common biochemistry wisdom. We can either stop here and start characterizing the module rate laws and hence the whole network model, or go one step further. The step involves lumping the regeneration and energization modules, and we call the resulting module *replenishment module*. This modularization and the resulting three-module cycle are shown in Fig. 3.14

We can go further if needed, by, for example, lumping the fixation and replen-

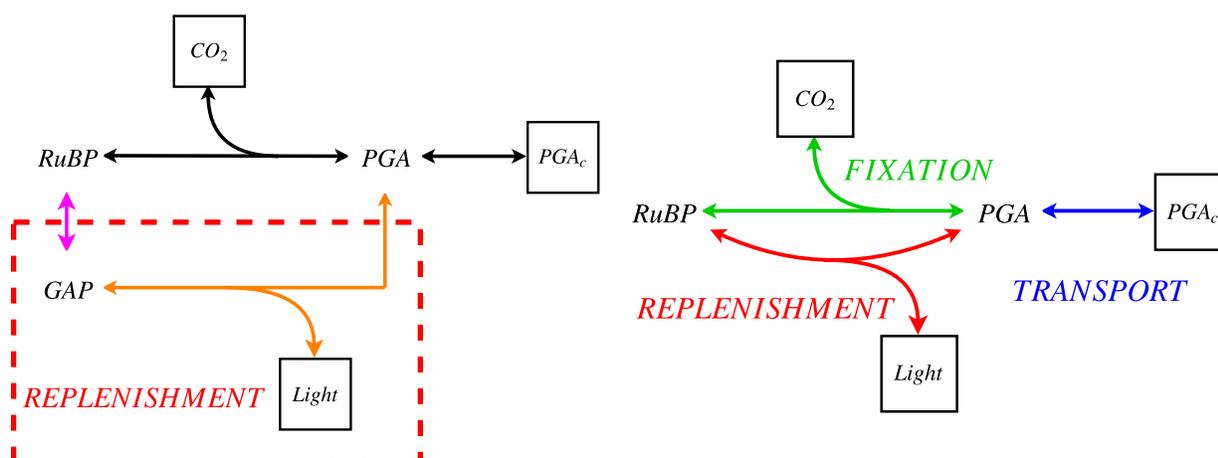


Figure 3.14: (left) The replenishment module of Calvin cycle. The red box contains all the species and reactions in the module, effectively converting PGA all the way to $RuBP$ supported by the energy provided by light. The magenta and orange reactions represent the coarse-grained regeneration and energization modules, respectively. (right) Calvin cycle in fixation, replenishment and transport modules.

ishment modules to a new *production module*, but the resulting network architecture would cease being a cycle, but a linear pathway. To preserve the cycle nature, we stop at the three-module cycle. Our whole modularization procedure can be visualized by a "reaction condensation diagram" (or reversely, "network modularization diagram"), shown in Fig. 3.15.

Let us now focus on the three-module cycle for a moment. Conceptually, it is now simply composed of three modules; mathematically, each module has its own module rate law derived from all its constituent reaction rate laws. On one hand, this means no loss of information; on the other hand, this means that if we

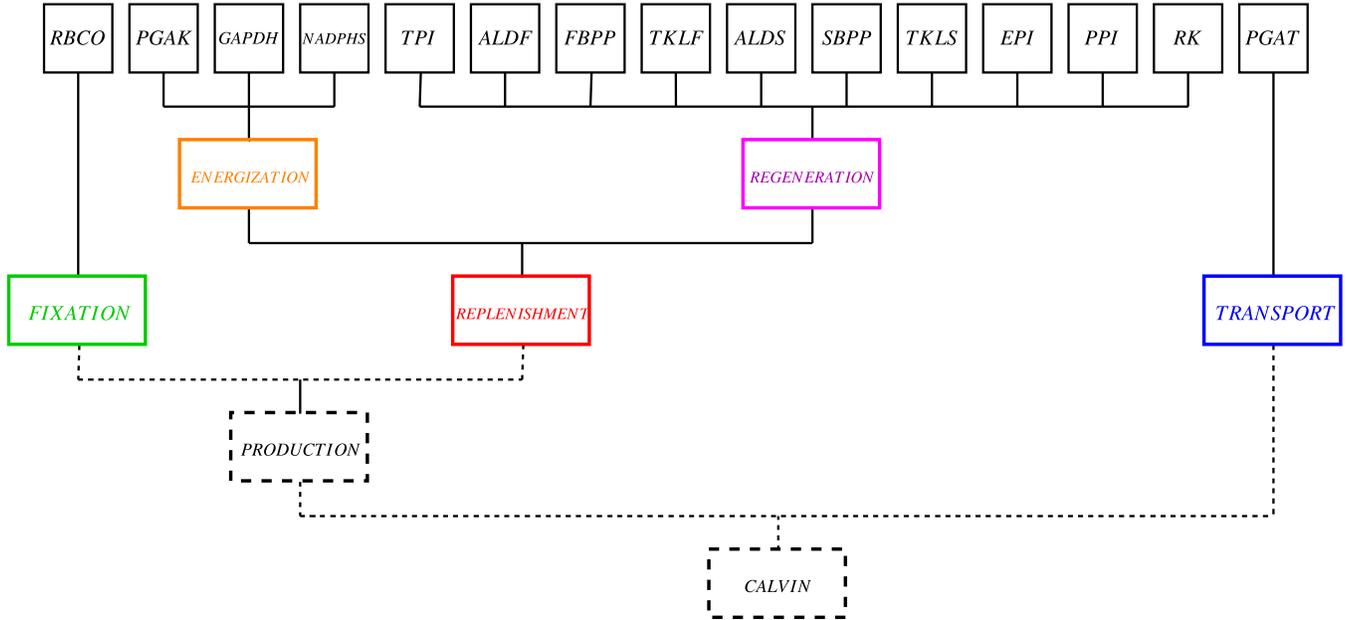


Figure 3.15: Reaction condensation diagram for our modularization of Calvin cycle. The modules are in wide rectangular boxes, and we stop at the colored ones, without further adopting the modularizations represented by the dashed boxes.

simply use the native module rate laws without any coarse-grainings, the model would be mathematically as complicated as the original one (for example, with the same number of parameters). Therefore, upon a chosen modular representation of the network, the next task is to simplify the module rate laws. Since the fixation and transport modules are individual reactions, we can leave them intact for now. The question is how to coarse-grain the replenishment module rate law. We hypothesize that the essential features of a rate law in biochemistry are *saturability*, *monotonicity*, *reversibility* and thermodynamics-compliant, which

are possessed by any reversible Michaelis-Menten-Haldane rate laws. Therefore, before any systematic characterization of the replenishment module rate law has been attempted (like in Section 3.4.3), for now we just replace it with a Michaelis-Menten-Haldane rate law, whose specific form can be chosen to be one that lies on the boundary of the module rate law. One such choice begets a ten-parameter Calvin cycle model.

$$\left\{ \begin{array}{l} v_{fixation} = \frac{\frac{V_1}{K_1 K_2} \left(RuBP CO_2 - \frac{PGA^2}{Q_1} \right)}{1 + \frac{RuBP}{K_1} + \frac{CO_2}{K_2} + \frac{RuBP CO_2}{K_1 K_2} + \frac{2PGA}{K_3} + \frac{PGA^2}{K_3^2}} \\ v_{replenishment} = \frac{\frac{V_2}{K_4} \left(PGA^{\frac{5}{3}} Light^{\frac{5}{3}} - \frac{RuBP}{Q_2} \right)}{1 + \frac{PGA^{\frac{5}{3}} Light^{\frac{5}{3}}}{K_4} + \frac{RuBP}{K_5}} \\ v_{transport} = \frac{\frac{V_3}{K_6} \left(PGA - \frac{PGA_c}{Q_3} \right)}{1 + \frac{PGA}{K_6} + \frac{PGA_c}{K_7}} \end{array} \right. \quad (3.1)$$

The Q 's are equilibrium constants and are assumed known while the V 's and K 's are kinetic parameters.

What can we learn from a model like this about Calvin cycle? So far we are still in the process of exploring it, and we outline below a few thoughts on the possible lines of attack.

The first scenario is when one has data on JC behaviors of Calvin cycle. In this case, one can proceed just like MBAM, but fare better with the global information on the boundary structure provided by information topology: starting from the original model, one descends down the Hasse diagram rank by rank, greedily

choosing the best-fitting model at each rank and preserving the modular structure. What is returned eventually is like what MBAM returns, a minimal model able to describe the data.

If one does not have the data, simulated data can be generated; since we do not know what parameter values to use for generating simulated data, one approach is to sample the parameter space and explore a wide range of possible behaviors. With the sampled behaviors, one can proceed like the first case, ultimately arriving at a reduced model that is able to describe the simulated data for *all* sampled parameter values. In this case, we intentionally choose to be *conservative* and impose a more stringent criterion: the reduced model needs to be able to approximate the full model not only at one parameter point, but at many points. Such a reduced model may not exist, but if it does, it would be a valuable substitute for the full model due to the generality of the approximation validity.

Aside from using goodness-of-fit, a quantitative measure, as the criterion to choose reduced models, one can also focus on qualitative features of models, and descend down the Hasse diagram of the full model checking at each step if the reduced models still or start to possess the features of interest, thereby pinpointing where the features start to appear or disappear and finding the changes in rate laws responsible for the feature changes. Candidate features of this kind include:

1. Is the reduced generalized rate law *reversible*?
2. Is the reduced generalized rate law *thermodynamics-compliant*?

3. Is the reduced generalized rate law *monotonic* in each reaction? Or, in the language of MCA, do all reactions have positive flux controls?
4. Is the reduced generalized rate law *saturable*?
5. Does the reduced generalized rate law *depend* on all reactant concentrations?
6. Is the form of the reduced generalized rate law *similar* to that of the full generalized rate law?

Let us use Feature 5 as an example to explain a bit more the reasoning. In the studies of Calvin cycle, there is one particularly popular model that is used to describe the *JC* behaviors [31, 20], which we call Farquhar-von-Caemmerer-Berry model, or FCB model, after its creators. FCB model has one intriguing feature: it decouples the simultaneous dependence on CO_2 and Light, and models the dependence separately before joining them by a min operation. What is the relationships between FCB model and a kinetic model? Inspecting the reduced models over a Hasse diagram and looking for such a feature as the lack of simultaneous dependence on all reactant concentrations in the reduced models can be a way to find out.

Lastly, one can attempt to establish the algebraic connections between rate laws. In deriving Michaelis-Menten rate laws, one arrives at explicit formulae that express the new emergent parameters in the rate law as functions of the original microscopic parameters (that is, the rate constants). Can we attempt something similar for module rate laws and network rate laws? This understanding may

be very challenging to obtain: our module and network rate laws are derived effectively using QSSA, and as we have seen in Section 1.3, QSSA tends to generate complicated and hard-to-interpret rate laws. This is an area where *algebraic geometry* can potentially play a significant role, since the algebraicity of Michaelis-Menten rate law family implies that any module or network rate law corresponds to an *algebraic variety* in the flux-concentration-parameter space. We are currently attempting to understand the module rate laws for `path2mmh` (Section 3.4.3) in this light.

CHAPTER 4

ESTIMATING RELATIVE CHANGES OF METABOLIC FLUXES USING KINETIC FLUX PROFILING

This chapter contains part of a paper published earlier [49]. There we developed a method to estimate relative flux changes from metabolomic data. Around that time, metabolomics started coming into fashion, and the capacity of measuring many metabolites in one shot with reasonable error bars enjoyed a sudden increase. However, the new technology resembled the old ones in two key characteristics: first, the readout is relative and absolute concentration benchmarking is somehow challenging; second, it was all about concentrations not about fluxes. How to make use of the new technological advance and extract as much information from it as possible, especially about fluxes, became of interest. We had this simple idea, that tweaking an existing method (Kinetic Flux Profiling) a bit would enable us to measure relative flux changes from relative concentration readout, so the method felt like custom made for the metabolomic technology. Both simulation and application to real data about glucose metabolism showed its utility. Note that the estimated flux changes, when combined with some absolute flux benchmarking which is easier than concentration, can be thought as J^C data (the real glycolysis data had C , the external glucose concentration, as the control variable).

After developing and applying the method, we did a bit extra on examining how certain *ad hoc* model reductions would bias the estimates. If we were to do

this project again today, we would probably carry out the analysis on model manifolds to assess the bias and other quantities.

