SUPPORTING INFORMATION

Matthew Thomson,[1] Jeremy Gunawardena,[2*]

[1]Biophysics Program, Harvard University
Cambridge, MA 02138, USA

[2]Department of Systems Biology, Harvard Medical School
200 Longwood Avenue, Boston, MA 02115, USA

[*]To whom correspondence should be addressed; E-mail: jeremy@hms.harvard.edu

Abstract

This note provides supporting information for *"Unlimited multistability in multisite phosphorylation systems"*. It should be read in conjunction with the paper. The material is roughly organised in the order in which it appears in the paper but the table of contents provides more information.

Contents

1 Rationality and positivity

This section provides an outline of the proof of the two main mathematical results required for the paper: the rationality of the phospho-forms at steady state (Paper Equation 1) and the positivity of the rational functions. The key concepts are defined and the results stated in sufficient detail that an expert should be able to reconstruct the argument in full. Complete details of the proofs will appear elsewhere [35].

1.1 The network of reactions

Paper Figure 1 used a notation for phospho-forms in which the subscript is a bit string, indicating which sites are phosphorylated. It will be more convenient here to index the 2^n phospho-forms by an integer. Let index i correspond to the n-bit string given by the expansion of i modulo 2, padded with leading zeros as required, so that i lies in the range $[0, 2^n - 1]$. It will be helpful to let $N = 2^n - 1$. S_0 now corresponds to the unphosphorylated phospho-form and S_N to the fully phosphorylated phospho-form. For $n = 3$ the 8 phospho-forms in Paper Figure 1 are encoded as follows:

$$S_0 \equiv S_{000},\ S_1 \equiv S_{001},\ S_2 \equiv S_{010},\ S_3 \equiv S_{011},\ S_4 \equiv S_{100},\ 5 \equiv S_{101},\ S_6 \equiv S_{110},\ S_7 \equiv S_{111}\,.$$

Let $[i] \subseteq \{1, \cdots, n\}$ denote the set of bit positions (counting from the right, starting from 1 and going up to n) at which the bit string of i has a 1, corresponding to the presence of a phosphate. Let $i \rightarrow j$ indicate that there is a reaction having S_i as substrate and S_j as product. This happens only if, either $[i] \subseteq [j]$, in which case phosphates are gained and it is a kinase reaction, or $[j] \subseteq [i]$, in which case phosphates are lost and it is a phosphatase reaction. Note that some reactions, like $3 \rightarrow 5$, are ruled out. The notation allows for both distributive reactions, in which exactly one phosphate is gained or lost (Paper Figure 1a) and processive reactions, in which more than one phosphate is gained or lost (Paper Figure 1b). It will sometimes be convenient to confirm that a reaction is a kinase or a phosphatase reaction by using the extra annotation $i \xrightarrow{X} j$, where $X = E$ or $X = F$, respectively.

It could be, for some specific kinase, phosphatase, substrate system, that certain phospho-forms do not appear. In this case, we assume that they are not present in the model and the value of N is modified accordingly (see Example 1 in §1.7). The assertion $i \rightarrow j$ carries with it the assumption that the parameters arising from the underlying biochemistry (Paper Figure 1c) are all positive:

$$a_i^X\,,\ \ b_i^X\,,\ \ c_{i,j}^X > 0\,.$$

The set of assertions $i \rightarrow j$ defines a directed graph on the set of phospho-forms, which summarises the network of biochemical reactions of the particular multisite phosphorylation system being studied. For the main results of the paper, we assume a nonsequential system with n sites, as illustrated in Paper Figure 1 for $n = 3$. All 2^n phospho-forms are assumed to appear and the kinase and phosphatase are assumed to act at least distributively (Paper Figure 1a). The kinase and phosphatase can also act with arbitrary amounts of processivity (Paper Figure 1b). We also use two other examples for specific calculations, as described in §1.7.

1.2 The differential equations

We can now rewrite the model equations (Paper Figure 1d) using this new notation.

$$\frac{dS_i}{dt} = \sum_{X=E,F} \{b_i^X XS_i - a_i^X X.S_i + \sum_{k \xrightarrow{X} i} c_{k,i}^X XS_k\} \tag{1}$$

$$\frac{dXS_i}{dt} = a_i^X X.S_i - (b_i^X + \sum_{i \xrightarrow{X} j} c_{i,j}^X) XS_i \tag{2}$$

$$\frac{dX}{dt} = \sum_{\substack{i \\ \exists j,\, i \xrightarrow{X} j}} \{-a_i^X X.S_i + (b_i^X + \sum_{i \xrightarrow{X} j} c_{i,j}^X) XS_i\} \tag{3}$$

Equation (2) summarises both the case $X = E$, for which $0 \leq i < N$ (because there is no enzyme-substrate complex ES_N), and the case $X = F$, for which $0 < i \leq N$ (because there is no enzyme-substrate complex FS_0). In (3), the restriction $\exists j,\, i \xrightarrow{X} j$ is necessary to avoid including in the sum these same non-existent enzyme-substrate complexes. With the new notation, this can be made clearer by separating (3) into its two constituent equations, which have different summation ranges.

$$\frac{dE}{dt} = \sum_{i=0}^{N-1} \{-a_i^E E.S_i + (b_i^E + \sum_{i \xrightarrow{E} j} c_{i,j}^E) ES_i\} \tag{4}$$

$$\frac{dF}{dt} = \sum_{i=1}^{N} \{-a_i^F F.S_i + (b_i^F + \sum_{i \xrightarrow{F} j} c_{i,j}^F) FS_i\} \tag{5}$$

1.3 Parameter algebra over $\mathbb{R}(a)$

If equation (2) is considered at steady state, so that $dXS_i/dt = 0$, it is easy to see that

$$XS_i = \frac{X.S_i}{K_i^X}\,, \tag{6}$$

provided a generalised Michaelis-Menten constant, K_i^X, is defined by

$$K_i^X = \frac{b_i^X + \sum_{i \xrightarrow{X} j} c_{i,j}^X}{a_i^X}\,. \tag{7}$$

Let $a_1, \cdots, a_m$ be a list of the parameters in the model. These consist of the various site-specific rate constants, $a_i^X, b_i^X, c_{i,j}^X$, for $X = E$, $0 \leq i < N$ and $X = F$, $0 < i \leq N$, in some order. It will be necessary to form algebraic expressions involving the parameters, as in (7). These expressions will be either polynomials or rational functions, with real coefficients. (For more information on the material in this section, see, for instance, [22].) Polynomials are expressions formed by addition, subtraction and multiplication but not division. Formally, a monomial in the parameters $a_1, \cdots, a_m$ is an expression of the form

$$a_1^{\alpha_1} a_2^{\alpha_2} \cdots a_m^{\alpha_m} \tag{8}$$

where the degrees $\alpha_1, \cdots, \alpha_m$ are all nonnegative. For instance, $a_1^2 a_3$ is a monomial but a_2^{-2} is not. The monomial (8) can be abbreviated to a^α. A polynomial in the parameters is then a finite sum of monomials with real coefficients:

$$\sum_\alpha c_\alpha a^\alpha \,,$$

with $c_\alpha \in \mathbb{R}$. The collection of all polynomials in the parameters, denoted $\mathbb{R}[a_1, \cdots, a_m]$, will usually be abbreviated to $\mathbb{R}[a]$. It is a commutative ring under the operations of polynomial addition and multiplication. Finally, a rational function in the parameters is any expression formed by addition, subtraction, multiplication and division. The collection of all rational functions in the parameters will be denoted $\mathbb{R}(a)$. (We used $\mathbb{R}(\vec{a})$ in the paper but we drop this vector notation here to reduce symbol cluttering.) Note that any $u \in \mathbb{R}(a)$ can always be expressed as a fraction in the form $u = p/q$ where $p, q \in \mathbb{R}[a]$; p is the numerator of u, while q is its denominator. For instance, $K_i^X \in \mathbb{R}(a)$; as expressed in (7), its denominator is a_i^X.

$\mathbb{R}(a)$ is a field under the operations of addition and multiplication of fractions. For many purposes it is as good a field as that of the real numbers, $\mathbb{R}$. Working over $\mathbb{R}(a)$ allows the parameters to be treated as uninterpreted symbols, enabling us to do calculations without knowing the parameter values in advance.

