Model balancing: consistent in-vivo kinetic constants and metabolic states obtained by convex optimisation

Wolfram Liebermeister

Model balancing: consistent *in-vivo* kinetic constants and metabolic states obtained by convex optimisation

Wolfram Liebermeister[1,2]

[1] Université Paris-Saclay, INRAE, MaIAGE, 78350, Jouy-en-Josas, France
[2] Institut für Biochemie, Charité – Universitätsmedizin Berlin, Germany

Abstract

Enzyme kinetic constants *in vivo* are largely unknown, which limits the construction of large metabolic models. In theory, kinetic constants can be fitted to measured metabolic fluxes, metabolite concentrations, and enzyme concentrations, but these estimation problems are typically non-convex. This makes them hard to solve, especially if models are large. Here I assume that the metabolic fluxes are given and show that consistent kinetic constants, metabolite levels, and enzyme levels can then be found by solving a convex optimality problem. If logarithmic kinetic constants and metabolite concentrations are used as free variables and if Gaussian priors are employed, the posterior density is strictly convex. The resulting estimation method, called model balancing, can employ a wide range of rate laws, accounts for thermodynamic constraints on parameters, and considers the dependences between flux directions and metabolite concentrations through thermodynamic forces. It can be used to complete and adjust available data, to estimate *in-vivo* kinetic constants from omics data, or to construct plausible metabolic states with a predefined flux distribution. To demonstrate model balancing and to assess its practical use, I balance a model of *E. coli* central metabolism with artificial or experimental data. The tests show what information about kinetic constants can be extracted from omics data and reveal practical limits of estimating kinetic constants *in vivo*.

Keywords: Metabolic model, enzyme kinetic constant, parameter estimation, convex optimality problem, parameter balancing, enzyme cost minimisation

1 Introduction

The number of metabolic network reconstructions is constantly growing, and there have been attempts to convert metabolic networks automatically into kinetic models. To build models with plausible parameter values and metabolic states (characterised by enzyme levels, metabolite levels, and fluxes), one needs to reconstruct the metabolic network, add allosteric regulation arrows, choose enzymatic rate laws, find kinetic constants, and make sure the model shows plausible metabolic states. These subproblems have been addressed in various ways. Pathway models have been built from *in-vitro* enzyme kinetics [1, 2]. To simplify model construction and to replace unknown rate laws, standardised rate laws have been proposed [3, 4], used for automatic model generation [5], and evaluated for their practical use [6]. *In-vitro* kinetic constants, available from the Brenda database [7, 8], are widely used and unknown k_{cat} values have been estimated by machine learning [9]. Directly inserting measured or sampled kinetic constants into models can lead to inconsistencies because thermodynamic dependencies between kinetic constants will be ignored. To address this problem, methods to construct consistent parameter sets have been devised [10, 11, 12, 13, 14] and applied in modelling [15]. In parallel, there have been attempts to estimate kinetic constants *in vivo* from flux, metabolite, and enzyme data [16, 17]. Methods for parameter fitting have been developed and benchmarked [18, 19], and the question of parameter identifiability has been addressed [20].

Large models have been parameterised [21, 22], and pipelines for model parameterization have been developed [23, 24]. Finally, even if parameters are unknown, methods for parameter sampling and ensemble modelling allow to find plausible parameter sets [25] and to draw conclusions about possible dynamic behaviour [26].

The key problem here is to obtain realistic, consistent values of kinetic constants. *In-vivo* values are hard to measure, and *in-vitro* values as proxies may be unreliable – or at least, this is hard to assess unless *in-vivo* values are known. So how can we infer model parameters from omics data? To obtain realistic model parameters and metabolic states, various types of measurement data must be combined. In an ideal case, in which kinetic constants and enzyme concentrations are precisely known, metabolite levels and fluxes could be computed by simulating the model. In another ideal case, in which enzyme levels, metabolite levels and fluxes are precisely known for a number of states, we can solve for the kinetic constants. In reality, we are in between these two cases: data of different types are available, but these data are incomplete and too noisy to be used directly in models. In practice, in model construction we may have different aims, for example (i) finding consistent kinetic constants in plausible ranges; (ii) adjusting and completing a data set of measured kinetic constants to obtain consistent model parameters; (iii) estimating *in-vivo* kinetic constants from omics data (measured enzyme levels, metabolite levels, and fluxes); (iv) completing and adjusting omics data for consistent metabolic states, which may involve predictions of physilogical metabolite concentrations [27, 28] or predictions of metabolite and enzyme concentrations based on resource allocation principles [29, 30, 31].