4.1 Introduction

In recent years there has been renewed interest in metabolism resulting from discoveries of its connections to gene regulation [68], epigenetics [52], immunity [84], and pathogenesis of diseases such as cancer [112, 111, 63]. Independently, technological advances in metabolomics promise great improvement of our capabilities in metabolism studies and drug development [83, 88, 11, 78]. However, despite the surge of interest and technological advances, quantitative systems-level characterization of the central trait of metabolism, metabolic flux, has been scarce and challenging. This is in part due to the mathematical nature of flux: rather than the amount of something that is experimentally measurable, it is defined as the rate of change in that amount and has to be inferred through modeling. Several modeling frameworks exist for the purpose. First, the century-old enzyme kinetics [72] and its systems analog, kinetic models of metabolic networks, offer a natural bridge from amount to flux, but unfortunately suffer from the “parameter problem” [40, 45] of depending on many and usually poorly-characterized kinetic parameters. Second, structural models such as Flux Balance Analysis ambitiously aim to predict global distributions of fluxes with minimal data, but the prediction accuracy is still at a stage where validation against more direct estimation results is necessary. Third, isotope-based methods exploit the elegant and

powerful experimental design of isotopes, and are the workhorse for reliable flux estimations.

Among the isotope-based methods, Kinetic Flux Profiling (KFP) [120, 119] has been proven to be powerful [75, 101, 92, 26], with a good balance between experimental ease, model simplicity, and prediction accuracy. In many ways complementary to Metabolic Flux Analysis (MFA) [115], another major isotope-based method which typically uses stationary isotopomer distribution data and is good at estimating relative flux distributions at branch points, KFP uses kinetic isotopomer distribution data and is good at estimating absolute flux scales along linear pathways.

The basic idea of KFP can be illustrated using a toy metabolic network. Consider a system of only one metabolite A connected to the environment by an influx J_1 and an outflux J_2 ; the system is at steady state so $J_1 = J_2 = J$ (Figure 4.1a). KFP works by switching the system from a ^{12}C -labeled environment to a ^{13}C -labeled one at time $t = 0$, measuring the concentrations of ^{13}C -labeled A (termed A^*) at several time points thereafter, and estimating J from the time series data of A^* . After the switching of environment, A^* will gradually infiltrate the pool of A as a result of A^* -carrying influx, with the dynamics described by

$$\frac{dA^*}{dt} = J - J\frac{A^*}{A}, \text{ with the initial condition } A^*(0) = 0. \quad (4.1)$$

The two terms J and $J\frac{A^*}{A}$ in the right-hand side respectively describe the in-

filtration of A^* into the A pool by the influx and the opposing depletion by the outflux, and their net difference describes the rate of change of A^* . The equation is a first-order linear ordinary differential equation (ODE), and can be solved using standard techniques (see Text S1). Its solution is the simple exponential approach function

$$A^*(t) = A(1 - e^{-\frac{t}{A}}), \quad (4.2)$$

and geometrically corresponds to a family of curves parametrized by A and J (Figure 4.1b).

With some measurements of A^* along the curve, parameters A and J can be estimated in a standard way: a least-squares fitting algorithm gives the best fit, and sensitivity analysis or Monte Carlo simulations give the uncertainties. However, it helps to understand *why* KFP should work in this case. First, it is easy to see from Eq. 4.2 that parameter A determines the saturation level of A^* and J/A determines the rate at which the saturation level is approached; in other words, A determines the *scale* and J/A determines the *rate*. To highlight this, we define a rate parameter, $\mu \equiv J/A$; its inverse, $t_c \equiv 1/\mu = A/J$, is conventionally called the characteristic time-scale and numerically corresponds to the time needed to go from the initial condition to $1 - \frac{1}{e}$ (≈ 0.63) of the saturation level. Second, like the familiar Michaelis-Menten hyperbolic curves, the exponential approach curves can also be thought as having three regimes, defined with respect to the characteristic time-scale: the linear regime when $t \ll t_c$, the constant regime when $t \gg t_c$,

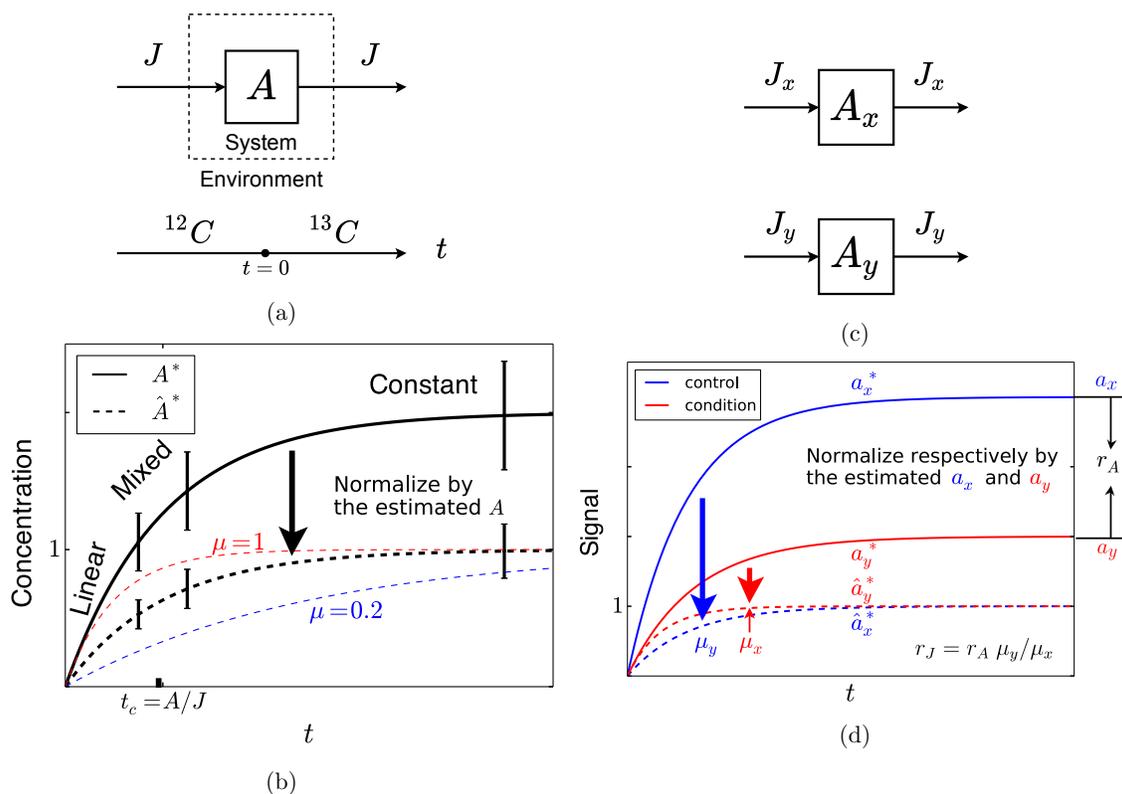


Figure 4.1: Understanding KFP and rKFP. (a) A schematic diagram of KFP applied to a toy metabolic network. At $t = 0$, the system is switched from a ^{12}C -labeled environment to ^{13}C -labeled one, and A^* is measured at a few time points thereafter. (b) For a given trajectory of $A^*(t)$ (the black solid curve), the three time regimes (linear, mixed and constant) are marked and three measurements are made (two in the mixed regime and one in the constant). Normalizing it gives $\hat{A}^*(t)$ between 0 and 1 (the black dashed curve), parametrized by a single parameter μ , which can be estimated by comparing the normalized measurements to \hat{A}^* 's of different μ 's (the red and blue dashed curves). (c) A schematic diagram of rKFP applied to the same network in (a). Relative quantitation is performed on A^* in two conditions (with subscripts x and y respectively) with the goal of estimating $r_J = J_y/J_x$. (d) The ratio in μ between $\hat{a}_x^*(t)$ and $\hat{a}_y^*(t)$ is r_J/r_A (Eq. 4.6), and since μ 's and r_A are identifiable from relative quantitation, so is r_J .

and the mixed regime in between when $t \sim t_c$ (Figure 4.1b). Measurements of A^* in the linear regime are informative of only J (for the slope of that regime is J), the constant regime of only A (for A^* in that regime is constantly A), and the mixed regime of both. In practice, however, usually only the constant and mixed regimes are measured due to their experimental accessibility. Finally, after A is estimated from measurements in the constant regime, the estimation of J from measurements in the mixed regime can be understood in the following way: normalize all measurements by the estimated A to describe the normalized variable $\hat{A}^* \equiv A^*/A = 1 - e^{-\mu t}$; parametrized only by μ and now increasing from 0 to 1, \hat{A}^* changes from a sharply rising curve to a gently rising one as μ decreases; the normalized measurements in the mixed regime nail down the specific \hat{A}^* within the family of curves, together with μ and J (Figure 4.1b). The discussion above can be succinctly summarized as “parameters A and J are *identifiable* when applying KFP to the system in Figure 4.1a”. Later in the paper we will see how the reasoning described here in terms of normalization and rate can be used again to understand the estimation of relative changes of fluxes and the ratios of pool sizes, and the selection of measuring times.

Applying KFP to systems of arbitrary size and network topology and with multiple influxes is less straightforward and requires care [101]. When a reaction involves more than one substrate, the labeling states are no longer just labeled or unlabeled as in KFP, and tracing the origins and fates of ^{13}C labels requires the knowledge of Carbon Transition Map [74] of the reactions. The assumption of irreversibility can eliminate this complication for decomposition reactions, but

is only valid for far-from-equilibrium ones. Also, multiple influxes to a system complicate modeling the dynamics of ^{13}C -labeled metabolites in the same way as multi-substrate reactions do. For this reason, KFP works best for systems consisting of mono-molecular reactions, and works for general systems only through gathering additional information, making assumptions, or using only part of the data that is more amenable to model. The last strategy corresponds to the idea used in an extension of KFP called extended KFP (or eKFP) [101], and is relevant in our later discussion on the capacity of KFP and our extension of it in studying metabolic cycles.

Two considerations motivate us to extend KFP beyond its current scope. First, KFP requires *absolute quantitation* of metabolites, meaning that their absolute concentrations have to be measured, while many experimental techniques such as mass spectrometry can only perform *relative quantitation* readily, meaning that the measurement output is scaled from the absolute concentration by a metabolite-specific unknown constant; going from relative quantitation to absolute quantitation typically requires performing relative quantitation on some reference samples whose absolute concentrations are known, which can often be a challenge due to the increased effort of both additional experiments and procurement of reference samples. Second, often it is the relative changes of fluxes (or biological quantities in general) between two conditions that are of interest or biological relevance (eg, wildtype *vs.* mutant, control *vs.* drug-treated [75, 37]), and estimating the absolute fluxes of the two conditions only to get their ratios is inefficient (the information regarding their scales is eventually discarded) and roundabout (three

rounds of estimation are carried out, one for each condition and one for their ratio).

In this paper, we report an extension of KFP that can estimate the relative changes of fluxes using only relative quantitation, which we call rKFP, hence addressing the two considerations above. To improve the reliability and strength of KFP and rKFP, we examine some issues in the application of the methods, on both setting up models and selecting measuring times. Finally, we apply rKFP to experimental data collected in normal and glucose-deprived conditions, estimating the relative flux changes in glycolysis and its branching pathways and arriving at new biological insight.

4.2 Results/Discussions

4.2.1 Extending KFP to estimate relative flux changes

Consider again the toy system in Figure 4.1a, now in two different conditions; the same experimental procedures of switching environment at $t = 0$ and subsequent measurements of A^* apply, but only with relative quantitation (Figures 4.1c and 4.1d). The aim is to estimate the relative change of J between the two conditions.

We start by writing down the relationship between the relative quantitation measurements (which we call signals) and the absolute quantitation measure-

ments (concentrations) for the two conditions:

$$a_i^* = p_A A_i^*, \quad i = x, y, \quad (4.3)$$

where an upper-case letter denotes concentration and lower-case signal, p their ratio, superscript $*$ a labeled quantity, and subscripts x and y quantities for the two conditions respectively (a list of notation can be found in Table 4.1). Since now we can only perform relative quantitation and hence a_i^* becomes the measurable, we establish its dynamics by plugging in the dynamics of A_i^* which we have solved in studying KFP:

$$a_i^*(t) = p_A A_i (1 - e^{-J_i t / A_i}), \quad i = x, y. \quad (4.4)$$

The above two equations highlight a simple but important fact: $a_i^*(t)$ is simply $A_i^*(t)$ scaled by an unknown constant p_A , and they share the same intrinsic dynamics. Therefore, the reasoning of normalization and rate described in the discussions of KFP in the introduction appears even *more* natural in this situation: although relative quantitation leaves us oblivious to the scale, we can still normalize the measurements to uncover the rate.

$$\hat{a}_i^*(t) \equiv \frac{a_i^*}{a_i} = 1 - e^{-J_i t / A_i} = 1 - e^{-\mu_i t}, \quad i = x, y. \quad (4.5)$$

Defining the relative changes of pool size $r_A \equiv A_y / A_x$ and flux $r_J \equiv J_y / J_x$, the

relative change of μ can then be expressed in terms of r_A and r_J :

$$\mu_y = \frac{J_y}{A_y} = \frac{r_J J_x}{r_A A_x} = \frac{r_J}{r_A} \mu_x. \quad (4.6)$$

Since μ_x and μ_y are identifiable using normalized measurements in the same way as μ described in the introduction, and r_A is obviously identifiable (Figure 4.1d), so is r_J . This concludes our explanation of why rKFP should work for the toy system in Figure 4.1c.