1.4 The phospho-form matrix Q

Equation (6) shows that the enzyme-substrate complexes are determined by the phospho-forms and the free enzymes. It is now possible to substitute for XS_i in terms of X and S_i in equation (1) to obtain at steady state

$$-\left\{\left(\frac{\sum_{i \xrightarrow{E} j} c_{i,j}^E}{K_i^E}\right) E + \left(\frac{\sum_{i \xrightarrow{F} j} c_{i,j}^F}{K_i^F}\right) F\right\} S_i + \left(\sum_{j \xrightarrow{E} i} \frac{c_{j,i}^E}{K_j^E}\right) E.S_j + \left(\sum_{l \xrightarrow{F} i} \frac{c_{l,i}^F}{K_l^F}\right) F.S_l = 0\,. \tag{9}$$

There are 2^n equations summarised here, with $0 \leq i \leq N$. Note that the parameters appear only in the form of catalytic efficiencies: ratios of catalytic constants to generalised Michaelis-Menten constants. Let $\kappa_1, \cdots, \kappa_{m'}$ be all the catalytic efficiencies that appear in (9). It will be convenient for technical reasons to sometimes work over the polynomial ring $\mathbb{R}[\kappa_1, \cdots, \kappa_{m'}] = \mathbb{R}[\kappa]$ and the corresponding field of rational functions, $\mathbb{R}(\kappa)$, in preference to $\mathbb{R}[a]$ and $\mathbb{R}(a)$.

At steady state, $F \neq 0$, for otherwise equation (18) shows that $F_{tot} = 0$, which we may reasonably assume not to be the case. Hence, we may multiply equation (9) through by $1/F$ to rewrite it in the form

$$\sum_{j=0}^{N} p_{i,j}(t) S_j = 0$$

where $p_{i,j}(t)$ is a polynomial in $t = E/F$ (in fact, at most a linear polynomial) whose coefficients are polynomial functions in the catalytic efficiencies. In other words, $p_{i,j}(t) \in \mathbb{R}[\kappa, t]$. Let Q be the $(N+1) \times (N+1)$ matrix (ie: the $2^n \times 2^n$ matrix) with entries in $\mathbb{R}[\kappa, t]$, given by $Q_{i,j} = p_{i,j}(t)$. If σ denotes the row vector of phospho-forms at steady state, $\sigma = (S_0, \cdots, S_N)$, then, evidently, $Q.\sigma^T = 0$. This is the basic equation satisfied by the steady state phosphoforms. Note that the nonlinear equations (1) have become linearised by working over $\mathbb{R}[\kappa, t]$.

1.5 S-positivity

As is evident from (9), the entries in Q have distinctive signs, but this needs to be appropriately formalised. A polynomial $p \in \mathbb{R}[a]$ (or any other polynomial ring like $\mathbb{R}[\kappa]$) is said to be s-positive ("sum positive") if it is a sum of positive monomials: $p = \sum_\alpha c_\alpha a^\alpha$ where $p \neq 0$ and $c_\alpha > 0$ whenever $c_\alpha \neq 0$. It is s-negative if $p = -q$ where q is s-positive. Although there is no corresponding order relation, and we use a different name to avoid drawing unwarranted conclusions, s-positivity shares many properties in common with positivity of numbers. For instance, the sum and product of s-positive polynomials are both s-positive. S-positivity can also be seen as a generalisation of positivity, which corresponds to the special case where there are no polynomial variables. Note that it follows from the form of the catalytic efficiencies appearing in (9) that if $p \in \mathbb{R}[\kappa]$ is s-positive, then it is also s-positive when rewritten as an element of $\mathbb{R}[a]$.

Considered as a matrix over $\mathbb{R}[\kappa, t]$, the diagonal terms of Q are all s-negative and if an off-diagonal term of Q is non-zero, then it is s-positive. In other words, if $-$ denotes an s-negative expression and $*$ denotes an expression which is either zero or s-positive, then Q looks like

$$\begin{pmatrix} - & * & \cdots & * & * \\ * & - & \cdots & * & * \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ * & * & \cdots & - & * \\ * & * & \cdots & * & - \end{pmatrix} \tag{10}$$

It is convenient to extend s-positivity from polynomials to rational functions in the obvious way. A rational function $f \in \mathbb{R}(a)$ is said to be s-positive if f can be expressed as a fraction $f = p/q$ where $p, q, \in \mathbb{R}[a]$ are both s-positive. As before, f is s-negative if $f = -g$ where g is s-positive.

1.6 Rationality of phospho-forms

The rationality of the steady state phospho-forms, together with an algorithm for determining the rational functions, begins with the following observation.

Lemma 1 If $1 = (1, \cdots, 1)$ is the all-ones row vector, then $1.Q = 0$ and Q has rank N (ie: one less than full rank) over $\mathbb{R}(\kappa, t)$.

Recall that the process of Gaussian elimination uses a sequence of elementary row operations to transform a $r \times r$ matrix, M, into upper-triangular block form. The k-th stage of this, denoted $M(k)$, brings M into the block form

$$\begin{pmatrix} A & B \\ 0 & D \end{pmatrix}$$

where A is a $k \times k$ upper triangular matrix whose diagonal elements are either 1 or 0 and the lower left block is a $(r-k) \times k$ zero matrix. If this process is undertaken on Q over the field $\mathbb{R}(\kappa, t)$ then it can be shown using Lemma 1 and Lemma 3 below that none of the pivots on the diagonal are zero until $Q(N+1)$ is reached. It follows that

$$Q(N+1) = \begin{pmatrix} 1 & 0 & \cdots & 0 & -\rho_0(t) \\ 0 & 1 & \cdots & 0 & -\rho_1(t) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -\rho_{N-1}(t) \\ 0 & 0 & \cdots & 0 & 0 \end{pmatrix} \tag{11}$$

where $\rho_i(t) \in \mathbb{R}(\kappa, t)$. Since $Q(N+1).\sigma^T = 0$, it follows that $S_i = S_N.\rho_i(t)$, giving a rational parameterisation in terms of S_N. By setting $r_i(t) = \rho_i(t)/\rho_0(t)$ (and taking $\rho_N(t) = 1$), this may be rewritten as

$$S_i = S_0.r_i(t) \tag{12}$$

for $0 \leq i \leq N$. This gives Paper Equation 1.

Gaussian elimination over symbolic fields like $\mathbb{R}(\kappa, t)$ is implemented in many computer algebra systems, including Mathematica, so that explicit parameterisations can be calculated for a given system.

1.7 Examples

We discuss three examples, of which the first two are used for calculations in the paper and the third is for purposes of illustration. None of the examples make use of processive phosphorylation or dephosphorylation, so it will be convenient to use the abbreviation $\kappa^X_{i,j}$ for the catalytic efficiency associated to a reaction $i \xrightarrow{X} j$:

$$\kappa^X_{i,j} = \frac{c^X_{i,j}}{K^X_i} \,.$$

Example 1. Consider a distributive, sequential system with n sites. In such a system the kinase phosphorylates distributively in a strict site order and the phosphatase dephosphorylates distributively in the reverse order. For a substrate with n sites, only $n+1$ phospho-forms appear (so that $N = n$). The phospho-forms must also be re-indexed: let S_i denote the phospho-form with the first i sites phosphorylated in order. The maximally phosphorylated phospho-form is then S_n. The network of reactions is given by

$$0 \xrightarrow{E} 1, \quad \cdots, \quad (n-1) \xrightarrow{E} n, \;\; 1 \xrightarrow{F} 0, \quad \cdots, \quad n \xrightarrow{F} (n-1)\,.$$

Paper Figure 3a depicts such an example with $n = 4$.

It will be convenient to introduce a further abbreviation for the ratio of the catalytic efficiency of the kinase to that of the phosphatase. For $0 \leq i < n$, let

$$\lambda_i = \frac{\kappa^E_{i,i+1}}{\kappa^F_{i+1,i}}\,. \tag{13}$$

The parameterisation for such a sequential system can be calculated by hand, as previously [20]:

$$r_i(t) = \left(\prod_{j=0}^{i-1} \lambda_j\right) t^i\,. \tag{14}$$

We see that the parameterisation is polynomial with coefficients in $\mathbb{R}(\kappa)$.