Taking all this together, our general task is not only to fit kinetic constants, but also to reconstruct consistent metabolite concentrations, enzyme concentration, and metabolic fluxes, for a given model and based on data for all these quantities. A notorious problem in model fitting is that data are uncertain, inconsistent and incomplete. Thus, in our estimation task uncertainties in data and estimated parameters need to be quantified. Luckily, we can employ some further constraints: our solutions must satisfy some general physical laws, namely thermodynamic relations (Wegscheider conditions and Haldane relationships) between kinetic constants [3, 4] and relations between kinetic constants and metabolic variables (e.g. the flux sign constraints, which couple flux directions, equilibrium constants, and metabolite levels). Moreover, we can use prior distributions (for kinetic constants, metabolite levels, or enzyme levels); and we can use measured (*in vitro*) parameter values as additional data (of course, these data cannot be used as test data anymore). However, one problem remains. The resulting optimality problems, e.g. in maximum-likelihood estimation, will be non-convex: they may show multiple local optima and globally optimal solutions may be hard to find, especially with larger networks.

Here I address the general problem of finding consistent kinetic constants and metabolic states, based on heterogenous (kinetic, metabolomics, and proteomics) data. Under the strong assumption that the metabolic fluxes are known (from measurements or previous calculations), I show that parameter fitting in kinetic metabolic models can be formulated as a convex optimality problem. The estimation method, called model balancing, uses the following input data: measured or assumed values of kinetic constants (which may be incomplete and uncertain), and measured or assumed values of metabolite and enzyme levels in a number of metabolic states (which may be incomplete and uncertain); and it relies on precise metabolic fluxes from one or several metabolic states (which may be stationary or non-stationary). It determines a set of kinetic parameters and state variables (metabolite and enzyme levels for all metabolic states in question) that are consistent with the rate laws and all other dependencies in the model, plausible (i.e. respecting constraints and prior distributions), and resemble the data (showing large likelihood values). For the estimation, we can either follow a maximum-likelihood approach (leading to a convex optimality problem) or a Bayesian maximum-posterior approach (where Gaussian priors ensure strict convexity). The maximum-posterior problem has a single solution which can be found be gradient-descent methods, and also posterior sampling is facilitated by strict convexity. Model balancing relies on two main assumptions: (i) all fluxes are predefined and thermodynamically consistent (i.e. infeasible cycle fluxes must be excluded) and (ii) kinetic constants and metabolite concentrations are treated on logarithmic scale, while enzyme concentrations are treated on absolute scale. Model balancing builds upon some methods for metabolic model construction:

parameter balancing [11, 14], elasticity sampling [32], and enzyme cost minimisation [31], which I review in the discussion section.

2 Parameter estimation in kinetic models as a convex problem

2.1 Estimating kinetic constants from omics data

To see how information about *in-vivo* kinetic constants is extracted from omics data, let us review an existing approach. To estimate k_{cat} values, Davidi *et al.* [16] compared measured proteomics data to flux data obtained from flux balance analysis (FBA), without presupposing any knowledge of metabolite concentrations or specific kinetic laws. The method works as follows. We assume unknown rate laws of the form $v = e\,k(\mathbf{c})$ (with flux v and enzyme concentration e), where the catalytic rate k depends on (unknown) metabolite levels (vector $\mathbf{c}$) and can vary between zero and a value k_{cat} (called turnover rate or catalytic constant). To determine k_{cat} from data, we consider a cell in different metabolic states and assume that each enzyme reaches its maximal capacity in at least one of these states. Based on this assumption, a k_{cat} value is estimated by computing the empirical catalytic rates $v^{(s)}/e^{(s)}$ in all states and taking their maximum value. The method was applied to a large number of enzymes in *E. coli*, and the estimated k_{cat} values were found to resemble the measured *in-vivo* values. Some of the deviations could be explained by enzyme kinetics and thermodynamics, but this was not quantitatively modelled. The limitations of this method are clear: since the $\max$ function is only sensitive to the highest value, one high outlier value can completely distort the result. Such outliers may arise if a small protein level, due to measurement errors, appears even smaller. But aside from this practical problem, what if the basic assumption is not satisfied? We cannot be sure that an enzymes reaches its maximal capacity in one of the samples, so the estimated *in-vivo* k_{cat} value should be seen as a lower bound $k_{\mathrm{cat}} \geq \max_s v^{(s)}/e^{(s)}$. But how far is this bound from the true *in-vivo* k_{cat} value?