To summarize the mechanics of rKFP for the system, one sets up the following two equations:

$$\begin{cases} \frac{d a_x^*}{dt} = p_A(J_x - J_x \frac{a_x^*}{a_x}) = p_A(J_x - J_x \frac{a_x^*}{A_x p_A}), \\ \frac{d a_y^*}{dt} = p_A(J_y - J_y \frac{a_y^*}{a_y}) = p_A(J_x r_J - J_x r_J \frac{a_y^*}{A_x r_A p_A}), \end{cases} \quad (4.7)$$

which form a model that is parametrized by $\theta = (A_x, J_x, p_A, r_A, r_J)$, and predicts $a_x^*(t)$ and $a_y^*(t)$ (the solutions are Eq. 4.4); measurements of $a_x^*(t)$ and $a_y^*(t)$ allow for estimating r_J (and r_A) with precision (identifiable). Two remarks follow: first, while parameters A_x , J_x and p_A are obviously non-identifiable, two of their functions, $a_x = p_A A_x$ and $\mu_x = J_x/A_x$, are, amounting to two constraints on (A_x, J_x, p_A) ; second, in light of the constraints, the model can be parametrized in other natural ways: for example, one can replace A_x with a_x , or J_x with μ_x , and the resulting new parametrization would be as interpretable.

Table 4.1: Definitions of variables and their identifiabilities in rKFP

Symbol	Meaning	Identifiable in rKFP
A (uppercase)	metabolite A or its pool size (absolute quantitation) ¹	No
a (lowercase)	signal of the pool size (relative quantitation)	Yes
A^*	concentration of ^{13}C -labeled A (absolute quantitation)	No
a^*	signal of ^{13}C -labeled A (relative quantitation)	Yes
\hat{A}^*	the fraction of A^* in A , $\hat{A}^* \equiv A^*/A$	Yes
\hat{a}^*	the fraction of a^* in a , $\hat{a}^* \equiv a^*/a$	Yes
J	flux	No
μ	rate, $\mu \equiv J/A$	Yes
p_A	proportionality constant between signal (relative quantitation) and concentration (absolute quantitation) of A , $p_A \equiv a/A = a^*/A^*$	No
\bullet_x (subscript x)	a quantity of control	–
\bullet_y (subscript y)	a quantity of condition	–
r_A	relative change of pool size, $r_A \equiv A_y/A_x$	Yes
r_J	relative change of flux, $r_J \equiv J_y/J_x$	Yes
ρ_{A_i,A_j}	ratio of pool size, $\rho_{A_i,A_j} \equiv A_j/A_i$	Yes

For larger networks consisting of single-substrate reactions arranged in linear pathways and branch points, rKFP proceeds in a similar way: equations like Eq. 4.7 are constructed for each metabolite in the network, and relative quantitation measurements of the metabolites allow for estimating the relative changes of all the independent fluxes with precision. However, for metabolic networks involving reactions of multiple substrates, especially cycles (multi-substrate reactions are usually present in cycles as part of the network design), rKFP in its above form cannot handle the situation. Fortunately, a variant of KFP, termed extended KFP (or eKFP), has been developed to overcome the problem [101], and

its rKFP version, which we term reKFP, can estimate the relative flux changes for cycles without having to deal with the complications of modeling carbon transitions that arise in multi-substrate reactions [4]. The essential idea of eKFP is that while there can be multiple labeled states for the reactants in multi-substrate reactions and keeping track of their transitions can be complicated, there is always only one unlabeled state and modeling its dynamics is relatively simple and stays within the KFP framework; the downsides are that only a fraction of the information in the data is used and that the measurements of the unlabeled metabolites have to be accurate which sometimes can be nontrivial to achieve due to media contamination.

Before we conclude the discussion of rKFP, we mention one more nontrivial quantity identifiable from relative quantitation, the ratio of pool sizes. As an illustrating example, consider a metabolic pathway of two metabolites with relative quantitation; we again normalize the measurements to uncover the intrinsic dynamics. The idea is that the further the second metabolite lags behind the first one, the more abundant it is compared to the first one.

Formally, one can plug $\rho_{A_1, A_2} \equiv \rho \equiv A_2/A_1$ and $\mu_1 \equiv J/A_1$ into the normalized $A_2^*(t)$ (Text S1 contains a derivation of $A_2^*(t)$):

$$\begin{aligned}\hat{A}_2^*(t) &= 1 - \left(\frac{A_1}{A_1 - A_2} e^{-\frac{Jt}{A_1}} + \frac{-A_2}{A_1 - A_2} e^{-\frac{Jt}{A_2}} \right) \\ &= 1 - \left(\frac{1}{1 - \rho} e^{-\mu_1 t} + \frac{-\rho}{1 - \rho} e^{-\frac{\mu_1 t}{\rho}} \right).\end{aligned}$$

Since μ_1 is identifiable from relative quantitation of A_1^* , the single parameter that is left, ρ , determines how much \hat{A}_2^* lags behind \hat{A}_1^* and is identifiable from the comparison. In the introduction we discussed the general challenge of absolute quantitation, but here we show that if absolute quantitation can be performed on, or good prior knowledge exists for, *some* metabolites (eg, the cellular glucose concentration is known to be about 5mM [113]), this information of scale can be propagated across the network to *other* metabolites through the estimates of pool size ratios, estimating absolute concentrations from relative quantitation.

4.2.2 Missing data: effects and pitfalls

Despite the great advances in metabolomic technologies, it is nevertheless common to have missing data. It is a result of the chemical properties of metabolites: some are too unstable to be accurately measured, and some isomers are too similar to each other to be distinguishable. To set up a computational model for KFP or rKFP under missing data, one in principle has the following options: (1) use a reduced model where the network components corresponding to the missing data are removed; (2) use a full model where all components are kept and the part of the model corresponding to missing data represents additional degrees of freedom unconstrained by data; (3) use the full model but incorporate prior information for the part of model uncovered by data; (4) spend additional effort to collect all data and use the full model. We observe that a common practice in applying KFP is to choose the first option and use reduced models when there is missing

data (eg, [75]). We acknowledge that this option is tempting and reduced models do offer many advantages such as conceptual simplicity and computational manageability (after all, “all models are wrong; some are useful”); however, because of the potential bias reduced models might introduce to parameter estimation and model prediction, we believe that their use would be better justified after a careful consideration of their effects.

Consider the following example. Figure 4.2a provides a cartoon of a typical situation in applying KFP: in a linear pathway of three metabolites, A_2 is hard to measure and therefore constitutes missing data. The above four options concretize to the followings: (1) use the reduced model consisting of only A_1 and A_3 ; (2) use the full model consisting of all three metabolites with A_2 uncovered by data; (3) use the full model but put a prior distribution (in the Bayesian sense) on the A_2 pool size based on previous knowledge; (4) try to collect A_2 to complete the data and use the full model. We note that the desirability of option (3) depends on what prior distribution is available and how close it is to the true value, *i.e.*, how well one *a priori* knows the missing piece. Hence it has to be judged on a case-by-case basis and cannot be discussed generally. We therefore exclude option (3) in our subsequent analysis, and only note that in the limit of a correct tight prior the case converges to option (4) of completing the data, and in the limit of a loose prior the case converges to option (2) of effectively having no prior information.

We call this scenario of missing data in Figure 4.2a *missing metabolite*, and the corresponding procedure of constructing reduced models *metabolite removal*. We

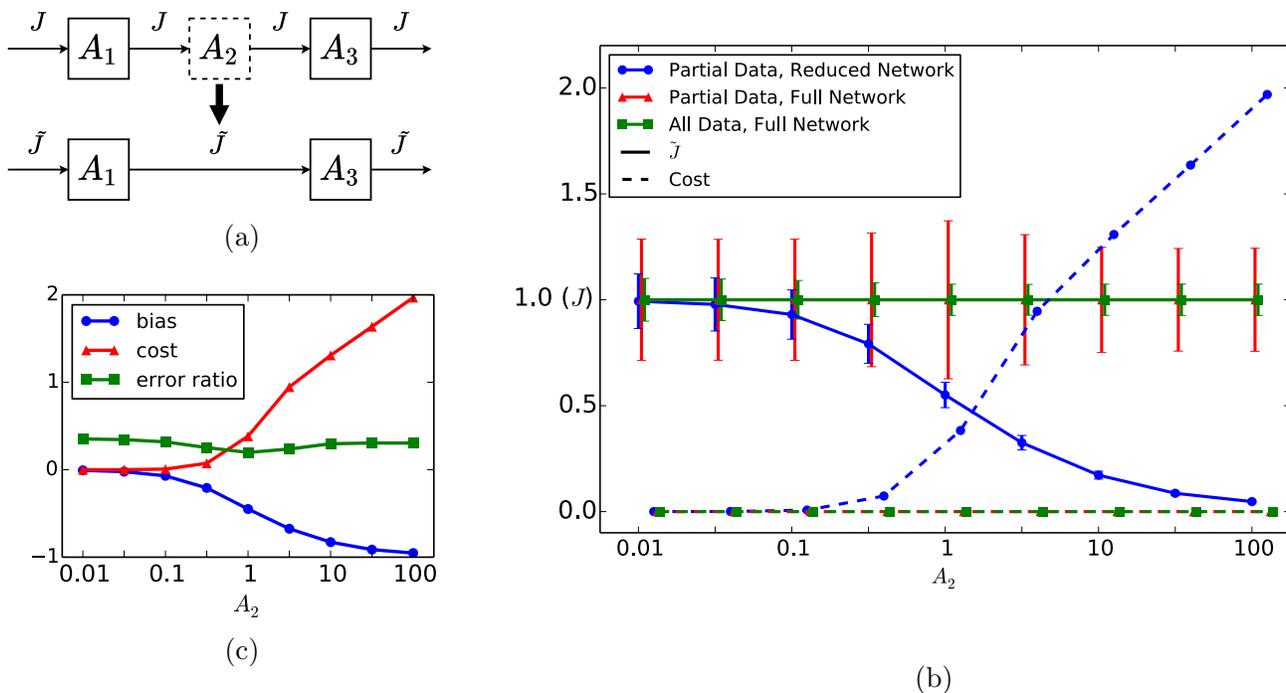


Figure 4.2: Metabolite removal in KFP. (a) A schematic diagram of getting the reduced model through metabolite removal in KFP. Dashed squares represent metabolites removed in the reduced model; thick dark arrow represents reduction; \tilde{J} represents the estimated J (potentially biased). (b) The estimation results for the three options. The solid curves represent \tilde{J} , the dashed curves represent the cost of fitting (normalized by the number of data points to be comparable across options), and three colors represent the three options. Parameter values used for generating the simulated data: $A_1 = A_3 = J = 1$ (overall patterns independent of the choice here). (c) The estimation results in (b) in terms of the three summary statistics.

identify and name three additional common scenarios of missing data with their associated reduction procedures: first, *missing pathway*, where a branching pathway has poor data coverage with most metabolites unmeasured, associated with *pathway removal*, where the branching pathway is removed from the model; second, *undistinguished metabolites*, where the individual identity of a set of metabolites in a measurement cannot be resolved, associated with *lumping*, where the undistinguished metabolites are lumped into a single pool; third, *unknown reversibility*, where the extent of reversibility of a reaction is unknown, associated with *assuming irreversibility*, where potentially reversible reactions are modeled as irreversible for simplicity. In the remainder of this section, we describe in details the effects of metabolite removal on both KFP and rKFP, for this case is the simplest and hence most illustrative, and also for this case turns out to be important (see below); after that we briefly describe the results on pathway removal and lumping; results on assuming irreversibility are discussed in the next section.