Example 2. Now consider a nonsequential system with n sites with only distributive phosphorylation and dephosphorylation and let us revert to the indexing scheme in §1.1. Paper Figure 1a and 1b depict such an example for $n = 3$. As before, $N = 2^n - 1$. Note that $i \xrightarrow{E} j$ if, and only if, $j \xrightarrow{F} i$. Suppose further that $\kappa^E_{i,j} = \kappa^F_{j,i}$. In this case, although the system is nonsequential with all 2^n phospho-forms, it behaves mathematically as if it were sequential. For $0 \leq i \leq N$, let $b(i)$ denote the size of the set $[i]$, or the number of bits set to 1 in the binary expansion of i. Let σ be the row vector such that $\sigma_i = t^{b(i)}$. It is not difficult to show that $Q.\sigma^T = 0$. This is quite surprising as, despite the simplifying assumption, the entries of Q are still functions of the $\kappa^X_{i,j}$. It follows immediately that

$$r_i(t) = t^{b(i)} \tag{15}$$

We see that the parameterisation is also polynomial in this case.

Example 3. The simple polynomial parameterisations in (14) and (15) are quite special. Consider a nonsequential system with $n = 2$ sites with only distributive phosphorylation and dephosphorylation. This example is treated in more detail in [27]. The network of reactions is given by

$$0 \xrightarrow{E} 1,\;\; 0 \xrightarrow{E} 2,\;\; 1 \xrightarrow{E} 3,\;\; 2 \xrightarrow{E} 3,\;\; 1 \xrightarrow{F} 0,\;\; 2 \xrightarrow{F} 0,\;\; 3 \xrightarrow{F} 2,\;\; 3 \xrightarrow{F} 1\,.$$

The rational functions $r_i(t)$ can be calculated in Mathematica:

$$\begin{aligned} r_1(t) &= \frac{t(\kappa^F_{3,1} + \kappa^F_{3,2})\kappa^E_{0,1}\kappa^F_{2,0} + t^2(\kappa^E_{0,1} + \kappa^E_{0,2})\kappa^E_{2,3}\kappa^F_{3,1}}{(\kappa^F_{3,1} + \kappa^F_{3,2})\kappa^F_{1,0}\kappa^F_{2,0} + t(\kappa^E_{2,3}\kappa^F_{1,0}\kappa^F_{3,1} + \kappa^E_{1,3}\kappa^F_{2,0}\kappa^F_{3,2})} \\ r_2(t) &= \frac{t(\kappa^F_{3,1} + \kappa^F_{3,2})\kappa^E_{0,2}\kappa^F_{1,0} + t^2(\kappa^E_{0,1} + \kappa^E_{0,2})\kappa^E_{1,3}\kappa^F_{3,2}}{(\kappa^F_{3,1} + \kappa^F_{3,2})\kappa^F_{1,0}\kappa^F_{2,0} + t(\kappa^E_{2,3}\kappa^F_{1,0}\kappa^F_{3,1} + \kappa^E_{1,3}\kappa^F_{2,0}\kappa^F_{3,2})} \qquad (16) \\ r_3(t) &= \frac{t^2(\kappa^E_{0,2}\kappa^E_{2,3}\kappa^F_{1,0} + \kappa^E_{0,1}\kappa^E_{1,3}\kappa^F_{2,0}) + t^3(\kappa^E_{0,1} + \kappa^E_{0,2})\kappa^E_{1,3}\kappa^E_{2,3}}{(\kappa^F_{3,1} + \kappa^F_{3,2})\kappa^F_{1,0}\kappa^F_{2,0} + t(\kappa^E_{2,3}\kappa^F_{1,0}\kappa^F_{3,1} + \kappa^E_{1,3}\kappa^F_{2,0}\kappa^F_{3,2})} \,. \end{aligned}$$

Note the substantial increase in complexity over Examples 1 and 2. All three examples exhibit certain features which turn out to be general properties. Most significantly, the $r_i(t)$ are all s-positive elements of $\mathbb{R}(\kappa, t)$.

1.8 Positivity of $r_i(t)$

The positivity property states that, for $0 \leq i < N$, $\rho_i(t)$—and hence also $r_i(t)$—is s-positive as a rational function in $\mathbb{R}(\kappa, t)$. From what was said previously, $r_i(t)$ is necessarily also s-positive as a rational function in $\mathbb{R}(a, t)$. It follows that, for any positive parameter values and positive values of t, $r_i(t)$ is well defined and positive, as claimed in the paper. The proof requires two general matrix lemmas.

If M is an $r \times r$ matrix over a field $\mathbb{K}$ recall that $M(i_1, \cdots, i_u; j_1, \cdots, j_v)$ denotes the $u \times v$ submatrix consisting of the elements in rows $i_1, \cdots, i_u$ and columns $j_1, \cdots, j_v$. Note that the row and column sequences do not have to retain their original order. It is convenient to use a shorthand for certain sequences, in which $i_1, \cdots, \hat{i_t}, \cdots, i_u, k$ denotes the sequence in which k appears in place of the hatted index, i_t: $i_1, \cdots, \hat{i_t}, \cdots, i_u, k = i_1, \cdots, i_{t-1}, k, i_{t+1}, \cdots, i_u$.

Lemma 2 During Gaussian elimination of M, suppose that none of the pivots on the diagonal are zero up to and including stage k. Then $\det M(1, \cdots, k; 1, \cdots, k) \neq 0$ and the entries of $M(k)$ are given by

$$M(k)_{i,j} = \begin{cases} \dfrac{\det M(1, \cdots, k; 1, \cdots, \hat{i}, \cdots, k, j)}{\det M(1, \cdots, k; 1, \cdots, k)} & \text{if } 1 \leq i \leq k \\ \dfrac{\det M(1, \cdots, k, i; 1, \cdots, k, j)}{\det M(1, \cdots, k; 1, \cdots, k)} & \text{if } k+1 \leq i \leq r. \end{cases}$$

This can be proved using standard results in matrix theory. Lemma 1 allows the rational functions in (11) to be calculated as

$$\rho_{i-1}(t) = \frac{\det Q(1, \cdots, N; 1, \cdots, \hat{i}, \cdots, N, N+1)}{\det Q(1, \cdots, N; 1, \cdots, N)} \,, \qquad (17)$$

for $1 \leq i \leq N$. The determinants appearing in (17) are elements of $\mathbb{R}[\kappa, t]$, so that (17) gives an expression for $\rho_i(t) \in \mathbb{R}(\kappa, t)$ as a fraction.

Let M be a $r \times r$ matrix with entries in a polynomial ring like $\mathbb{R}[\kappa, t]$. We say that M satisfies property S if it has the same pattern as Q in (10), with diagonal elements s-negative and non-zero off-diagonal elements s-positive, and, in addition, each column sum, $\sum_j M_{i,j}$, is also s-negative.

Lemma 3 Suppose that $M(1, \cdots, k; 1, \cdots k)$ satisfies property S for $1 \leq k \leq r$. Then, $\det M$ is s-positive (respectively, s-negative) according as r is even (respectively, odd).

This is straightforward to prove by induction in the special case that the entries of M are real numbers. A similar argument can be adapted to the more general case of entries in $\mathbb{R}[\kappa, t]$. Lemma 3 allows the positivity of the determinants in (17) to be determined. The positivity property of $r_i(t)$ follows immediately.

1.9 The broader perspective

It is helpful to set these results in the context of related theoretical work. We summarise some of the relevant literature here. Chemical Reaction Network Theory (CRNT) was initiated by Horn and Jackson [24] and developed in great depth by Feinberg and his students [17]. In its early manifestations, CRNT introduced the *deficiency* of a reaction network and showed that networks with low deficiency could be very complex but were, nevertheless, monostable irrespective of parameter values (for detailed references, see [17, 19]). Recent results have uncovered more nuanced graph-theoretic conditions which confirm the existence of a steady state, or rule out the possibility of multiple steady states [12, 13, 14]. The deficiency of the multisite phosphorylation network discussed here increases with increasing n and CRNT is not obviously applicable to our results. Adelman and his students have initiated an axiomatic approach to mass-action kinetics [1], generalising Horn and Jackson's early work but with a continued focus on monostability. Sontag has introduced methods of monotone dynamical systems by which high-dimensional models can sometimes be reduced to two dimensions [4]. However, these methods rely on Michaelis-Menten approximations and the underlying mass-action systems, as studied here, are not monotone.