If we manage to explain the (non-maximal) catalytic rates in the different states, can we maybe obtain a better estimate of the true k_{cat} value, even if this value is reached in none of the samples? To do this, we need to consider metabolic concentrations and enzyme kinetics, i.e. the functional form of $k_l(\mathbf{c})$. A typical form of $k(\mathbf{c})$ for a uni-uni reaction, the Michaelis-Menten kinetics, is given by $v = \frac{k^+_{\mathrm{cat}}\, s/K_s - k^-_{\mathrm{cat}}\, p/K_p}{1+s/K_S+p/K_p}$ [4] or, in factorised form [33], by $v = k^+_{\mathrm{cat}} \cdot \eta^{\mathrm{rev}}(\mathbf{c}) \cdot \eta^{\mathrm{sat}}(\mathbf{c})$, where the efficiency terms η^{rev} (for reversibility, or thermodynamics) and η^{sat} (for enzyme saturation and allosteric regulation) are numbers between 0 and 1 depending on metabolite levels. If the efficiency terms are close to 1, then k approaches its maximal value k_{cat}; but normally k is lower. Given the rate laws, and given data for fluxes $\mathbf{v}$, metabolite levels $\mathbf{c}$, and enzyme levels $\mathbf{e}$, we might be able to estimate k_{cat} and K_{M} values, even if the maximal efficiency is not reached in any of the samples.

2.2 Metabolic model and statistical estimation model

Let us start by stating the estimation problem (shown in Figure 1 (a)). We consider a kinetic metabolic model with thermodynamically consistent modular rate laws [4] and kinetic constants (e.g. equilibrium constants[1], catalytic constants and Michaelis-Menten constants) in a vector $\mathbf{q} = \ln \mathbf{p}$. The model shows a number of metabolic states, each characterised by a flux vector $\mathbf{v}^{(s)}$, a metabolite concentration vector $\mathbf{c}^{(s)}$, and an enzyme concentration vector $\mathbf{e}^{(s)}$. These states can be stationary (with steady-state fluxes) or non-stationary (e.g. snapshots from a dynamic time course). The model formulae define dependencies among model parameters and state variables. The kinetic constants in a network are interdependent because of physical laws [3, 4]. Each rate law contains a

[1] Equilibrium constants are determined by thermodynamics and do not depend on specific enzymes, but for simplicity I will count them among the kinetic constants.