Back to Figure 4.2a, intuitively, using the reduced model with A_2 removed would *underestimate* J , for two reasons. First, since the influx J is ^{13}C -saturated and constant with time, after time t one would expect Jt amount of ^{13}C in the system, distributed across different metabolite pools; removing A_2 from the network excludes the ^{13}C in that pool, causing an underestimation of the ^{13}C in the system and hence an underestimated J . Second, the presence of any metabolite pool slows down the infiltration process of ^{13}C along the network, and if the slow-down of the infiltration by the A_2 pool is concealed by removing A_2 from the network, the slowed infiltration would be attributed to a lower J . Both factors become more

pronounced as the pool size of A_2 increases, and so should be the underestimation.

Figure 4.2b shows the results for the three options. First, using the reduced model indeed underestimates J , and it worsens as A_2 increases (the blue solid curve), confirming our intuition. Second, using the reduced model decreases the goodness-of-fit between the model and the data, quantified by the cost of fitting, and it increases with A_2 as well (the blue dashed curve). Third, using full models causes no bias or cost (red/green and solid/dashed curves). Fourth, the extra hard work of collecting the data of A_2 pays off by shrinking the uncertainties of the estimated J (the red *vs.* the green error bars). These observations suggest the following three summary statistics:

- bias, defined as $(\tilde{\theta} - \theta)/\theta$, where $\tilde{\theta}$ and θ are respectively the estimated and true value of a parameter (J in this case).
- cost, the cost of fitting using the reduced model.
- error ratio, defined as σ_c/σ_p where σ is the standard error of the parameter estimate and subscripts c and p refer respectively to the cases of complete and partial data.

These statistics summarize the effects and pitfalls of missing data: error ratio quantifies their adverse effects on parameter estimation (or, looking optimistically, the reward of completing them), and bias and cost quantify the harm of using reduced models. Figure 4.2c re-plots the results in Figure 4.2b in terms of the three summary statistics. From these plots we can see that KFP is rather sensitive to

metabolite removal: in the case of the toy model, a missing metabolite of pool size equal to others causes roughly a 50% bias. The same should hold for general pathways: if the total pool size of the missing metabolites in a pathway is relatively large, the bias should be considerable. For this reason, we give the general suggestion that, unless the missing metabolites are known *a priori* to have small pool sizes, one should use full models in KFP.

We next examine how metabolite removal affects rKFP (Figure 4.3a). We have concluded that removing A_2 in KFP underestimates J and the underestimation worsens as A_2 increases. Since r_J is the ratio between J_y and J_x , we expect that the bias in r_J depends on A_2 in the two conditions, A_{2x} and A_{2y} : if A_{2x} is large and A_{2y} small, J_x is implicitly more underestimated than J_y (implicitly as rKFP does not explicitly estimate J), giving an overestimated r_J , and in the same way a small A_{2x} and a large A_{2y} give an underestimated r_J . Like KFP, we expect the bias to increase with the pool size difference of A_2 between two conditions.

both Figure 4.3b plots the bias of r_J , and confirms our intuition. However, an important feature distinguishes the case in rKFP from KFP. Figure 4.3b shows a red line where the relative change of A_2 is the same as A_1 and A_3 , and above the red line r_J is underestimated and below overestimated; in other words, when the relative change of A_2 pool size exactly matches the others in the pathway, r_J is estimated without bias. This hints at another important advantage of rKFP over KFP: bias can be introduced to the two conditions in such a similar way that some of it is canceled out. The question remains of how likely the pool size of a metabo-

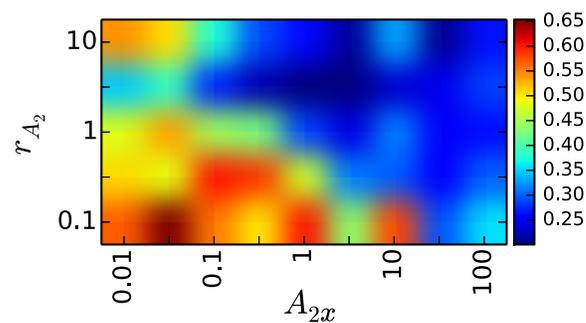
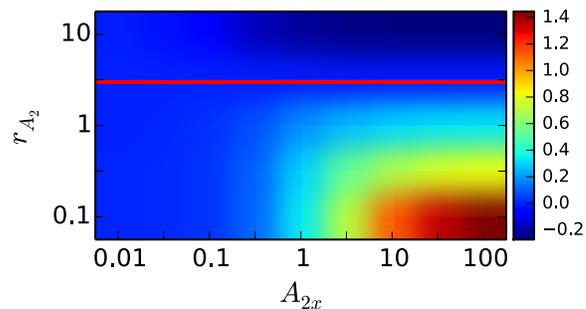
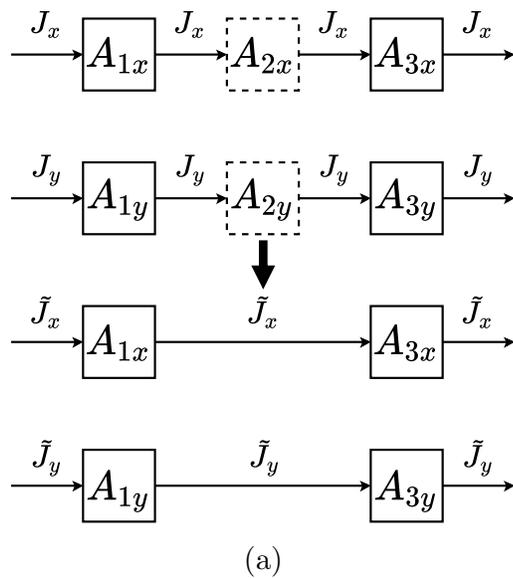


Figure 4.3: Metabolite removal in rKFP. (a) A schematic diagram of getting the reduced model through metabolite removal in rKFP. The same figure scheme as in Figure 4.2a applies here. Parameter values used in generating the simulated data: $A_{1x} = A_{3x} = J_x = 1$, $A_{1y} = A_{3y} = 3$, and $J_y = 2$. (b) Dependence of bias on the pool size of the missing metabolite in two conditions. (c) Dependence of error ratio on the pool size of the missing metabolite in two conditions.

lite changes in a similar way to all others in a pathway. From an enzyme kinetic perspective, this says that the reactions along the pathway are in similar *elasticity* regimes [27], which is also consistent with the consensus that reactions in a pathway in general share the *flux control* [34]. We hence believe that in rKFP some of the bias is indeed canceled out which makes it *more robust* to metabolite removal than KFP. However, we also note in Figure 4.3b that the bias worsens quickly as the relative change in pool size of different metabolites diverge: when it is off by a factor of three, say, $r_{A_2} = 1$ (while $r_{A_1} = r_{A_3} = 3$), the bias becomes about 40%. This implies that in a typical situation despite the canceling the remaining bias is still too large to justify using the reduced model, and hence we again suggest using full models in rKFP for missing metabolites. The ideas of canceling and metabolites along a pathway changing pool sizes similarly, however, find their use in the next section where they serve to justify using reduced models for unknown reversibility.

To summarize, we have demonstrated above the effects of missing metabolites on (r)KFP and the pitfalls of removing them; we suggest that unless one has good prior knowledge about the missing pool size or its relative change between two conditions that ensures a small bias, full models should be used. The approach we use for the demonstration is an intuitive and computational one. Also included there are some detailed studies of how two other reduction scenarios, pathway removal and lumping, might affect KFP and rKFP. We summarize the results in the following: first, both KFP and rKFP are rather robust to the removal of a minor branching pathway (that is, a branching pathway that does not carry the majority

of the flux), while using the full model greatly inflates the confidence interval of the parameter estimates due to the additional degrees of freedom; second, since lumping is usually applied to isomers which are often close to chemical equilibrium, it is generally rather innocuous in light of the results in the next section. In a later section on data analysis, we see that the results in this section are verified by comparing the estimation results of different models.

4.2.3 Modeling reversible reactions

Biochemical reactions often operate close to equilibrium [12] (for example, it is conventionally thought that this applies to seven out of the ten reactions in glycolysis [36]), and a reaction's distance to equilibrium is commonly quantified by the difference in Gibbs free energy, ΔG , between substrates and products: the closer ΔG is to zero, the closer the reaction is to equilibrium. One of the properties of a close-to-equilibrium reaction is that both of its forward and backward fluxes are large and most of them are canceled out to give a much smaller net flux; quantitatively this is described by the *flux-force relationship*: $J^+ / J^- = e^{-\frac{\Delta G}{RT}}$, where J^+ and J^- are the forward and backward flux respectively, R the gas constant and T the temperature [10]. Quantity $\frac{J^+ + J^-}{J^+ - J^-}$, the total flux over the net flux, therefore describes how much "futile" work the reaction does compared to "useful" work, and predicts from the flux-force relationship that as the reaction approaches equilibrium, exponentially more fluxes simply go back and forth compared to the net flux. This has an important implication in the context of (r)KFP: when the re-

versible fluxes are large compared to the net flux, the time-scale of the mixing of substrate and product pools is much shorter than that of the infiltration of the ^{13}C -labeled metabolites in each pool, making the two pools effectively one as far as (r)KFP is concerned.

This suggests another potential issue in setting up models for KFP or rKFP: ignoring the reversible fluxes when the reaction is close to equilibrium, as is commonly done, might introduce bias. To check this, we again use a toy system (Figure 4.4a) and, similar to the case of missing metabolite, conceive three options for modeling a reversible reaction: (a) model it as irreversible; (b) model it as reversible and let ΔG be a free parameter; (c) model it as reversible and measure ΔG to some precision (which typically requires absolute quantitation). Figure 4.4b plots the summary statistics for KFP, which shows that there is a small bias and a moderate cost when a highly reversible reaction is assumed irreversible. It suggests that KFP might be robust to assuming irreversibility; however, even the small bias can be avoided since in KFP ΔG can be calculated and is not missing information: KFP uses concentration data, which can be used to calculate ΔG through its definition (see below); therefore one can explicitly incorporate in the model J^+ and J^- which depend on the parameter J through the relationships $J^+ = \frac{J}{e^{-\frac{\Delta G}{RT}} - 1}$ and $J^- = \frac{J e^{-\frac{\Delta G}{RT}}}{e^{-\frac{\Delta G}{RT}} - 1}$. Also plotted is the estimated pool size ratio between A_1 and A_2 with one being the true value (the purple dashed line), which shows a vast underestimation when the reaction is close to equilibrium; this can be explained by the following: as the reaction becomes more reversible, A_1^* and A_2^* share the dynamics to a greater extent, and A_2^* closely following behind A_1^* is

interpreted by the algorithm as resulting from $A_2 \ll A_1$.

Figure 4.4c illustrates how ignoring reaction reversibilities biases rKFP, which can be understood in a similar way as Figure 4.3b in using reduced models. First, the direction of the bias depends on ΔG 's of the two conditions; since ignoring reaction reversibilities underestimates J in KFP (Figure 4.4b), if the reaction is more reversible in control (x) than in condition (y), then J would be more underestimated in control, giving an overestimated r_J (the red region), and *vice versa*. Second, if $\Delta G_x = \Delta G_y$, then r_J is estimated without bias (the red line).