The dynamical systems arising from mass-action kinetics possess the special property of being polynomial: their right-hand sides are polynomial functions of the state variables. Polynomial dynamical systems have a long history in mathematics going back to Hilbert's sixteenth problem [8]. Of particular interest here is that their steady states form *algebraic varieties*, which are the subject matter of algebraic geometry [10]. Equation (12) shows that the steady-state phospho-forms form a (projective) rational curve [35]. Rational varieties are special [33] and their appearance here is of considerable mathematical interest. Rationality provides an explicit way to compute points on a variety, in contrast to their implicit definition as solutions of polynomial equations. The geometric implications of this are explored in an earlier paper of ours for $n = 2$ sites [27].

Although mass-action kinetics have long been known to give rise to polynomial dynamics and steady-state varieties, algebraic geometry has not been widely exploited in biological modelling. Computational algebra and Gröbner bases have been previously used to calculate rate functions in Metabolic Control Analysis [5, 6] and the use of Gröbner bases in the context of reaction networks has been reviewed in [28]. Algebraic geometry over finite fields has been used for gene network reconstruction from microarray time-series data [2, 25]. A particularly interesting development has been the use of toric varieties to reinterpret certain results on monostability in CRNT [11, 18]. Parrilo, Doyle and others have exploited semi-algebraic geometry to develop "sum-of-squares" techniques and associated software tools for stability determination and model validation [16, 31]; see also [32]. The collection of papers in [21] exhibits the broad scope of these control-theoretic methods. While our results have exploited the special structure of PTM systems and use only linear methods (albeit over extension fields), pushing beyond this to more complex regulatory mechanisms will require more powerful methods from algebraic geometry and related mathematical disciplines.

Although unrelated to biological modelling, there has been much interest in using algebraic geometric methods to analyze discrete statistical models, particularly in phylogeny reconstruction [30]. This has given rise to the new field of algebraic statistics [29].

2 Exact calculation of steady states using Φ

2.1 Definition of Φ

It follows from (6) and (12) that all 3.2^n state variables in the system are determined by just S_0, E and F. We can now calculate the total amounts of substrate, S_{tot}, and enzymes, E_{tot}, F_{tot} in terms of S_0, E and F. Define the following three rational functions in $\mathbb{R}(a, t)$,

$$\phi_1(t) = \sum_{i=0}^{N} r_i(t)\,, \quad \phi_2(t) = \sum_{i=0}^{N-1} \frac{r_i(t)}{K_i^E}\,, \quad \phi_3(t) = \sum_{i=1}^{N} \frac{r_i(t)}{K_i^F}\,.$$

Note that ϕ_1, ϕ_2 and ϕ_3 are all s-positive. It follows from (6) that

$$S_0\phi_1(t) = \sum_{i=0}^{N} S_i\,, \quad E.S_0\phi_2(t) = \sum_{i=0}^{N-1} ES_i\,, \quad F.S_0\phi_3(t) = \sum_{i=1}^{N} FS_i\,.$$

The total substrate is therefore given by $S_{tot} = S_0(\phi_1 + E\phi_2 + F\phi_3)$. We can hence define a 2×2 function, Φ, such that

$$\begin{array}{rcccl} \Phi_1(E,F) & = & E\left(1 + \dfrac{S_{tot}\phi_2}{\phi_1 + E\phi_2 + F\phi_3}\right) & = & E_{tot} \\ \Phi_2(E,F) & = & F\left(1 + \dfrac{S_{tot}\phi_3}{\phi_1 + E\phi_2 + F\phi_3}\right) & = & F_{tot}\,. \end{array} \tag{18}$$

This gives two algebraic equations for E, F, as required for Paper Equation 2. We have shown that the E, F values in any steady state are solutions of (18). Now suppose that (E, F) is any solution of (18). Let $S_0 = S_{tot}/(\phi_1 + E\phi_2 + F\phi_3)$, $S_i = S_0 r_i(E/F)$ and $XS_i = X.S_i/K_i^X$ for $X = E, F$. It is easy to check that this defines a steady state. Hence, (18) characterises the steady states.

2.2 Numerical solution of Φ

If parameter values are specified and enzyme and substrate totals are chosen, equation (18) can be solved numerically in Matlab and the steady states determined as in Paper Figure 2. The parameter values for the sequential example used in Paper Figure 2 are shown in Table 1. We do the Matlab calculation as follows. (Further information on Matlab is available at www.mathworks.com.) We first calculate the values of Φ on a grid in the (E, F) plane and use `contourc` on the output of this to determine the sets of points satisfying $\Phi_1(E, F) = E_{tot}$ (the E_{tot} curve) and $\Phi_2(E, F) = F_{tot}$ (the F_{tot} curve). `Contourc` interpolates to find these "isolines". They provide the visual plots in which the steady states appear at the intersections of the curves, as in Paper Figure 2a. For automated searches we use a 120×120 grid, where $\log_{10}$ of each coordinate is equally spaced in $[-6, 6]$. For manual inspection at finer resolution we use a 1200×1200 grid. We then calculate the steady states via `fsolve`, which uses an iterative nonlinear search starting from a specified initial condition. We separately calculate the derivative of Φ (the Jacobian) and provide that to `fsolve` to speed up the search.

An appropriate choice of initial conditions is essential for both speed and accuracy. We found that points lying on either the E_{tot} curve or the F_{tot} curve provided good initial conditions, while other points sometimes caused `fsolve` to diverge or return an error. We used the E_{tot} curve for the set of initial conditions. We first chose three points on the E_{tot} curve, one each at either extreme of E/F value and the third in the middle. If, for each of these initial conditions, `fsolve` returns a solution and the solutions agree to within a specified tolerance (usually 10^{-4}) in each coordinate, we return that solution as the unique steady state of the system. If any of these conditions fails, we take every other point lying on the E_{tot} curve and run `fsolve` on all of them. We count the resulting solutions as distinct if they differ by more than the tolerance in any coordinate. The distinct solutions are returned as the steady states. This protocol was fine-tuned from numerical experiments to provide a reasonable balance between speed and accuracy, using the visual plot and the numerical calculation to cross-check each other. It can take up to 30 seconds to find all the steady states for a system with 4 sites but the improvement over numerical simulation of the differential equations is substantial.

2.3 Stability of steady states

Given a system of ordinary differential equations, $dx/dt = f(x)$, where $x \in \mathbb{R}^m$ and $f : \mathbb{R}^m \to \mathbb{R}^m$, the Jacobian matrix, J, is the $m \times m$ matrix defined by $J_{ij} = \partial f_i / \partial x_j$. According to standard theory, the stability of a steady state is determined by the eigenvalues of the Jacobian evaluated at the state [23]. If all the eigenvalues have negative real part, the state is stable; if not, it is unstable. We computed the Jacobian symbolically in terms of the parameters $a_i^X, b_i^X, c_{i,j}^X$ and the steady-state species concentrations. For a given steady state defined by S_{tot}, E, F, we computed all the steady-state species concentrations using (6) and (12) and substituted these values into the symbolic Jacobian along with the parameter values. We then calculated the eigenvalues using Matlab's `eig` function.

In tests of stability we found that if the steady states are ordered by increasing $t = E/F$, unstable states occur between stable ones, just as a hill is found between two valleys. Hence, if there are $2j + 1$ steady states (we found only odd numbers of steady states in all examples studied), then there are $j + 1$ stable states.