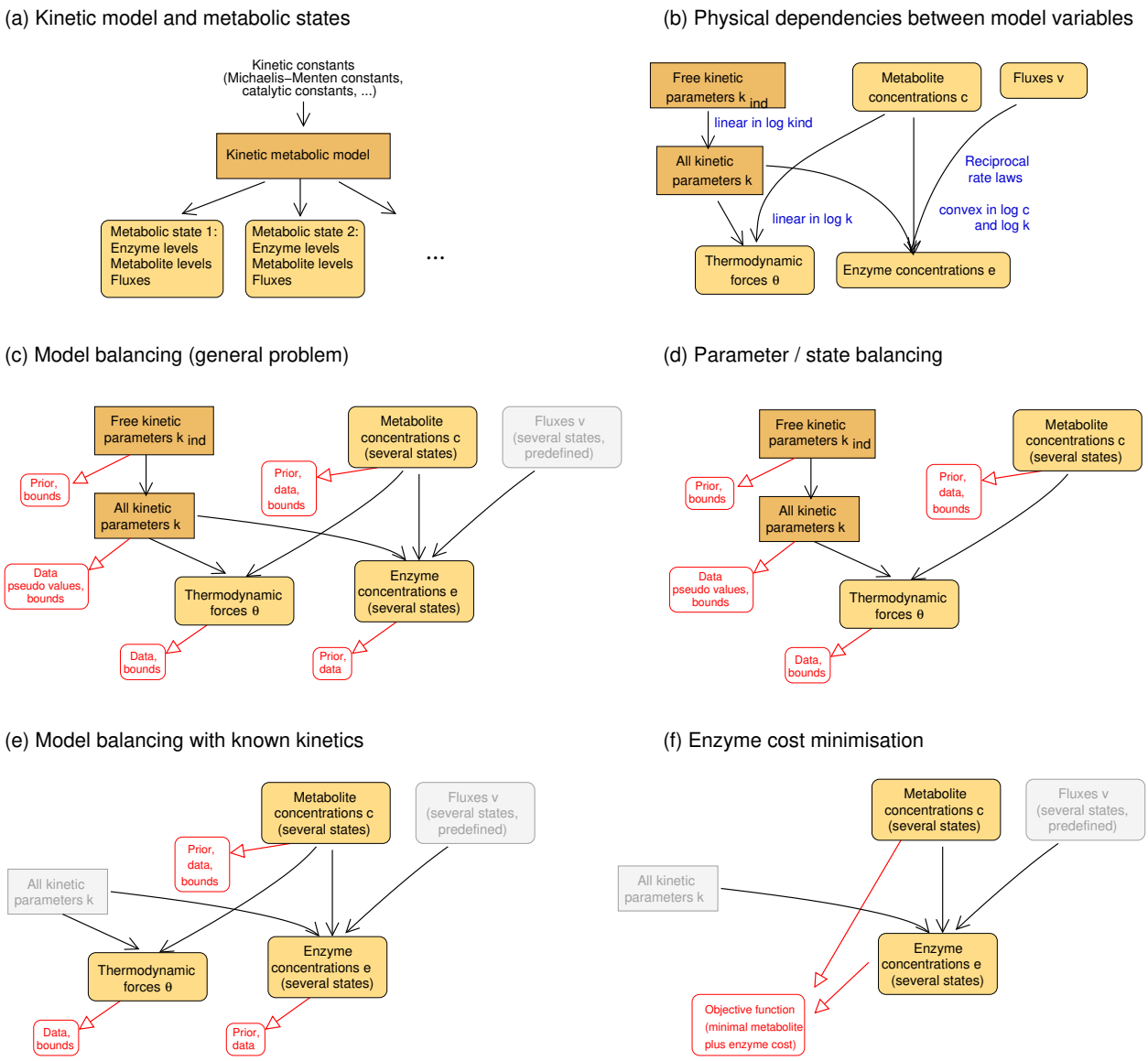

Figure 1: Parameter estimation in kinetic metabolic models. (a) Kinetic model and metabolic states. A model is parameterised by kinetic constants (e.g. equilibrium constants, catalytic constants, and Michaelis-Menten constants) and gives rise to a number of metabolic states (characterised by enzyme levels, metabolite levels, and fluxes). These states may be stationary (with steady-state fluxes) or not (e.g. states during dynamic time courses). (b) Dependencies between kinetic constants and state variables. All kinetic constants are described on logarithmic scale, and a subset of kinetic constants determines all other kinetic constants through linear relationships. If kinetic constants, metabolite levels, and fluxes are known, the enzyme levels can be computed from rate laws and fluxes: each enzyme level is a convex function of the (logarithmic) kinetic constants and metabolite levels. (c) Parameter estimation. Kinetic constants and metabolite levels (for a number of metabolic states) are the free variables of a statistical model. Dependent kinetic constants, thermodynamic driving forces, and enzyme levels (bottom) are treated as dependent variables, and the fluxes (top right) are predefined. For estimating the variables, priors and available data may be used. The other subfigures show similar estimation and optimisation methods, in which (d) only kinetic data are balanced (no metabolic data), (e) only metabolic data are balanced (kinetic parameters are predefined), or (f) enzyme and metabolite levels are optimised for a low biological cost.

forward and a reverse catalytic constant as well as the Michaelis-Menten constants[2], and all these parameters in a model may depend on each other via Haldane relationships and Wegscheider conditions.