We naturally wonder how likely the change of ΔG falls around the red line, and find that its definition offers the clue. Since $\Delta G \equiv \Delta G^\circ + RT \ln Q$, where ΔG° is the standard Gibbs free energy change and Q the reaction quotient defined as the product concentrations divided by the substrate concentrations. For a monomolecular reaction with S the substrate and P the product, ΔG_x and ΔG_y can be related in the following way:

$$\Delta G_y - \Delta G_x = RT(\ln Q_y - \ln Q_x) = RT\left(\ln \frac{P_y}{S_y} - \ln \frac{P_x}{S_x}\right) = RT \ln \frac{P_y S_x}{P_x S_y} = RT \ln \frac{r_P}{r_S}. \quad (4.8)$$

In other words, the change in ΔG depends on the relative changes in pool size of P and S . In the previous section we conclude that the relative changes in pool size of the metabolites along a pathway are likely to be similar, and Figure 4.4c shows that, *unlike* the case of metabolite removal, the bias increases *slowly* as the relative pool size changes of different metabolites diverge (the red dashed curves

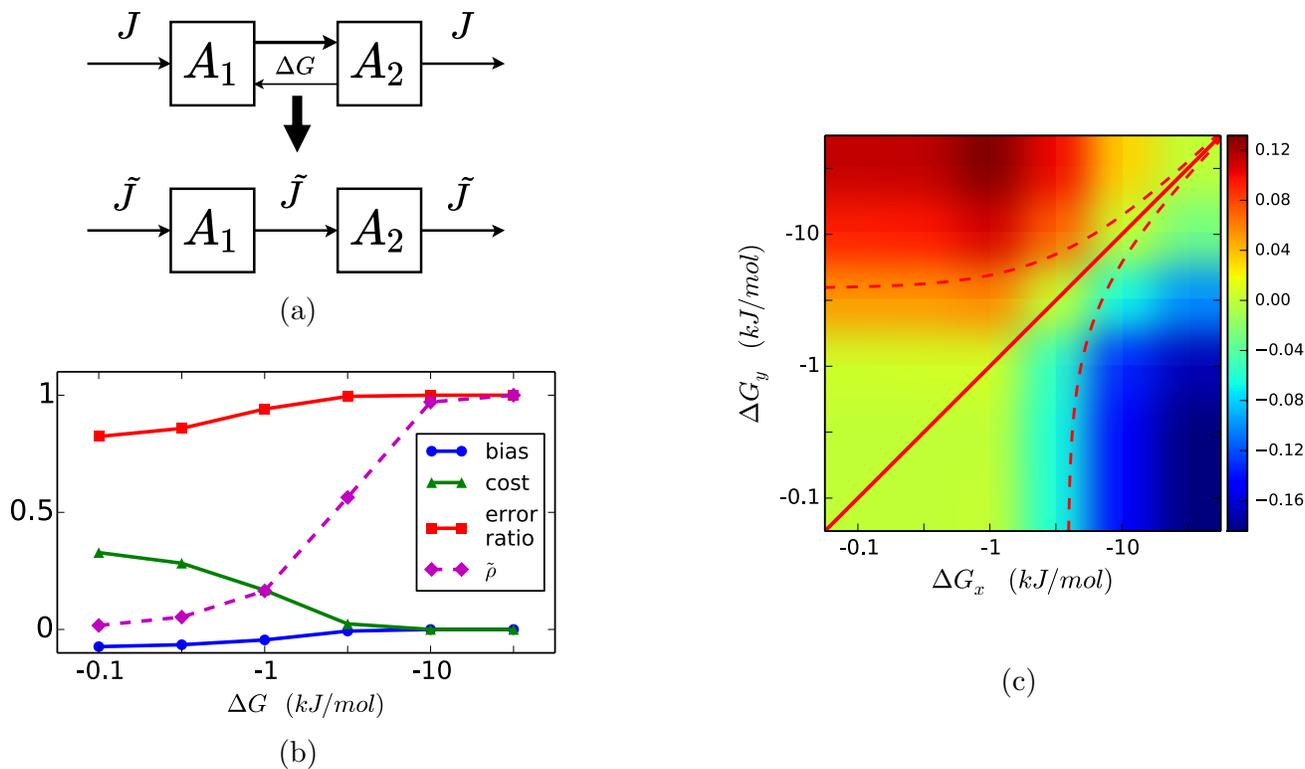


Figure 4.4: Modeling reversible reactions in KFP and rKFP. (a) A schematic diagram of the toy system considered here; for rKFP two copies of each network are similarly made as in Figure 4.3a. Parameter values used in generating the simulated data: $A_1 = A_2 = A_{1x} = A_{2x} = J = J_x = 1$, $A_{1y} = A_{2y} = 3$, and $J_y = 2$. (b) Dependence of the summary statistics on ΔG in KFP. (c) Dependence of the bias of r_J on ΔG of the two conditions in rKFP. The red solid diagonal line corresponds to $\Delta G_x = \Delta G_y$ where there is no bias. The red dashed curves correspond to a five-fold difference in the relative pool size changes between the substrate and product, a range we observe in our data.

correspond to a five-fold difference in the relative pool size changes, a range of variation we observe in our glucose-deprivation data, and delineate a region surrounding the diagonal line of very small bias). We therefore conclude rKFP should be robust to assuming irreversibility and reduced models can be used.

We last note that using the full model in this case and incorporating ΔG 's as free parameters would leave the model underconstrained by data and lead to huge uncertainties in parameter estimates, since one additional parameter would be accumulated for each reaction modeled as reversible. Incorporating prior information on ΔG 's would help, but at a systems level the information is scarce as it requires accurate absolute quantitation [12] and predictive computational methods are still being developed [80]. It is also our experience that rKFP models constructed this way with additional ΔG parameters are prone to numerical problems in data fitting and parameter sampling, since for each reaction modeled as reversible two copies of ΔG (ΔG_x and ΔG_y) need to be made and they are entangled with parameters of relative pool size changes through Eq. 4.8. In light of all these challenges, the robustness of rKFP to assuming irreversibility is advantageous.

4.2.4 Selecting measuring times

In addition to the issues on setting up models as discussed in the previous two sections, we have also investigated the experimental design issue of selecting measuring times for a metabolic pathway in (r)KFP. We list below the main conclu-

sions.

First, we note that the problem can be formulated as a classical experimental design one [7]: select a set of measuring times such that the errors of the estimated parameters (J 's in KFP and r_j 's in rKFP) are the smallest. However, such an approach suffers from computational intractability, physical uninterpretability and unrealisticness (Text S1).

We choose to adopt an alternative approach for which there is a physical intuition: since the dynamics of the metabolites in a pathway are linear combinations of exponential functions (Text S1), and in the introduction we show that for a single exponential function, there is a well-defined characteristic time-scale, we wonder if we can make use of this special mathematical structure and target the measuring times on the different time-scales.

We show that for the first metabolite in a pathway, if a late time has been measured so that its (relative) scale is estimated, the optimal measuring time would be around $t_c = A_1/J$, the characteristic time-scale. This highlights another significance of t_c : if the curve of $A^*(t)$ in Figure 4.1b is held fixed at two ends and the middle left wiggling as μ changes, placing a pin at the characteristic time-scale leaves the least wiggle room.

We also show by using a similar reasoning that for the second metabolite in a pathway the optimal measuring time is around $(A_1 + A_2)/J$, and similarly for the k -th metabolite around $\sum_{i=1}^k A_i/J$. That is, the measuring time for the k -th

metabolite should be around the sum of the characteristic time-scales of all the metabolites up to the k -th one; this makes intuitive sense, given that the dynamics of A_k^* is a mixture of k time-scales due to the actions of the k pools. We therefore provide the following practical suggestion: for a metabolic pathway of n metabolites with a rough prior guess of the flux (and pool sizes for rKFP), after measuring at a late time so that the (relative) scales of all metabolites can be estimated, other measurements should go to $\sum_{i=1}^k A_i/J$, $k = 1, \dots, n$. Given this scheme of selecting measuring times, it is calculated through simulation how the precision of estimated parameters depends on the number of data points and how those data points should be optimally distributed across different metabolites and times.

We last note that a typical metabolic network exhibits strong separation of concentration scales. For the example of glycolysis, one source reports that the most abundant metabolite (glucose) is about 35 times more than the second most abundant one and the full range of variation spans over three orders of magnitude [36]. This has an important implication: the most abundant metabolite dominates the numerator of $\sum_{i=1}^k A_i/J$, and hence the sampling time. If the dominant metabolite happens to be the first one, as is suggested for glycolysis in [36], then the whole pathway effectively shares one dynamics and measuring time.

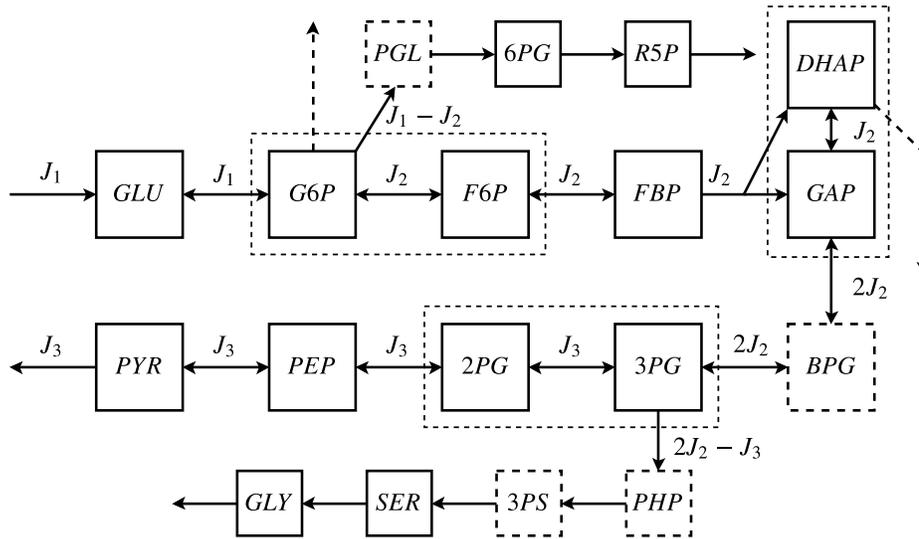
4.2.5 Applying rKFP to glucose deprivation data

In this last section, we apply rKFP to experimental data collected on glycolysis and its two main branching pathways (pentose-phosphate pathway, or PPP, and serine synthesis pathway) from cells in normal (5mM glucose) and glucose-deprived (0.5mM glucose) media conditions, and estimate the relative changes in the fluxes of the three pathways.

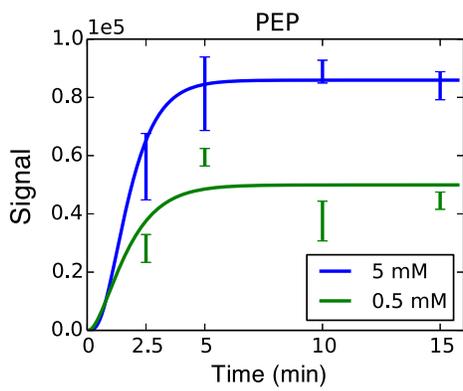
Figure 4.5a shows the diagram of the network, where the upper pathway branching off glycolysis at G6P is (part of) the PPP, and the lower pathway branching off glycolysis at 3PG is (part of) the serine synthesis pathway; two additional branching pathways, glycogen synthesis pathway and glycerol synthesis pathway, emanating from G6P and DHAP, respectively, are also portrayed (the dashed arrows). Further described in the network diagram is our data coverage, which contains all types of missing data we have considered: data of metabolites BPG and a few others in branching pathways are missing, data of the glycogen and glycerol synthesis pathways are missing, isomers such as G6P/F6P, GAP/DHAP and 3PG/2PG are not distinguished, and the extent of reversibility of the reactions is not known.

We set up three computational models for rKFP, corresponding to three different choices of treating missing data as discussed in earlier sections, and the estimation results of all three models are shown in Figure 4.5c: dark blue histograms correspond to the choice recommended in the earlier sections, namely,

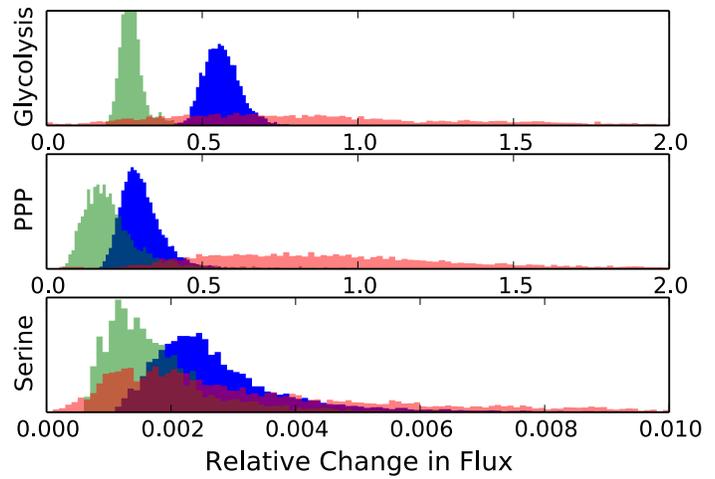
using the full model only for missing metabolite, light green the choice of using the reduced model for all types of missing data, and light red the choice of using the full model for both missing metabolite and pathway. We make two observations from the comparison of the histograms: the green histograms are tighter than the blue but shifted, and the red histograms have similar averages to the blue (except for PPP) but much larger variances.



(a)



(b)



(c)

Figure 4.5: Analysis of experimental data.