3 Approximate calculation of steady states using $P(t)$

3.1 Definition of $P(t)$

If substrate is in excess over enzymes, so that $S_{tot} \gg E_{tot}$ and $S_{tot} \gg F_{tot}$, then the total amounts of enzyme-substrate complexes may be considered negligible in comparison to S_{tot}. Hence, we may write, approximately,

$$S_{tot} = [S_0] + \cdots + [S_N] = [S_0]\phi_1(t)\,. \tag{19}$$

We can then rewrite the expressions for E_{tot} and F_{tot} in (18) to get

$$\begin{aligned} E_{tot} &= E\left(1 + S_{tot}\frac{\phi_2(t)}{\phi_1(t)}\right) \\ F_{tot} &= F\left(1 + S_{tot}\frac{\phi_3(t)}{\phi_1(t)}\right)\,. \end{aligned} \tag{20}$$

For given, E_{tot}, F_{tot} and S_{tot}, the (E, F) pairs which are solutions of (20) are the steady states of the system, to within the approximation..

Dividing the first equation by the second, and setting $E_{tot}/F_{tot} = w$, we see that

$$w(\phi_1(t) + S_{tot}\phi_3(t)) = t(\phi_1(t) + S_{tot}\phi_2(t))\,, \tag{21}$$

and so, rearranging this,

$$R(t) = (t - w)\phi_1(t) + S_{tot}(t\phi_2(t) - w\phi_3(t)) = 0\,. \tag{22}$$

Since $\phi_1(t), \phi_2(t)$ and $\phi_3(t)$ are rational functions in $\mathbb{R}(a, t)$, so too is $R(t)$. Furthermore, since the ϕ's are all s-positive, it is evident that $R(t)$ can be written in the form $R(t) = P(t)/Q(t)$ where $P(t), Q(t)$ are polynomials in $\mathbb{R}[a, t]$ and the denominator, $Q(t)$, is s-positive. (Of course, this need not be the case for the numerator, $P(t)$.) Accordingly, for positive parameter values and positive values of t, $R(t) = 0$ if, and only if, $P(t) = 0$. Note the crucial role played by the positivity property at this point.

If (E, F) is a solution of the approximate equation (20) then $t = E/F$ is a solution of $P(t) = 0$. Now suppose $t_1 > 0$ and $P(t_1) = 0$, so that $R(t_1) = 0$. It follows from (22) that

$$t_1 = w\left(\frac{\phi_1(t_1) + S_{tot}\phi_3(t_1)}{\phi_1(t_1) + S_{tot}\phi_2(t_1)}\right) .$$

Let

$$\begin{aligned} E_2 &= \frac{E_{tot}\phi_1(t_1)}{\phi_1(t_1) + S_{tot}\phi_2(t_1)} \\ F_2 &= \frac{F_{tot}\phi_1(t_1)}{\phi_1(t_1) + S_{tot}\phi_3(t_1)} . \end{aligned} \tag{23}$$

Since $t_1 > 0$, and ϕ_1, ϕ_2 and ϕ_3 are s-positive, this can always be done so that E_2 and F_2 are positive. Let $t_2 = E_2/F_2$. Dividing the equations for E_2 and F_2 in (23), we see that

$$t_2 = w\left(\frac{\phi_1(t_1) + S_{tot}\phi_3(t_1)}{\phi_1(t_1) + S_{tot}\phi_2(t_1)}\right) = t_1 .$$

It follows from (23) that t_1 satisfies (20) and is therefore a solution of the approximate system. Hence, solutions of the approximate system (20) correspond precisely to positive solutions of $P(t) = 0$.

3.2 Numerical solution of $P(t) = 0$

We used Matlab's `roots` function, which is fast and accurate. For n up to 12 sites, ~ 6000 polynomials per second can be solved, giving a substantial improvement over numerical solution of equation (18).

3.3 Approximation of $\Phi(E, F)$ by $P(t)$

To assess quantitatively how close $P(t) = 0$ is to the exact steady state solution provided by equation (18), we proceeded as follows for the $n = 4$ distributive, sequential system studied in the paper. Using the notation from Example 1 in §1.7, we chose K_i^E and K_i^F randomly from the uniform distribution on $[1, 1000]$ nM and $\log_{10}(\lambda_i)$ randomly from the uniform distribution on $[-3, 3]$. We set $E_{tot} = F_{tot}$ and chose $\log_{10} E_{tot}$ and $\log_{10} S_{tot}$ randomly from the uniform distribution on $[0, 4]$, corresponding to a concentration range of $[1 - 10000]$ nM. We generated 10,000 such systems, for which we solved both equation (18) and $P(t) = 0$ for the steady states. We found 108 systems for which the number of steady states differed between Φ and P. We first set those aside but analyze them further below. For the remaining systems, we calculated E/F for each steady state coming from Φ and listed them in order of increasing E/F: $s_1 < s_2 < \cdots < s_k$, where k is the number of steady states. (We found $k = 1$ and $k = 3$

only, with no $k = 5$.) We matched these with the ordered list of positive solutions of $P(t) = 0$, $a_1 < a_2 < \cdots < a_k$. We measured the discrepancy between the exact solution coming from (18) and the approximate solution coming from $P(t) = 0$ by calculating the average normalized difference,

$$\sigma = \frac{1}{k} \sum_{i=1}^{k} \frac{|s_i - a_i|}{s_i} \,. \tag{24}$$

Figure 1A shows that for nearly 80% of the randomly chosen systems, the approximation is good to within $\sigma < 0.1$, irrespective of the values of S_{tot} and E_{tot}. Figure 1B shows that the approximation gets steadily better as S_{tot}/E_{tot} increases from 1. We took $S_{tot}/E_{tot} \geq 5$ as our cut-off. In this range, $\sigma < 0.23$.

We then considered the 108 omitted systems for which Φ and $P(u)$ differed in the number of roots found. A histogram of these is plotted against $\log_{10} S_{tot}/E_{tot}$ on the bottom of Figure 1B. We found 52 miscounted systems for which $S_{tot}/E_{tot} \geq 5$. We examined each of these by hand and determined, on a conservative basis, that 45 of them were caused by numerical errors in finding solutions to Φ. That is, when these systems were re-computed with finer tolerances and a denser set of initial conditions, the number of steady states was found to converge and to agree with those obtained from solving $P(t) = 0$. The remaining 7 systems were adjudged to be possible errors arising from using $P(t) = 0$ as an approximation for equation (18). Since there were 3385 systems for which $S_{tot}/E_{tot} \geq 5$, this gives a miscounting rate of 0.2%.

3.4 Examples

Equation (22) allows $P(t)$ to be calculated once the rational parameterisation is known. For both Examples 1 and 2, the $r_i(t)$ are polynomials in $\mathbb{R}[\kappa, t]$, so that $R(t) = P(t)$, and $P(t)$ has degree $n + 1$. Suppose that $P(t) = \alpha_{n+1} t^{n+1} + \alpha_n t^n + \cdots + \alpha_1 t + \alpha_0$, where $\alpha_i \in \mathbb{R}(a)$.

Example 1. The sequential system. Recall the definition of λ_i in (13). In this case,

$$\begin{aligned} \alpha_{n+1} &= \lambda_0 \cdots \lambda_{n-1} \quad \text{and} \quad \alpha_0 = -w \\ \alpha_k &= \lambda_0 \cdots \lambda_{k-2} \left[(1 - \lambda_{k-1} w) + S_{tot} \left(\frac{1}{K^E_{k-1}} - \frac{\lambda_{k-1} w}{K^F_k} \right) \right] \\ & \quad \text{for } 0 < k \leq n. \end{aligned} \tag{25}$$

Example 2. The nonsequential system in which $\kappa^E_{i,j} = \kappa^F_{j,i}$. Recall that $b(i)$ is the number of bits set to 1 in the binary expansion of i. In this case,

$$\begin{aligned} \alpha_{n+1} &= 1 \quad \text{and} \quad \alpha_0 = -w \\ \alpha_k &= \binom{n}{k-1} - w \binom{n}{k} + S_{tot} \left[\left(\sum_{b(i)=k-1} \frac{1}{K^E_i} \right) - w \left(\sum_{b(i)=k} \frac{1}{K^F_i} \right) \right] \\ & \quad \text{for } 0 < k \leq n. \end{aligned} \tag{26}$$

3.5 Finding systems with specified steady states

Suppose given a polynomial of degree $n+1$, $Q(t) = A_{n+1}t^{n+1} + A_n t^n + \cdots + A_1 t + A_0$, with $A_i \in \mathbb{R}$. We claim that, under suitable conditions on the A_i, we can construct a multisite phosphorylation system for which $P(t) = Q(t)$. By choosing an appropriate Q, as shown below, the existence of multiple steady states can be demonstrated. This can be done for both the sequential system given by equation (25) and the nonsequential system given by (26). The latter example shows that the existence of multiple steady states is not a special property of sequential systems.