The dependencies between model variables are summarised in appendix A.1. To satisfy all parameter dependencies

[2]Activation and inhibition constants are independent of all other constants and are therefore independent parameters.

in our model, we introduce a set of independent kinetic parameters (independent equilibrium constants, Michaelis-Menten constants, and velocity constants) from which all remaining constants can be derived (see Figure 1 (b), top left). The vector $\mathbf{p}$ contains all kinetic constants. In each metabolic state s, the rate laws define equalities $v_l^{(s)} = e_l^{(s)}\, k_l(\mathbf{p}, \mathbf{c}^{(s)})$ for enzyme levels $e_l^{(s)}$, metabolite levels $c_i^{(s)}$, and catalytic rates k_l. By inverting this equation, the enzyme levels can be written as functions

$$e_l^{(s)} = \frac{v_l^{(s)}}{k_l(\mathbf{p}, \mathbf{c}^{(s)})} \tag{1}$$

of kinetic constants, metabolite levels, and fluxes (Figure 1 (b), bottom). The signs of thermodynamic forces given by a vector $\boldsymbol{\theta}^{(s)} = \ln \mathbf{k}_{\mathrm{eq}} - \mathbf{N}_{\mathrm{all}}^{\top} \ln \mathbf{c}^{(s)}$ determine the flux directions (where reactions with vanishing fluxes are always allowed). This law holds for all thermodynamically feasible rate laws.

Given a metabolic model with all its variables and dependencies, we now define estimation problems (see Figure 1 (c)). The most general aim is to estimate kinetic constants, metabolite profiles and enzyme profiles in a number of metabolic states. Available data may comprise kinetic constants, metabolite and enzyme concentrations, and possibly thermodynamic forces in a number of metabolic states, and metabolic fluxes in the same metabolic states. All data may be uncertain and incomplete, except for the fluxes, which must be precisely given. Moreover, we may use prior distributions and impose upper and lower bounds on the model parameters and on metabolite and enzyme levels. In the model, all dependencies must be satisfied. To get to a convex optimality problem, we treat the (logarithmic[3]) independent kinetic constants and the (logarithmic) dependent kinetic constants and (logarithmic) metabolite concentrations as free variables, while the (non-logarithmic) enzyme levels and thermodynamic forces are dependent variables to be computed from kinetic constants, metabolite levels, and fluxes. The vector of free variables (logarithmic kinetic constants and metabolite concentrations) is constrained by thermodynamic laws, and the resulting feasible space is a convex polytope. We may consider two variants of the estimation problem, maximum-likelihood estimation and maximum-posterior estimation [34]. In maximum-likelihood estimation, we minimise the negative log-likelihood (or "likelihood loss"), a convex function on the feasible polytope. In maximum-posterior estimation, we consider Gaussian priors, which make the negative log-posterior density (or "posterior loss") strictly convex on the feasible polytope. This means: the posterior mode is unique and can be obtained by convex optimisation. Formulae are summarised in appendix A.1.

2.3 A simplified estimation problem: fitting of metabolite and enzyme levels

Before we get to the full model balancing proablem, let us first assume that the kinetic constants are known and let us estimate metabolite and enzyme levels for a single steady state[4], based on data with error bars for (some or all) metabolite and enzyme levels. To fit consistent metabolite and enzyme levels to these data, we maximise either their likelihood or the posterior density. For the log-metabolite vector $\mathbf{x}$, we assume an uncorrelated Gaussian prior (with mean vector $\bar{\mathbf{x}}_{\mathrm{prior}}$ and covariance matrix $\mathbf{C}_{\mathrm{x,prior}} = \mathrm{Dg}(\boldsymbol{\sigma}_{\mathrm{x,prior}})^2$) and lower and upper bounds (possibly different for each metabolite). For the enzyme vector $\mathbf{e}$, we assume an uncorrelated Gaussian prior (with mean vector $\bar{\mathbf{e}}_{\mathrm{prior}}$ and covariance matrix $\mathbf{C}_{\mathrm{e,prior}} = \mathrm{Dg}(\boldsymbol{\sigma}_{\mathrm{e,prior}})^2$). Negative values are not allowed ($e_l \geq 0$). The possible logarithmic metabolite profiles $\mathbf{x}$ form a convex polytope $\mathcal{P}_{\mathbf{x}}$ in log metabolite space [31]. This shape of this polytope is defined by physiological upper and lower bounds and by thermodynamic constraints, depending on flux directions and equilibrium constants. The logarithmic metabolite concentrations x_i, our free variables, determine the enzyme levels e_l through Eq. (1), and the enzyme levels are convex functions on the metabolite polytope. As a consequence, the likelihood function is convex. Thus, to define an estimation problems,