(Continue from last page.) (a) The diagram of glycolysis and its two branching pathways used in the analysis, where dashed squares (eg, BPG) represent missing metabolites, dashed arrows represent missing pathways, and dashed rectangles containing solid squares represent undistinguished metabolites. Abbreviations for metabolites: GLU: glucose; G6P: glucose-6-phosphate; F6P: fructose-6-phosphate; FBP: fructose-1,6-bisphosphate; DHAP: dihydroxyacetone phosphate; GAP: glyceraldehyde-3-phosphate; BPG: 1,3-bisphosphoglycerate; 3PG: 3-phosphoglycerate; 2PG: 2-phosphoglycerate; PEP: phosphoenolpyruvate; PYR: pyruvate; PGL: 6-phosphogluconolactone; 6PG: phosphogluconate; R5P: the pool of ribose 5-phosphate, ribulose 5-phosphate and xylulose 5-phosphate; PHP: phosphohydroxypyruvate; 3PS: 3-phosphoserine; SER: serine; GLY: glycine. (b) An exemplary plot of the data of a metabolite and its fit. (c) Histograms of r_j 's as generated by sampling the corresponding posterior distributions in a way detailed in the Methods. Glycolysis flux refers to J_3 in the diagram, PPP flux $J_1 - J_2$, and serine synthesis flux $2J_2 - J_3$. The three histograms for each flux correspond to three different modeling choices described in the text.

The observations support our recommended choice. The shifted green histograms relative to the blue suggest that metabolite removal introduces significant bias, likely because the pool sizes of the missing metabolites change differently from the other metabolites. The flattened red histograms relative to the blue suggest that keeping in the model the missing pathways with no data leaves the

estimation greatly underconstrained; on the other hand, their similar averages suggest that the bias introduced by pathway removal is small in this case, likely because the missing pathways have much lower fluxes than the main glycolysis pathway which is consistent with some previous studies (eg, [75, 70]) and that the cell line in our study exhibits the Warburg effect [112] with most of the glycolytic flux going to lactate fermentation. In summary, through the comparison of different models for both toy and real networks, we have chosen the following choice: in the face of missing data, we keep the model components that matter (metabolites) and leave out those that do not and would make the estimation underconstrained or numerically unstable (pathways, distinct pools and reversibilities). For future applications of rKFP, we suggest users either follow our recommended choice, or, better yet, set up different models and compare the results to verify the choice as is done here.

From the parameter distributions under our recommended model choice, we observe that, despite a 10-fold decrease in the media glucose concentration, glycolysis and PPP fluxes are reduced by about 40% and 60% respectively, while the serine pathway is basically incapacitated by the glucose deprivation. We make the following interpretation of the results. As it is generally believed that tumor cell proliferation sometimes requires metabolic adaptation to a microenvironment deprived of glucose, the effect of glucose deprivation on metabolism has been of interest [121, 54, 13]. Also of interest is the activity of serine biosynthesis pathway for its implication in tumorigenesis [64]. Here we show that when the external glucose source is depleted, the cells adapt their metabolism by largely maintain-

ing the backbone of glycolysis flux at the expense of serine biosynthesis flux. This suggests the possibility of additional contextual requirements of PHGDH, the enzyme that diverts flux from glycolysis into *de novo* serine biosynthesis, and the possibility of other mechanisms for serine utilization in the condition of low glucose availability.

4.3 Materials and methods

4.3.1 Computation

Individual steps of the computation in this study are listed and explained below using KFP as the example (many of them use *SloppyCell*, a Python package originally developed for analyzing biochemical networks [76]). Relevant Python codes are deposited at <http://github.com/leihuang/rkfp>.

- Encoding models: models are encoded in an SBML-compliant format [50] in *SloppyCell* and mathematically correspond to systems of ODEs of \mathbf{A}^* , the concentration vector of labeled metabolites.
- Simulating models: *daskr*, an algebraic-differential equation solver [18] used in *SloppyCell* numerically integrates the models.
- Generating simulated data: given the integrated trajectories of \mathbf{A}^* , simulated data are generated by a selection of measuring times following the sugges-

tions in the section on selecting measuring times, and the associated noise is generated by choosing a constant value when deriving analytical results for its tractability and assuming to be proportional to data when carrying out simulations for its realism (a proportionality constant of 0.2 is used).

- Estimating parameters: letting Y_{ik} be the measurement of the model prediction $A_i^*(t_k)$ and σ_{ik} be the noise associated with Y_{ik} , one defines the cost of fitting as a function of parameter θ : $C(\theta) \equiv \sum_{i,k} \left(\frac{A_i^*(t_k, \theta) - Y_{ik}}{\sigma_{ik}} \right)^2$ (also known as χ^2); SloppyCell uses the Levenberg-Marquardt algorithm [66] to minimize $C(\theta)$ and find the parameter estimate $\tilde{\theta}$. In our analysis of experimental data where the model is large and the data is noisy, we find that having a good initial guess is crucial [86] and having a large L-M parameter λ and a small trust region helps.
- Integrating sensitivity curves: SloppyCell calculates $\frac{\partial A_i^*}{\partial \theta_\alpha}$ as a function of t in an accurate way (more details in Text S1), which is important for calculating Jacobian and errors (below);
- Estimating errors: (1) construct the Jacobian of the model, $\mathbf{D} = \left(\frac{\partial A_i^*}{\partial \theta_\alpha}(t_k) \right)_{ik,\alpha}$ by filling a matrix with the sensitivities at the assumed measuring times along the sensitivity curves; (2) normalize each entry by σ_{ik} ; (3) perform the singular value decomposition $\mathbf{D} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$; (4) the square roots of the diagonal entries of the matrix $\mathbf{V}(\mathbf{\Sigma}\mathbf{\Sigma})^{-1}\mathbf{V}^T$ give the estimates of errors.
- Generating posterior distribution: assuming Gaussian-distributed measurement noise, the measurement of $A_i^*(t_k)$ is also Gaussian distributed: $Y_{ik} \sim \mathcal{N}(A_i^*(t_k, \theta), \sigma_{ik})$; treating the parameter estimation problem in the Bayesian

framework and assuming an uninformative prior, θ has a posterior distribution with a probability density proportional to that of observing Y_{ik} in $\mathcal{N}(A_i^*(t_k, \theta), \sigma_{ik})$, following the Bayes' rule: $p(\theta|\mathbf{Y}) \propto p_{\mathcal{N}}(\mathbf{Y}|\theta)$; Metropolis algorithm [24] is used to sample the posterior distribution in SloppyCell, and parameters of 100,000 steps, 1% burn-in and 50-step thinning interval [35] are used for generating the distributions in Figure 4.5c.

Note that the last two points constitute the two standard ways of estimating parameter uncertainties: the first one also goes by the name of *sensitivity analysis* or *delta method*, and is computationally cheap but less accurate; the second one is also known as *ensemble method* [17], and is computationally expensive but more accurate. In this study, simulations intended to illustrate basic principles use the first method, and data analyses intended to draw realistic conclusions use the second.

4.3.2 Experiments

Experimental procedures of cell culture, metabolite extraction, mass spectrometry, liquid chromatography and data processing follow those in [62]. On the procedures specific to rKFP, HCT116 human colon cancer cells cultured in 6 well dishes with RPMI 1640 were washed with PBS and transferred to two media with ^{12}C -glucose of concentrations 5mM and 500uM respectively, where they were incubated for 2.5 hours before switching to media with ^{13}C -glucose of the same con-

centrations; relative quantitation of triplicates were then performed on the cells at 0, 2.5, 5, 10 and 15 minutes after the switching.

BIBLIOGRAPHY

- [1] Uri Alon. *An Introduction to Systems Biology: Design Principles of Biological Circuits*. CRC Press, July 2006. Google-Books-ID: pAUdPQICZ54C.
- [2] Shun-ichi Amari and Hiroshi Nagaoka. *Methods of Information Geometry*, 2000.
- [3] P. W. Anderson. More Is Different. *Science*, 177(4047):393–396, August 1972.
- [4] Maciek R. Antoniewicz, Joanne K. Kelleher, and Gregory Stephanopoulos. Elementary Metabolite Units (EMU): a novel framework for modeling isotopic distributions. *Metabolic Engineering*, 9(1):68–86, January 2007.
- [5] Joshua F. Apgar, David K. Witmer, Forest M. White, and Bruce Tidor. Sloppy models, parameter uncertainty, and the role of experimental design. *Molecular bioSystems*, 6(10):1890–1900, October 2010.
- [6] Stphanie Arrivault, Manuela Guenther, Alexander Ivakov, Regina Feil, Daniel Vosloh, Joost T. Van Dongen, Ronan Sulpice, and Mark Stitt. Use of reverse-phase liquid chromatography, linked to tandem mass spectrometry, to profile the Calvin cycle and other metabolic intermediates in Arabidopsis rosettes at different carbon dioxide concentrations. *The Plant Journal*, 59(5):826–839, September 2009.
- [7] A. C. Atkinson and A. N. Donev. *Optimum Experimental Designs*. Oxford University Press, November 1992.
- [8] Arren Bar-Even, Elad Noor, Yonatan Savir, Wolfram Liebermeister, Dan Davidi, Dan S. Tawfik, and Ron Milo. The Moderately Efficient Enzyme: Evolutionary and Physicochemical Trends Shaping Enzyme Parameters. *Biochemistry*, 50(21):4402–4410, May 2011.
- [9] D. Battogtokh, D. K. Asch, M. E. Case, J. Arnold, and H.-B. Schttler. An ensemble method for identifying regulatory circuits with special reference to the qa gene cluster of *Neurospora crassa*. *Proceedings of the National Academy of Sciences of the United States of America*, 99(26):16904–16909, December 2002.

- [10] Daniel A. Beard and Hong Qian. *Chemical Biophysics: Quantitative Analysis of Cellular Systems*. Cambridge University Press, November 2010.
- [11] Daniel I Benjamin, Benjamin F Cravatt, and Daniel K Nomura. Global profiling strategies for mapping dysregulated metabolic pathways in cancer. *Cell Metabolism*, 16(5):565–577, November 2012.
- [12] Bryson D Bennett, Elizabeth H Kimball, Melissa Gao, Robin Osterhout, Stephen J Van Dien, and Joshua D Rabinowitz. Absolute Metabolite Concentrations and Implied Enzyme Active Site Occupancy in *Escherichia coli*. *Nature Chemical Biology*, 5(8):593–599, August 2009.
- [13] Kvan Birsoy, Richard Possemato, Franziska K Lorbeer, Erol C Bayraktar, Prathapan Thiru, Burcu Yucel, Tim Wang, Walter W Chen, Clary B Clish, and David M Sabatini. Metabolic determinants of cancer cell sensitivity to glucose limitation and biguanides. *Nature*, 508(7494):108–112, April 2014.
- [14] Rogier Braakman and Eric Smith. The compositional and evolutionary logic of metabolism. *Physical Biology*, 10(1):011001, February 2013. arXiv: 1207.5532.
- [15] George Edward Briggs and John Burdon Sanderson Haldane. A Note on the Kinetics of Enzyme Action. *Biochemical Journal*, 19(2):338–339, 1925.
- [16] K. S. Brown, C. C. Hill, G. A. Calero, C. R. Myers, K. H. Lee, J. P. Sethna, and R. A. Cerione. The statistical mechanics of complex signaling networks: nerve growth factor signaling. *Physical Biology*, 1(3-4):184–195, December 2004.
- [17] Kevin S. Brown and James P. Sethna. Statistical mechanical approaches to models with many poorly known parameters. *Physical Review E*, 68(2):021904, August 2003.
- [18] Peter N. Brown, Alan C. Hindmarsh, and Linda R. Petzold. Using Krylov methods in the solution of large-scale differential-algebraic systems. *SIAM Journal on Scientific Computing*, 15:1467–1488, 1994.