Example 1. Suppose that $Q(t)$ satisfies $A_{n+1} > 0$ and $A_0 < 0$. We claim that for $S_{tot} > 0$ chosen arbitrarily and appropriate choices of w, K_i^X and λ_i, all positive, the corresponding $P(t)$ polynomial defined by (25) coincides with $Q(t)$.

Note first that the term in square brackets in (25) can be rewritten as

$$\left(1 + \frac{S_{tot}}{K_i^E}\right) - \lambda_i w \left(1 + \frac{S_{tot}}{K_{i+1}^F}\right) .$$

Start by choosing $S_{tot} > 0$ arbitrarily. Choose $w = -A_0 > 0$. For $0 \le i \le n-2$, choose K_i^E, K_{i+1}^F and λ_i inductively so that

$$\left(1 + \frac{S_{tot}}{K_i^E}\right) - \lambda_i w \left(1 + \frac{S_{tot}}{K_{i+1}^F}\right) = \frac{A_{i+1}}{\lambda_0 \cdots \lambda_{i-1}} = B_{i+1}$$

as follows. (When $i = 0$, the induction starts with $A_1 = B_1$ but the argument below is identical.) If $B_{i+1} = 0$, take $\lambda_i = 1/w$ and choose $K_i^E = K_{i+1}^F > 0$ arbitrarily. If $B_{i+1} > 0$, choose $K_i^E > 0$ so that

$$\left(1 + \frac{S_{tot}}{K_i^E}\right) > B_{i+1} ,$$

which may always be done. Now choose K_{i+1}^F and λ_i so that

$$\left(1 + \frac{S_{tot}}{K_i^E}\right) - B_{i+1} = \lambda_i w \left(1 + \frac{S_{tot}}{K_{i+1}^F}\right) , \tag{27}$$

which may also always be done. If $B_{i+1} < 0$ then K_{i+1}^E may be chosen arbitrarily and the left hand side of (27) will always be positive. Hence, K_{i+1}^F and λ_i can always be chosen positive so that (27) is satisfied.

By following this inductive procedure for $0 \le i \le n-2$ we will have chosen S_{tot}, w, K_i^E for $0 \le i \le n-2$, K_i^F for $1 \le i \le n-1$ and λ_i for $0 \le i \le n-2$ all positive. With these choices we will have satisfied (25) for all coefficients A_i such that $0 \le i < n$. Now consider the last two coefficients A_n and A_{n+1}. Choose $\lambda_{n-1} = A_{n+1}/(\lambda_0 \cdots \lambda_{n-2}) > 0$, so that (25) is satisfied for A_{n+1}. Now choose K_{n-1}^E and K_n^F such that

$$\frac{1}{K_{n-1}^E} - \frac{\lambda_{n-1} w}{K_n^F} = \frac{1}{S_{tot}} \left(\frac{A_n}{\lambda_0 \cdots \lambda_{n-2}} - (1 - \lambda_{n-1} w) \right) = c ,$$

as follows. The right hand side consists of terms like A_n, which are given, or terms that have been previously determined. Let $\alpha = \lambda_{n-1} w > 0$. We have to find $x, y > 0$ such that

$$x - \alpha y = c\,.$$

Since $\alpha > 0$, this can always be done for any c, thereby satisfying (25) for A_n. This completes the proof.

Example 2. Suppose that $Q(t)$ satisfies $A_{n+1} = 1$ and $A_0 < 0$. We claim that for $S_{tot} > 0$ chosen arbitrarily and appropriate choices of w and K_i^X, all positive, the corresponding $P(t)$ polynomial defined by (26) coincides with $Q(t)$. The argument is simpler than in the sequential case and requires no induction.

Choose $S_{tot} > 0$ arbitrarily and choose $w = -A_0 > 0$. Now choose $1 \le k \le n$. We need to show, rearranging the formula for α_k in (26), that

$$\sum_{b(i)=k-1} \left(1 + \frac{S_{tot}}{K_i^E}\right) - w \left[\sum_{b(i)=k} \left(1 + \frac{S_{tot}}{K_i^F}\right)\right] = A_k\,, \tag{28}$$

for suitable choice of K_i^X. Consider the function of several variables

$$\mu(u_1, \cdots, u_m) = \sum_{i=1}^{m} \left(1 + \frac{a}{u_i}\right)\,,$$

where $a > 0$. It is easy to see that μ maps the positive orthant of $\mathbb{R}^m$ onto the open interval (m, ∞). In other words, for any $x > m$ we can find $u_1, \cdots, u_m$, all positive, such that $\mu(u_1, \cdots, u_m) = x$. Applying this, we can choose K_i^F with $b(i) = k$, all positive, such that,

$$A_k + w \left[\sum_{b(i)=k} \left(1 + \frac{S_{tot}}{K_i^F}\right)\right] > \binom{n}{k-1}\,.$$

We can do this no matter what the value of A_k. Applying the μ property again, we can choose K_i^E with $b(i) = k-1$, all positive, such that (28) holds. The completes the proof.

We can now construct steady states. If n is odd, and $t_1, \cdots, t_n$ are any n distinct positive numbers, then $(t - t_1) \cdots (t - t_n)(t + 1)$ is a polynomial of degree $n + 1$ that satisfies the conditions required for both Example 1 and 2. Hence, we can find rate constants for either example for which the corresponding polynomial is $P(t)$ and the system has n steady states at $t = t_1, \cdots, t_n$. Similarly, if n is even and $t_1, \cdots, t_{n+1}$ are any $n + 1$ distinct positive numbers, then $(t - t_1) \cdots (t - t_{n+1})$ is a polynomial of degree $n + 1$ that satisfies the conditions for both Example 1 and 2. Hence, we can find corresponding systems with $n + 1$ steady states at $t = t_1, \cdots, t_{n+1}$. Since S_{tot} can be chosen arbitrarily in both examples, we can ensure that the solutions of $P(t) = 0$ that have been found approximate as closely as desired to solutions of the exact equations given by equation (18). Assuming, on the basis of the findings in §2.3, that unstable states appear between stable ones we find that, for any n, we can construct sequential and nonsequential multisite phosphorylation systems for which the number of stable states is $\lfloor (n+2)/2 \rfloor$, as claimed in the paper. Following on from our earlier demonstration of this for sequential systems [34], Wang and Sontag found rigorous bounds on the number of stable states [36].

4 Dynamical arguments

The discussion above has centered on algebraic arguments at steady state. To understand how steady states arise dynamically, other methods are needed.

4.1 Kinetic trapping

The kinetic trapping argument in Paper Figure 3 uses the $n = 4$ distributive, sequential system and applies the Michaelis-Menten rate formula. Recall [9] that this approximation is given by

$$\frac{dS_1}{dt} = \frac{c^E_{0,1} E_{tot} . S_0}{K^E_0 + S_0} \,,$$

for the production of S_1 from S_0 by E. A similar formula to this holds for the production of S_0 from S_1 by F. The shape of the corresponding hyperbolas in Paper Figure 3 can be summarised in two graphical quantities: the slope of the hyperbola at the origin, given by $c^E_{0,1} E_{tot}/K^E_0$, and the maximal rate or asymptotic value, given by $c^E_{0,1} E_{tot}$. There are four possible dispositions of the two curves, depending on the relative values of these two quantities, and the disposition shown in Paper Figure 3 requires that

$$\frac{c^F_{1,0} F_{tot}}{K^F_1} > \frac{c^E_{0,1} E_{tot}}{K^E_0} \quad \text{and} \quad c^F_{1,0} F_{tot} > c^E_{0,1} E_{tot} \,.$$

Recalling that $w = E_{tot}/F_{tot}$ and the definition of λ_i in (13), these conditions can be rewritten as

$$1 - \lambda_0 w > 0 \quad \text{and} \quad \frac{1}{K^E_0} - \frac{\lambda_0 w}{K^F_1} > 0 \,. \tag{29}$$

These are the same expressions that appear in (25), and it follows from (29) that $\alpha_1 > 0$. Applying the same reasoning to the kinetic trapping condition at the other end of the sequential chain, we find that $\alpha_n < 0$. The conditions that emerged from the dynamic argument of kinetic trapping turn out to be sign conditions on the coefficients of $P(t)$.