[3] Natural logarithms are used throughout the text.

[4] Mathematically, this estimation problem resembles Enzyme Cost Minimisation [31]. Both methods are based on kinetic models with known parameters and predefined fluxes, and both of them optimise metabolite and enzyme levels, but in different ways. In enzyme cost minimisation, while in the present estimation problem, metabolite and enzyme levels are fitted to measured data.

we construct the poltyope, consider prior, likelihood and posterior functions on this polytope, and use them to estimate metabolite concentrations and corresponding enzyme levels.

Assuming prior distributions for $\mathbf{x}$ and $\mathbf{e}$, we define the *preprior loss function*[5]

$$P'(\mathbf{x},\mathbf{e}) \quad = \quad (\mathbf{x}-\bar{\mathbf{x}}_{\mathrm{prior}})^\top \mathbf{C}^{-1}_{\mathbf{x}\mathrm{prior}} (\mathbf{x}-\bar{\mathbf{x}}_{\mathrm{prior}}) + (\mathbf{e}-\bar{\mathbf{e}}_{\mathrm{prior}})^\top \mathbf{C}^{-1}_{\mathbf{e}\mathrm{prior}} (\mathbf{e}-\bar{\mathbf{e}}_{\mathrm{prior}}), \tag{2}$$

the negative logarithmic prior density, where constant terms and the prefactor[6] $\frac{1}{2}$ are ignored. Similarly, using data for $\mathbf{x}$ and $\mathbf{e}$, we define the *prelikelihood loss function*

$$L'(\mathbf{x},\mathbf{e}) \quad = \quad (\mathbf{P}_\mathrm{x}\,\mathbf{x}-\bar{\mathbf{x}}_{\mathrm{data}})^\top \mathbf{C}^{-1}_{\mathbf{x}\mathrm{data}} (\mathbf{P}_\mathrm{x}\,\mathbf{x}-\bar{\mathbf{x}}_{\mathrm{data}}) + (\mathbf{P}_\mathrm{e}\,\mathbf{e}-\bar{\mathbf{e}}_{\mathrm{data}})^\top \mathbf{C}^{-1}_{\mathbf{e}\mathrm{data}} (\mathbf{P}_\mathrm{e}\,\mathbf{e}-\bar{\mathbf{e}}_{\mathrm{data}}), \tag{3}$$

the negative log-likelihood (again without constant terms and the prefactor). The vectors $\bar{\mathbf{x}}_{\mathrm{data}}$ and $\bar{\mathbf{e}}_{\mathrm{data}}$ contain mean values and the matrices $\mathbf{C}_{\mathbf{x}\mathrm{data}} = \mathrm{Dg}(\sigma_{\mathbf{x},\mathrm{data}})^2$ and $\mathbf{C}_{\mathbf{e}\mathrm{data}} = \mathrm{Dg}(\sigma_{\mathbf{e},\mathrm{data}})^2$ contain covariances for measurement data. The projection matrices $\mathbf{P}_\mathrm{x}$ and $\mathbf{P}_\mathrm{e}$ map the concentrations of all metabolite and enzyme levels to those concentrations that appear in the measured data. The function L' is convex in $\mathbf{x}$ and $\mathbf{e}$, and P' is strictly convex. If we add the two functions, we obtain the *preposterior loss function* $R'(\mathbf{x},\mathbf{e}) = P'(\mathbf{x},\mathbf{e}) + L'(\mathbf{x},\mathbf{e})$. By adding Eqs (2) and (4) and simplifying the quadratic functions (as in [10] and [11]), we obtain the formula