- [19] S. von Caemmerer and G. D. Farquhar. Some relationships between the biochemistry of photosynthesis and the gas exchange of leaves. *Planta*, 153(4):376–387, December 1981.
- [20] Susanne von Caemmerer, Graham Farquhar, and Joseph Berry. Biochemical Model of C3 Photosynthesis. In Agu Laisk, Ladislav Nedbal, and Govindjee, editors, *Photosynthesis in silico*, number 29 in Advances in Photosynthesis and Respiration, pages 209–230. Springer Netherlands, 2009. DOI: 10.1007/978-1-4020-9237-4_9.
- [21] Susanne von Caemmerer, W. Paul Quick, and Robert T. Furbank. The Development of C4 Rice: Current Progress and Future Challenges. *Science*, 336(6089):1671–1672, June 2012.
- [22] Ricky Chachra, Mark K. Transtrum, and James P. Sethna. Comment on Sloppy models, parameter uncertainty, and the role of experimental design. 7(8):2522–2522, August 2011.
- [23] B. Chance, D. Garfinkel, J. Higgins, and B. Hess. Metabolic control mechanisms. 5. A solution for the equations representing interaction between glycolysis and respiration in ascites tumor cells. *The Journal of Biological Chemistry*, 235:2426–2439, August 1960.
- [24] Siddhartha Chib and Edward Greenberg. Understanding the Metropolis-Hastings Algorithm. *The American Statistician*, 49(4):327–335, November 1995.
- [25] Oana-Teodora Chis, Julio R. Banga, and Eva Balsa-Canto. Structural Identifiability of Systems Biology Models: A Critical Comparison of Methods. *PLOS ONE*, 6(11):e27755, November 2011.
- [26] Simon A. Cobbold, Ashley M. Vaughan, Ian A. Lewis, Heather J. Painter, Nelly Camargo, David H. Perlman, Matthew Fishbaugher, Julie Healer, Alan F. Cowman, Stefan H. I. Kappe, and Manuel Llins. Kinetic Flux Profiling Elucidates Two Independent Acetyl-CoA Biosynthetic Pathways in *Plasmodium falciparum*. *Journal of Biological Chemistry*, 288(51):36338–36350, December 2013.

- [27] Athel Cornish-Bowden. *Fundamentals of Enzyme Kinetics*. Wiley-Blackwell, Weinheim, Germany, 4 edition, February 2012.
- [28] Athel Cornish-Bowden. One hundred years of Michaelis-Menten kinetics. *Perspectives in Science*, 4:3–9, March 2015.
- [29] Francis Corson and Eric Dean Siggia. Geometry, epistasis, and developmental patterning. *Proceedings of the National Academy of Sciences*, 109(15):5568–5575, April 2012.
- [30] Emilie Dufresne, Heather A. Harrington, and Dhruva V. Raman. The geometry of sloppiness. *arXiv:1608.05679 [physics, q-bio, stat]*, August 2016. arXiv:1608.05679.
- [31] G. D. Farquhar, S. von Caemmerer, and J. A. Berry. A biochemical model of photosynthetic CO₂ assimilation in leaves of C₃ species. *Planta*, 149(1):78–90, June 1980.
- [32] Martin Feinberg. *Lectures On Chemical Reaction Networks*, 1979.
- [33] Martin Feinberg. Chemical reaction network structure and the stability of complex isothermal reactors. I. The deficiency zero and deficiency one theorems. *Chemical Engineering Science*, 42(10):2229–2268, January 1987.
- [34] David Fell. *Understanding the Control of Metabolism*. Portland Press, 1 edition, November 1996.
- [35] Dani Gamerman and Hedibert F. Lopes. *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*. CRC Press, May 2006.
- [36] Reginald Garrett and Charles Grisham. *Biochemistry*. Cengage Learning, 5 edition, January 2012.
- [37] Lea Goentoro and Marc W. Kirschner. Evidence that Fold-Change, and Not Absolute Level, of β -Catenin Dictates Wnt Signaling. *Molecular Cell*, 36(5):872–884, December 2009.

- [38] Sergio Grimbs, Anne Arnold, Aneta Koseska, Jrgen Kurths, Joachim Selbig, and Zoran Nikoloski. Spatiotemporal dynamics of the Calvin cycle: Multistationarity and symmetry breaking instabilities. *Biosystems*, 103(2):212–223, February 2011.
- [39] Jeremy Gunawardena. Chemical reaction network theory for in-silico biologists. 2003.
- [40] Jeremy Gunawardena. Models in Systems Biology: The Parameter Problem and the Meanings of Robustness. In Huma M. Lodhi and Stephen H. Muggleton, editors, *Elements of Computational Systems Biology*, pages 19–47. John Wiley & Sons, Inc., 2010.
- [41] Jeremy Gunawardena. A Linear Framework for Time-Scale Separation in Nonlinear Biochemical Systems. *PLOS ONE*, 7(5):e36321, May 2012.
- [42] Jeremy Gunawardena. Models in biology: accurate descriptions of our pathetic thinking. *BMC Biology*, 12:29, 2014.
- [43] Jeremy Gunawardena. Systems Biology, Minimalist vs Exhaustive Strategies. In Bjrn Engquist, editor, *Encyclopedia of Applied and Computational Mathematics*, pages 1458–1463. Springer Berlin Heidelberg, 2015. DOI: 10.1007/978-3-540-70529-1_87.
- [44] Ryan N. Gutenkunst, Fergal P. Casey, Joshua J. Waterfall, Christopher R. Myers, and James P. Sethna. Extracting falsifiable predictions from sloppy models. *Annals of the New York Academy of Sciences*, 1115:203–211, December 2007.
- [45] Ryan N Gutenkunst, Joshua J Waterfall, Fergal P Casey, Kevin S Brown, Christopher R Myers, and James P Sethna. Universally Sloppy Parameter Sensitivities in Systems Biology Models. *PLoS Comput Biol*, 3(10):e189, October 2007.
- [46] B. D. Hahn. Photosynthesis and photorespiration: Modelling the essentials. *Journal of Theoretical Biology*, 151(1):123–139, July 1991.

- [47] David Heckmann, Stefanie Schulze, Alisandra Denton, Udo Gowik, Peter Westhoff, Andreas P. M. Weber, and Martin J. Lercher. Predicting C4 Photosynthesis Evolution: Modular, Individually Adaptive Steps on a Mount Fuji Fitness Landscape. *Cell*, 153(7):1579–1588, June 2013.
- [48] R. Heinrich and T. A. Rapoport. A linear steady-state treatment of enzymatic chains. General properties, control and effector strength. *European Journal of Biochemistry*, 42(1):89–95, February 1974.
- [49] Lei Huang, Dongsung Kim, Xiaojing Liu, Christopher R. Myers, and Jason W. Locasale. Estimating Relative Changes of Metabolic Fluxes. *PLOS Computational Biology*, 10(11):e1003958, November 2014.
- [50] M. Hucka, A. Finney, H. M. Sauro, H. Bolouri, J. C. Doyle, H. Kitano, A. P. Arkin, B. J. Bornstein, D. Bray, A. Cornish-Bowden, A. A. Cuellar, S. Dronov, E. D. Gilles, M. Ginkel, V. Gor, I. I. Goryanin, W. J. Hedley, T. C. Hodgman, J.-H. Hofmeyr, P. J. Hunter, N. S. Juty, J. L. Kasberger, A. Kremling, U. Kummer, N. Le Novre, L. M. Loew, D. Lucio, P. Mendes, E. Minch, E. D. Mjolsness, Y. Nakayama, M. R. Nelson, P. F. Nielsen, T. Sakurada, J. C. Schaff, B. E. Shapiro, T. S. Shimizu, H. D. Spence, J. Stelling, K. Takahashi, M. Tomita, J. Wagner, and J. Wang. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531, March 2003.
- [51] H. Kacser and J. A. Burns. The control of flux. *Symposia of the Society for Experimental Biology*, 27:65–104, 1973.
- [52] Sayako Katada, Axel Imhof, and Paolo Sassone-Corsi. Connecting threads: epigenetics and metabolism. *Cell*, 148(1-2):24–28, January 2012.
- [53] Edward L. King and Carl Altman. A Schematic Method of Deriving the Rate Laws for Enzyme-Catalyzed Reactions. *The Journal of Physical Chemistry*, 60(10):1375–1378, October 1956.
- [54] Kakajan Komurov, Jen-Te Tseng, Melissa Muller, Elena G Seviour, Tyler J Moss, Lifeng Yang, Deepak Nagrath, and Prahlad T Ram. The glucose-

- deprivation network counteracts lapatinib-induced toxicity in resistant ErbB2-positive breast cancer cells. *Molecular Systems Biology*, 8:596, 2012.
- [55] Patrick De Leenheer, David Angeli, and Eduardo D. Sontag. Monotone Chemical Reaction Networks. *Journal of Mathematical Chemistry*, 41(3):295–314, April 2007.
- [56] Stephane Lefebvre, Tracy Lawson, Mike Fryer, Oksana V. Zakhleniuk, Julie C. Lloyd, and Christine A. Raines. Increased Sedoheptulose-1,7-Bisphosphatase Activity in Transgenic Tobacco Plants Stimulates Photosynthesis and Growth from an Early Stage in Development. *Plant Physiology*, 138(1):451–460, May 2005.
- [57] Hong-Bo Lei, Xin Wang, Ruiqi Wang, Xin-Guang Zhu, Luonan Chen, and Ji-Feng Zhang. A parameter condition for ruling out multiple equilibria of the photosynthetic carbon metabolism. *Asian Journal of Control*, 13(5):611–624, September 2011.
- [58] Melissa Lever, Philip K. Maini, P. Anton van der Merwe, and Omer Dushek. Phenotypic models of T cell activation. *Nature Reviews Immunology*, 14(9):619–629, September 2014.
- [59] Nathan E. Lewis, Harish Nagarajan, and Bernhard O. Palsson. Constraining the metabolic genotype-phenotype relationship using a phylogeny of in silico methods. *Nature Reviews Microbiology*, 10(4):291–305, April 2012.
- [60] Wolfram Liebermeister. Enzyme economy in metabolic networks. *arXiv:1404.5252 [q-bio]*, April 2014. arXiv: 1404.5252.
- [61] Wolfram Liebermeister and Edda Klipp. Bringing metabolic networks to life: integration of kinetic, metabolic, and proteomic data. *Theoretical Biology and Medical Modelling*, 3:42, 2006.
- [62] Xiaojing Liu, Zheng Ser, and Jason W Locasale. Development and Quantitative Evaluation of a High-Resolution Metabolomics Technology. *Analytical Chemistry*, 86(4):2175–2184, February 2014.

- [63] Jason W Locasale. The consequences of enhanced cell-autonomous glucose metabolism. *Trends in Endocrinology and Metabolism*, 23(11):545–551, November 2012.
- [64] Jason W. Locasale, Alexandra R. Grassian, Tamar Melman, Costas A. Lyssiotis, Katherine R. Mattaini, Adam J. Bass, Gregory Heffron, Christian M. Metallo, Taru Muranen, Hadar Sharfi, Atsuo T. Sasaki, Dimitrios Anastasiou, Edouard Mullarky, Natalie I. Vokes, Mika Sasaki, Rameen Beroukhim, Gregory Stephanopoulos, Azra H. Ligon, Matthew Meyerson, Andrea L. Richardson, Lynda Chin, Gerhard Wagner, John M. Asara, Joan S. Brugge, Lewis C. Cantley, and Matthew G. Vander Heiden. Phosphoglycerate dehydrogenase diverts glycolytic flux and contributes to oncogenesis. *Nature Genetics*, 43(9):869–874, September 2011.
- [65] Benjamin B. Machta, Ricky Chachra, Mark K. Transtrum, and James P. Sethna. Parameter space compression underlies emergent theories and predictive models. *Science (New York, N.Y.)*, 342(6158):604–607, November 2013.
- [66] D. W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11:431–441, 1963.
- [67] Hector Garca Martn, Vinay Satish Kumar, Daniel Weaver, Amit Ghosh, Victor Chubukov, Aindrila Mukhopadhyay, Adam Arkin, and Jay D. Keasling. A Method to Constrain Genome-Scale Models with ¹³C Labeling Data. *PLoS Computational Biology*, 11(9):e1004363, September 2015.
- [68] Steven L McKnight. On getting there from here. *Science*, 330(6009):1338–1339, December 2010.
- [69] Mahya Mehrmohamadi, Lucas K. Mentch, Andrew G. Clark, and Jason W. Locasale. Integrative modelling of tumour DNA methylation quantifies the contribution of metabolism. *Nature Communications*, 7:13666, December 2016.
- [70] Christian M. Metallo, Jason L. Walther, and Gregory Stephanopoulos. Eval-

uation of ^{13}C isotopic tracers for metabolic flux analysis in mammalian cells. *Journal of Biotechnology*, 144(3):167–174, November 2009.