4.2 Alternating sign condition leads to increasing multistability

For high multistability, it is necessary for $P(t)$ to have many positive roots. Calculating the number of real roots of a real polynomial is generally intractable and only probabilistic answers have been found to this class of questions [7]. For instance, if the coefficients of $P(t)$ are chosen randomly from the standard normal distribution, then the average number of real roots (ie: without restriction on the sign) is given by the Kac integral formula, which is approximated by $2\log(n{+}1)/\pi$ [15]. This may suggest that high multistability, while mathematically possible, is quite rare. However, the coefficients of $P(t)$ are not independent and evolution may sculpt the site-specific parameters in ways not yet understood. This leaves open the possibility that some bias in the coefficients may enrich for multistability. The sign conditions on the coefficients of $P(t)$ found above suggests how this might work. Recall that Descartes' Rule of Signs [3] states that the number of positive solutions of $P(t) = 0$ is always less than the number of sign changes in the coefficients of $P(t)$. We exploited this as follows.

For the $n = 4$ distributive, sequential system with $K_i^E = K_{i+1}^F$, it follows from (25) that a_i is positive if $\lambda_i < 1/w$ and negative if $\lambda_i > 1/w$. Note that the example used in the paper, whose parameter values are given in Table 1, has coefficients with alternating signs for $w = 1$. For each even n from 2 to 12 we generated 100,000 systems as follows. We chose $\log_{10}(K_i^X$ in nM) randomly from the uniform distribution on $[-1, 2]$ and $\log_{10}\lambda_i$ randomly from the uniform distribution on $[-2, 2]$. We set $S_{tot} = 1000\,\text{nM}$, forcing the enzymes into saturation, and $E_{tot} = F_{tot} = 200\,\text{nM}$, ensuring that substrate was in excess and that $w = 1$. We found the distribution of steady states in Figure 2A where monostability remains more likely than multistability up to $n = 12$ and five steady states do not appear until $n = 6$. We then repeated the calculation with $\log_{10}\lambda_i$ uniform on $[-2, 0]$ for i even and on $[0, 2]$ for i odd, thereby forcing the maximum number of sign changes among the coefficients of $P(t)$. The distribution shifted to that in Figure 2B in which multistability is now more likely than monostability as soon as $n > 2$, the frequency of five steady states is increased and becomes detectable for $n = 4$.

4.3 Simulation of the differential equations

We used the `little b` computational infrastructure to generate the differential equations from which the dynamical information in Paper Figures 2a and 4 was obtained [26]. `Little b` is a modular programming language in which models can be specified at a biological level of description and compiled into Matlab code, which can then be simulated. Programmable modularity allows us to write a generic multisite phosphorylation module which can construct the required differential equations for a multisite phosphorylation system having any number of sites, making it unnecessary to write new Matlab code each time. See [26, §3.2] for more details of the generic module. `Little b` is freely available as open source software from littleb.org and vcp.med.harvard.edu. For the simulations we used Matlab's `ode15s` solver with absolute tolerance of 10^{-35}.

References

[1] L. Adleman, M. Gopalkrishnan, M.-D. Huang, P. Moisset, and D. Reishus. On the mathematics of the law of mass-action. `arxiv.org/abs/0710.3421`, 2008.

[2] E. E. Allen, J. S. Fetrow, L. W. Daniel, S. J. Thomas, and D. J. John. Algebraic dependency models of protein signal transduction networks from time-series data. *J. Theor. Biol.*, 238:317–30, 2006.

[3] B. Anderson, J. Jackson, and M. Sitharam. Descartes' Rule of Signs revisited. *Amer. Math. Monthly*, 105:447–451, 1998.

[4] D. Angeli, J. E. Ferrell, and E. D. Sontag. Detection of multistability, bifurcations, and hysteresis in a large class of biological positive-feedback systems. *Proc. Natl. Acad. Sci. USA*, 101:1822–7, 2004.

[5] M. Bayram, J. P. Bennett, and M. C. Dewar. Using computer algebra to determine rate constants in biochemistry. *Acta. Biotheor.*, 41:53–62, 1993.

[6] J.P. Bennett, J. H. Davenport, M. C. Dewar, D. L. Fisher, M. Grinfeld, and H. M. Sauro. Computer algebra approaches to enzyme kinetics. In G. Jacob and F. Lamnabhi-Lagarrigue, editors, *Algebraic Computing in Control*, pages 23–30. Springer-Verlag, Berlin, 1991.

[7] A. T. Bharucha-Reid and M. Sambandham. *Random Polynomials*. Probability and Mathematical Statistics. Academic Press, Burlington, MA, USA, 1986.

[8] C. J. Christopher and N. G. Lloyd. Polynomial systems: a lower bound for the Hilbert numbers. *Proc. Roy. Soc. Lond. A*, 450:219–24, 1995.

[9] A. Cornish-Bowden. *Fundamentals of Enzyme Kinetics*. Portland Press, London, UK, 2nd edition, 1995.

[10] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties and Algorithms*. Springer, 2nd edition, 1997.

[11] G. Craciun, A. Dickenstein, A. Shiu, and B. Sturmfels. Toric dynamical systems. J. Symb. Comp., to appear, 2008.

[12] G. Craciun and M. Feinberg. Multiple equilibria in complex chemical networks I: the injectivity property. *SIAM J. Appl. Math.*, 65:1526–46, 2005.

[13] G. Craciun and M. Feinberg. Multiple equilibria in complex chemical networks II: the species-reaction graph. *SIAM J. Appl. Math.*, 66:1321–38, 2006.

[14] G. Craciun, Y. Tang, and M. Feinberg. Understanding bistability in complex enzyme-driven reaction networks. *Proc. Natl. Acad. Sci. USA*, 103:8697–02, 2006.

[15] A. Edelman and E. Kostlan. How many zeros of a random polynomial are real? *Bull. Amer. Math. Soc.*, 32:1–37, 1995.

[16] H. El-Samad, S. Prajna, A. Papachristodoulou, J. C. Doyle, and M. H. Khammash. Advanced methods and algorithms for biological networks analysis. *Proc. IEEE*, 94:832–53, 2006.

[17] M. Feinberg. Lectures on Chemical Reaction Networks. Lecture notes, Mathematics Research Center, University of Wisconsin, 1979. `www.che.eng.ohio-state.edu/~feinberg/research/`.

[18] K. Gatermann and B. Huber. A family of sparse polynomial systems arising in chemical reaction systems. *J. Symbolic Computation*, 33:273–305, 2002.

[19] J. Gunawardena. Chemical Reaction Network Theory for *in-silico* biologists. Lecture notes, Harvard Univ, 2003. `vcp.med.harvard.edu/papers/crnt.pdf`, 2003.

[20] J. Gunawardena. Multisite protein phosphorylation makes a good threshold but can be a poor switch. *Proc. Natl. Acad. Sci. USA*, 102:14617–22, 2005.

[21] D. Henrion and A. Garulli, editors. *Positive Polynomials in Control*. Number 312 in Lecture Notes in Control and Information Sciences. Springer, 2005.

[22] I. Herstein. *Topics in Algebra*. Wiley, 1975.

[23] M. W. Hirsch and S. Smale. *Differential Equations, Dynamical Systems and Linear Algebra*. Pure and Applied Mathematics. Academic Press, San Diego, USA, 1974.

[24] F. Horn and R. Jackson. General mass action kinetics. *Arch. Rational Mech. Anal.*, 47:81–116, 1972.

[25] R. Laubenbacher and B. Stigler. A computational algebra approach to the reverse engineering of gene regulatory networks. *J. Theor. Biol.*, 229:523–37, 2004.