$$R'(\mathbf{x},\mathbf{e}) \quad = \quad (\mathbf{x}-\bar{\mathbf{x}}_{\mathrm{pre}})^\top \mathbf{C}^{-1}_{\mathbf{x},\mathrm{pre}} (\mathbf{x}-\bar{\mathbf{x}}_{\mathrm{pre}}) + (\mathbf{e}-\bar{\mathbf{e}}_{\mathrm{pre}})^\top \mathbf{C}^{-1}_{\mathbf{e},\mathrm{pre}} (\mathbf{e}-\bar{\mathbf{e}}_{\mathrm{pre}}) \tag{4}$$

with covariance matrices and mean vectors

$$\begin{aligned} \mathbf{C}_{\mathbf{x},\mathrm{pre}} \quad &= \quad [\mathbf{C}^{-1}_{\mathbf{x},\mathrm{prior}} + \mathbf{P}_\mathrm{x}^\top \mathbf{C}^{-1}_{\mathbf{x},\mathrm{data}} \mathbf{P}_\mathrm{x}]^{-1} \\ \bar{\mathbf{x}}_{\mathrm{pre}} \quad &= \quad \mathbf{C}_{\mathbf{x},\mathrm{pre}}\,[\mathbf{C}^{-1}_{\mathbf{x},\mathrm{prior}}\,\bar{\mathbf{x}}_{\mathrm{prior}} + \mathbf{P}_\mathrm{x}^\top\,\mathbf{C}^{-1}_{\mathbf{x},\mathrm{data}}\,\bar{\mathbf{x}}_{\mathrm{data}}]. \end{aligned} \tag{5}$$

Analogous formulae hold for $\bar{\mathbf{e}}_{\mathrm{pre}}$ and $\mathbf{C}^{-1}_{\mathbf{e},\mathrm{pre}}$.

Why is R' called "preposterior" and not simply "posterior"? The preposterior contains enzyme levels as function arguments, but the enzyme levels are dependent on metabolite levels and fluxes. By inserting the enzyme demand function Eq. (1) into Eq. (4), we reobtain the three loss scores, but as functions of $\mathbf{x}$ alone:

$$\begin{aligned} \text{Prior loss } P(\mathbf{x}) \quad &= \quad (\mathbf{x}-\bar{\mathbf{x}}_{\mathrm{prior}})^\top \mathbf{C}^{-1}_{\mathbf{x}\mathrm{prior}} (\mathbf{x}-\bar{\mathbf{x}}_{\mathrm{prior}}) + (\mathbf{e}(\mathbf{x})-\bar{\mathbf{e}}_{\mathrm{prior}})^\top \mathbf{C}^{-1}_{\mathbf{e}\mathrm{prior}} (\mathbf{e}(\mathbf{x})-\bar{\mathbf{e}}_{\mathrm{prior}}) \\ \text{Likelihood loss } L(\mathbf{x}) \quad &= \quad (\mathbf{P}_\mathrm{x}\,\mathbf{x}-\bar{\mathbf{x}}_{\mathrm{data}})^\top \mathbf{C}^{-1}_{\mathbf{x}\mathrm{data}} (\mathbf{P}_\mathrm{x}\,\mathbf{x}-\bar{\mathbf{x}}_{\mathrm{data}}) + (\mathbf{P}_\mathrm{e}\,\mathbf{e}(\mathbf{x})-\bar{\mathbf{e}}_{\mathrm{data}})^\top \mathbf{C}^{-1}_{\mathbf{e}\mathrm{data}} (\mathbf{P}_\mathrm{e}\,\mathbf{e}(\mathbf{x})-\bar{\mathbf{e}}_{\mathrm{data}}) \\ \text{Posterior loss } R(\mathbf{x}) \quad &= \quad (\mathbf{x}-\bar{\mathbf{x}}_{\mathrm{pre}})^\top \mathbf{C}^{-1}_{\mathbf{x}\mathrm{pre}} (\mathbf{x}-\bar{\mathbf{x}}_{\mathrm{pre}}) + (\mathbf{e}(\mathbf{x})-\bar{\mathbf{e}}_{\mathrm{pre}})^\top \mathbf{C}^{-1}_{\mathbf{e}\mathrm{pre}} (\mathbf{e}(\mathbf{x})-\bar{\mathbf{e}}_{\mathrm{pre}}). \end{aligned} \tag{6}$$

The enzyme demand $\mathbf{e}(\mathbf{x})$ is a convex function on the metabolite polytope [31] for a wide range of plausible rate laws. Therefore, likelihood loss and persterior loss are convex functions, and the posterior mode can be found by convex optimisation[7].