- [71] Tabea Mettler, Timo Mhlhaus, Dorothea Hemme, Mark-Aurel Schttler, Jens Rupprecht, Adam Idoine, Daniel Veyel, Sunil Kumar Pal, Liliya Yaneva-Roder, Flavia Vischi Winck, Frederik Sommer, Daniel Vosloh, Bettina Seiwert, Alexander Erban, Asdrubal Burgos, Samuel Arvidsson, Stephanie Schnfelder, Anne Arnold, Manuela Gnther, Ursula Krause, Marc Lohse, Joachim Kopka, Zoran Nikoloski, Bernd Mueller-Roeber, Lothar Willmitzer, Ralph Bock, Michael Schroda, and Mark Stitt. Systems Analysis of the Response of Photosynthesis, Metabolism, and Growth to an Increase in Irradiance in the Photosynthetic Model Organism *Chlamydomonas reinhardtii*. *The Plant Cell*, 26(6):2310–2350, June 2014.
- [72] L. Michaelis and Miss Maud L. Menten. The kinetics of invertin action: Translated by T.R.C. Boyde Submitted 4 February 1913. *FEBS Letters*, 587(17):2712–2720, September 2013.
- [73] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network Motifs: Simple Building Blocks of Complex Networks. *Science*, 298(5594):824–827, October 2002.
- [74] Fangping Mu, Robert F Williams, Clifford J Unkefer, Pat J Unkefer, James R Faeder, and William S Hlavacek. Carbon-fate maps for metabolic reactions. *Bioinformatics*, 23(23):3193–3199, December 2007.
- [75] Joshua Munger, Bryson D Bennett, Anuraag Parikh, Xiao-Jiang Feng, Jessica McArdle, Herschel A Rabitz, Thomas Shenk, and Joshua D Rabinowitz. Systems-level metabolic flux profiling identifies fatty acid synthesis as a target for antiviral therapy. *Nature Biotechnology*, 26(10):1179–1186, October 2008.
- [76] Christopher R. Myers, Ryan N. Gutenkunst, and James P. Sethna. Python Unleashed on Systems Biology. *Computing in Science & Engineering*, 9(3):34–37, May 2007.

- [77] David L. Nelson and Michael M. Cox. *Lehninger Principles of Biochemistry*. W.H. Freeman, New York, 6 edition edition, November 2012.
- [78] Stephan Noack and Wolfgang Wiechert. Quantitative metabolomics: a phantom? *Trends in Biotechnology*, 32(5):238–244, May 2014.
- [79] Elad Noor, Arren Bar-Even, Avi Flamholz, Yaniv Lubling, Dan Davidi, and Ron Milo. An integrated open framework for thermodynamics of reactions that combines accuracy and coverage. *Bioinformatics*, 28(15):2037–2044, August 2012.
- [80] Elad Noor, Arren Bar-Even, Avi Flamholz, Ed Reznik, Wolfram Liebermeister, and Ron Milo. Pathway Thermodynamics Highlights Kinetic Obstacles in Central Metabolism. *PLoS Comput Biol*, 10(2):e1003483, February 2014.
- [81] Jeffrey D. Orth, Ines Thiele, and Bernhard Palsson. What is flux balance analysis? *Nature Biotechnology*, 28(3):245–248, March 2010.
- [82] Bernhard Palsson. *Systems Biology: Constraint-based Reconstruction and Analysis*. Cambridge University Press, January 2015. Google-Books-ID: QNBp-BgAAQBAJ.
- [83] Gary J Patti, Oscar Yanes, and Gary Siuzdak. Metabolomics: the apogee of the omic trilogy. *Nature Reviews Molecular Cell Biology*, 13(4):263–269, March 2012.
- [84] Erika L. Pearce, Maya C. Poffenberger, Chih-Hao Chang, and Russell G. Jones. Fueling Immunity: Insights into Metabolism and Lymphocyte Function. *Science*, 342(6155):1242454, October 2013.
- [85] M. G. Poolman, H. ler, J. C. Lloyd, C. A. Raines, and D. A. Fell. Computer modelling and experimental evidence for two steady states in the photosynthetic Calvin cycle. *European Journal of Biochemistry*, 268(10):2810–2816, 2001.
- [86] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flan-

nery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, Cambridge, UK ; New York, 3 edition, September 2007.

- [87] Przemyslaw Prusinkiewicz and Adam Runions. Computational models of plant development and form. *New Phytologist*, 193(3):549–569, February 2012.
- [88] J D Rabinowitz, J G Purdy, L Vastag, T Shenk, and E Koyuncu. Metabolomics in drug target discovery. *Cold Spring Harbor symposia on quantitative biology*, 76:235–246, 2011.
- [89] Joshua D. Rabinowitz and Livia Vastag. Teaching the design principles of metabolism. *Nature Chemical Biology*, 8(6):497–501, May 2012.
- [90] Alan D. Rendall. A Calvin bestiary. *arXiv:1609.04926 [math, q-bio]*, September 2016. arXiv: 1609.04926.
- [91] Marc Roussel. *Singular perturbation theory*, 2005.
- [92] Martin Rhl, Dominique Le Coq, Stphane Aymerich, and Uwe Sauer. 13c-flux Analysis Reveals NADPH-balancing Transhydrogenation Cycles in Stationary Phase of Nitrogen-starving *Bacillus subtilis*. *Journal of Biological Chemistry*, 287(33):27959–27970, August 2012.
- [93] Pedro A. Saa and Lars K. Nielsen. Construction of feasible and accurate kinetic models of metabolism: A Bayesian approach. *Scientific Reports*, 6:29635, July 2016.
- [94] Michael A. Savageau. Introduction to S-systems and the underlying power-law formalism. *Mathematical and Computer Modelling*, 11:546–551, January 1988.
- [95] Ida Schomburg, Antje Chang, and Dietmar Schomburg. BRENDA, enzyme data and metabolic information. *Nucleic Acids Research*, 30(1):47–49, January 2002.

- [96] Robert Schuetz, Lars Kuepfer, and Uwe Sauer. Systematic evaluation of objective functions for predicting intracellular fluxes in *Escherichia coli*. *Molecular Systems Biology*, 3(1):119, January 2007.
- [97] James Sethna. *Statistical Mechanics: Entropy, Order Parameters and Complexity*. OUP Oxford, April 2006.
- [98] Alexander A. Shestov, Xiaojing Liu, Zheng Ser, Ahmad A. Cluntun, Yin P. Hung, Lei Huang, Dongsung Kim, Anne Le, Gary Yellen, John G. Albeck, and Jason W. Locasale. Quantitative determinants of aerobic glycolysis identify flux through the enzyme GAPDH as a limiting step. *eLife*, 3:e03342, July 2014.
- [99] Natalie J. Stanford, Timo Lubitz, Kieran Smallbone, Edda Klipp, Pedro Mendes, and Wolfram Liebermeister. Systematic Construction of Kinetic Models from Genome-Scale Metabolic Networks. *PLOS ONE*, 8(11):e79195, November 2013.
- [100] Mark Stitt and Yves Gibon. Why measure enzyme activities in the era of systems biology? *Trends in Plant Science*, 19(4):256–265, April 2014.
- [101] Marek Szecowka, Robert Heise, Takayuki Tohge, Adriano Nunes-Nesi, Daniel Vosloh, Jan Huege, Regina Feil, John Lunn, Zoran Nikoloski, Mark Stitt, Alisdair R Fernie, and Stphanie Arrivault. Metabolic fluxes in an illuminated *Arabidopsis* rosette. *Plant Cell*, 25(2):694–714, February 2013.
- [102] Linh M. Tran, Matthew L. Rizk, and James C. Liao. Ensemble Modeling of Metabolic Networks. *Biophysical Journal*, 95(12):5606–5617, December 2008.
- [103] Mark Transtrum, Benjamin Machta, and James Sethna. Geometry of non-linear least squares with applications to sloppy models and optimization. *Physical Review E*, 83(3), March 2011.
- [104] Mark K. Transtrum. Manifold boundaries give “gray-box” approximations of complex models. *arXiv:1605.08705 [physics]*, May 2016. arXiv: 1605.08705.
- [105] Mark K. Transtrum, Gus Hart, and Peng Qiu. Information topology identi-

- fies emergent model classes. *arXiv:1409.6203 [cond-mat, physics:physics, stat]*, September 2014. arXiv: 1409.6203.
- [106] Mark K. Transtrum and Peng Qiu. Model Reduction by Manifold Boundaries. *Physical Review Letters*, 113(9):098701, August 2014.
- [107] Mark K. Transtrum and Peng Qiu. Bridging Mechanistic and Phenomenological Models of Complex Biological Systems. *PLOS Computational Biology*, 12(5):e1004915, May 2016.
- [108] Mark K. Transtrum and James P. Sethna. Improvements to the Levenberg-Marquardt algorithm for nonlinear least-squares minimization. *arXiv:1201.5885 [physics]*, January 2012. arXiv: 1201.5885.
- [109] Katja Tummler, Timo Lubitz, Max Schelker, and Edda Klipp. New types of experimental data shape the use of enzyme kinetics for dynamic network modeling. *The FEBS journal*, 281(2):549–571, January 2014.
- [110] Karen van Eunen, Jildau Bouwman, Pascale Daran-Lapujade, Jarne Postmus, Andr B. Canelas, Femke I. C. Mensonides, Rick Orij, Isil Tuzun, Joost van den Brink, Gertien J. Smits, Walter M. van Gulik, Stanley Brul, Joseph J. Heijnen, Johannes H. de Winde, M. Joost Teixeira de Mattos, Carsten Kettner, Jens Nielsen, Hans V. Westerhoff, and Barbara M. Bakker. Measuring enzyme activities under standardized in vivo-like conditions for systems biology. *The FEBS journal*, 277(3):749–760, February 2010.
- [111] Matthew G. Vander Heiden. Targeting cancer metabolism: a therapeutic window opens. *Nature Reviews Drug Discovery*, 10(9):671–684, September 2011.
- [112] Matthew G. Vander Heiden, Lewis C. Cantley, and Craig B. Thompson. Understanding the Warburg Effect: the metabolic requirements of cell proliferation. *Science*, 324(5930):1029–1033, May 2009.
- [113] Donald Voet, Judith G. Voet, and Charlotte W. Pratt. *Fundamentals of Biochemistry: Life at the Molecular Level*. John Wiley & Sons, Inc., 2011.

- [114] Susanne Von Caemmerer. Steady-state models of photosynthesis. *Plant, Cell & Environment*, 36(9):1617–1630, September 2013.
- [115] Wolfgang Wiechert. 13c Metabolic Flux Analysis. *Metabolic Engineering*, 3(3):195–206, July 2001.
- [116] Ulrike Wittig, Renate Kania, Martin Golebiewski, Maja Rey, Lei Shi, Lenneke Jong, Enkhjargal Alгаа, Andreas Weidemann, Heidrun Sauer-Danzwith, Saqib Mir, Olga Krebs, Meik Bittkowski, Elina Wetsch, Isabel Rojas, and Wolfgang Mller. SABIO-RKdatabase for biochemical reaction kinetics. *Nucleic Acids Research*, 40(Database issue):D790–D796, January 2012.
- [117] Guoyao Wu, Fuller W. Bazer, Teresa A. Davis, Sung Woo Kim, Peng Li, J. Marc Rhoads, M. Carey Satterfield, Stephen B. Smith, Thomas E. Spencer, and Yulong Yin. Arginine metabolism and nutrition in growth, health and disease. *Amino acids*, 37(1):153–168, May 2009.
- [118] Liang Wu, Weiming Wang, Wouter A. van Winden, Walter M. van Gulik, and Joseph J. Heijnen. A new framework for the estimation of control parameters in metabolic pathways using lin-log kinetics. *European Journal of Biochemistry*, 271(16):3348–3359, August 2004.
- [119] Jie Yuan, Bryson D Bennett, and Joshua D Rabinowitz. Kinetic flux profiling for quantitation of cellular metabolic fluxes. *Nature Protocols*, 3(8):1328–1340, 2008.
- [120] Jie Yuan, William U. Fowler, Elizabeth Kimball, Wenyun Lu, and Joshua D. Rabinowitz. Kinetic flux profiling of nitrogen assimilation in *Escherichia coli*. *Nature Chemical Biology*, 2(10):529–530, October 2006.
- [121] Jihye Yun, Carlo Rago, Ian Cheong, Ray Pagliarini, Philipp Angenendt, Harith Rajagopalan, Kerstin Schmidt, James K V Willson, Sandy Markowitz, Shibin Zhou, Luis A Diaz, Jr, Victor E Velculescu, Christoph Lengauer, Kenneth W Kinzler, Bert Vogelstein, and Nickolas Papadopoulos. Glucose deprivation contributes to the development of KRAS pathway mutations in tumor cells. *Science*, 325(5947):1555–1559, September 2009.

- [122] Xin-Guang Zhu, Rafael Alba, and Eric de Sturler. A simple model of the Calvin cycle has only one physiologically feasible steady state under the same external conditions. *Nonlinear Analysis: Real World Applications*, 10(3):1490–1499, June 2009.