[26] A. Mallavarapu, M. Thomson, B. Ullian, and J. Gunawardena. Programming with models: modularity and abstraction provide powerful capabilities for systems biology. *J. R. Soc. Interface*, 6:257–70, 2009.

[27] A. Manrai and J. Gunawardena. The geometry of multisite phosphorylation. *Biophys. J.*, 95:5533–43, 2008.

[28] M. Minimair and M. P. Barnett. Solving polynomial equations for chemical problems using Gröbner bases. *Mol. Phys.*, 102:2521–35, 2004.

[29] L. Pachter and B. Sturmfels, editors. *Algebraic Statistics for Computational Biology*. Cambridge University Press, 2005.

[30] L. Pachter and B. Sturmfels. The mathematics of phylogenomics. *SIAM Review*, 49:3–31, 2007.

[31] P. A. Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. PhD thesis, Caltech, 2000.

[32] S. Prajna, A. Papachristodoulou, P. Seiler, and P. A. Parrilo. SOSTOOLS and its control applications. Appearing in [21], 2005.

[33] J. R. Sendra, F. Winkler, and S. Pérez-Díaz. *Rational Algebraic Curves*, volume 22 of *Algorithms and Computation in Mathematics*. Springer, 2008.

[34] M. Thomson and J. Gunawardena. Multi-bit information storage by multisite phosphorylation. `arxiv.org/abs/0706.3735`, 2007.

[35] M. Thomson and J. Gunawardena. The rationality theorem for multisite post-translational modification systems. Submitted, 2009.

[36] L. Wang and E. D. Sontag. On the number of steady states in a multiple futile cycle. *J. Math. Biol.*, 57:29–52, 2008.

		$i = 0$	$i = 1$	$i = 2$	$i = 3$
a_i^E	nM^{-1}sec^{-1}	8.12×10^{-3}	1.02×10^{-1}	8.12×10^{-3}	1.02×10^{-1}
b_i^E	nM^{-1}sec^{-1}	1.60×10^{-2}	2.04×10^{-1}	1.60×10^{-2}	2.04×10^{-1}
$c_{i,i+1}^E$	sec^{-1}	1.00×10^{-1}	$1.00 \times 10^{+1}$	1.00×10^{-1}	$1.00 \times 10^{+1}$
a_{i+1}^F	nM^{-1}sec^{-1}	1.12×10^{-1}	2.64×10^{-3}	6.51×10^{-1}	2.85×10^{-3}
b_{i+1}^F	nM^{-1}sec^{-1}	2.24×10^{-1}	5.00×10^{-3}	$1.30 \times 10^{+0}$	6.00×10^{-3}
$c_{i+1,i}^F$	sec^{-1}	$1.10 \times 10^{+1}$	1.70×10^{-2}	$6.39 \times 10^{+1}$	1.36×10^{-1}
K_i^E	nM	$1.43 \times 10^{+1}$	$1.00 \times 10^{+2}$	$1.43 \times 10^{+1}$	$1.00 \times 10^{+2}$
K_{i+1}^F	nM	$1.00 \times 10^{+2}$	$8.33 \times 10^{+0}$	$1.00 \times 10^{+2}$	$5.00 \times 10^{+1}$
λ_i		6.38×10^{-2}	$5.05 \times 10^{+1}$	1.01×10^{-2}	$3.67 \times 10^{+1}$

Table 1: Parameter values for the $n = 4$ distributive, sequential model used in Paper Figure 2, following the notation for Example 1 in §1.7.

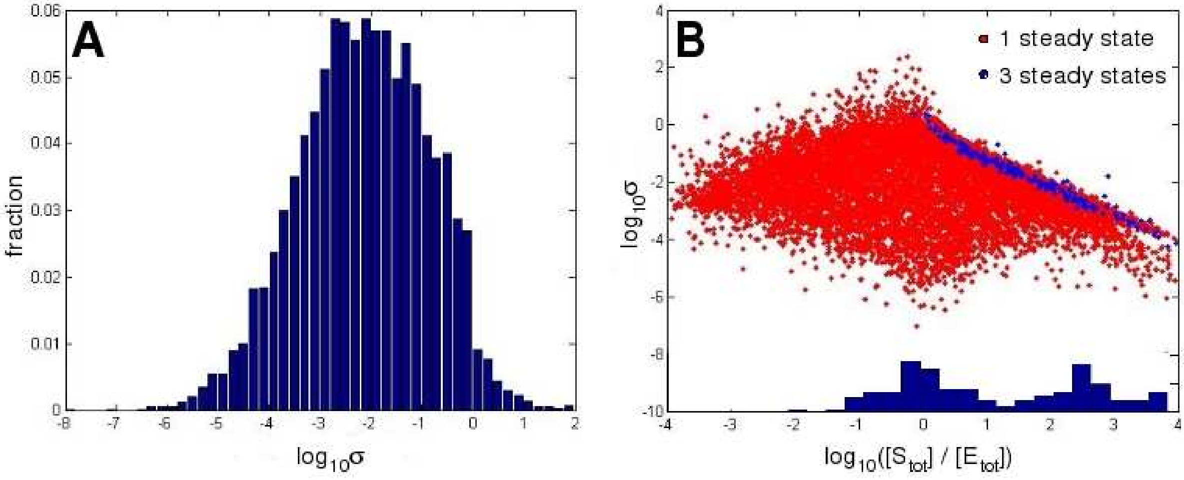

Figure 1: Approximation of Φ by $P(u)$. 10000 random systems were generated, as described in the text, and solved using Φ and $P(u) = 0$. For those which gave the same number of steady states, the discrepancy between the solutions was measured using σ, as described in the text. **A** Histogram of $\log_{10}\sigma$ values. **B** The top shows a scatter plot of $\log_{10}\sigma$ on the left vertical axis against $\log_{10} S_{tot}/E_{tot}$. The bottom shows the number of systems which gave different numbers of steady states for Φ and $P(u)$, using the lower part of the right vertical axis, binned against $\log_{10} S_{tot}/E_{tot}$.

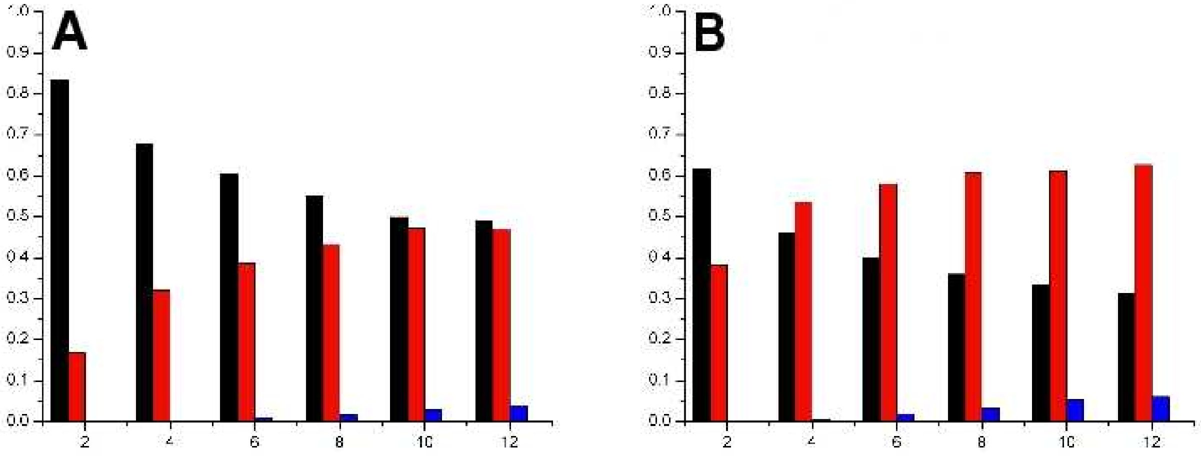

Figure 2: Frequency distributions of steady states for randomly chosen systems with $n = 2$ to 12 sites, as described in the text. **A** λ_i is chosen uniformly from site to site. **B** λ_i is biased to be low for even i and high for odd i. Vertical scales show frequency of occurrence of 1 (black), 3 (red) and 5 (blue) steady states, for 100,000 systems for each n.