Our estimation method can be extended to problems with several metabolic states, where each condition s has its own flux distribution, metabolite data, and enzyme data. In fact, in this case we can run the estimation separately for each state (see also appendix A.1). In an estimation problem with a single metabolic state, non-zero fluxes can be assumed (because reactions with vanishing flux can be simply omitted). For problems with several states, vanishing fluxes can be considered (see appendix C.2).

[5] If desired, prior and likelihood terms for thermodynamic forces may be included.

[6] In the matlab implementation, in contrast, this prefactor is used.

[7] Since P' and L' are convex in the vectors $\mathbf{x}$ and $\mathbf{e}$, and since $\mathbf{e}$ is convex in $\mathbf{x}$, the loss terms P and L are convex in $\mathbf{x}$. If P is strictly convex in $\mathbf{x}$, the posterior loss $P(\mathbf{x}) + L(\mathbf{x}) + \mathrm{const.}$ is also strictly convex. Since the feasible polytope is convex as well, computing the posterior mode is a convex optimality problem.

2.4 Simultaneous estimation of kinetic constants and metabolic states

We now consider the full model balancing problem, that is, the simultaneous estimation of kinetic constants, metabolite levels, and enzyme levels. Following [3], we parametrize the model by kinetic constants K_{eq}, K_{V}, K_{M}, and possibly K_{A} and K_{I} (all on log scale). Some of them may be available as data (for instance, equilibrium constants K_{eq} can be estimated from thermodynamic calculations) and the true values of all these quantities need to be estimated. This problem resembles our simplified problem, where the enzyme levels were convex in $\mathbf{x}$. Now the enzyme levels also depend on kinetic constants, but they are convex in the logarithmic kinetic constants as well! A description of the algorithm, including the convexity proof, is given in appendix B.1. Here I summarise some main points. Since state variables and kinetic constants are estimated together, and since the kinetic constants are kept constant across metabolic states, the state variable become coupled across metabolic states and need to be estimated in one go. Instead of a metabolite vector $\mathbf{x}$, we consider a larger vector $\mathbf{y}$, containing the log-metabolite levels for all metabolic states and the vector of logarithmic kinetic constants. Allowed ranges and thermodynamic constraints define a feasible polytope for the vector $\mathbf{y}$. The prior, likelihood, and posterior loss functions contain terms that depend on enzyme levels $\mathbf{e}(\mathbf{x})$. If we insert Eq. (1) into these formulae, these terms are convex in the logarithmic kinetic parameters, and independent of the metabolite levels[8]. Since $e_l(\mathbf{q}, \mathbf{x})$ is a convex function of the vector $\mathbf{y} = \binom{\mathbf{x}}{\mathbf{q}}$, all terms of the likelihood loss function are convex in $\mathbf{y}$. The prior loss function is strictly convex in $\mathbf{y}$ if pseudo values for kinetic constants are considered [11] (pseudo values are a way to define priors by which all model parameters, even dependent ones, have non-flat priors). Details are given in appendix B.1. Altogether, our estimation problem has the same good properties as the previous, simplified problem. In practice, the model balancing algorithm can be improved by a number of simplifications and tricks (appendix C). For example, enzyme levels (and therefore the likelihood function) increase very steeply close to some polytope boundaries; to avoid numerical problems, regions close to the boundary may be excluded by extra constraints, and the log(log posterior) may be minimised instead of the log posterior.

3 Example applications

Our test case for model balancing is a model of *E. coli* central metabolism (Figure 16 in appendix16), including metabolite, enzyme, and kinetic data, taken from [31]. The model contains no allosteric regulation, but such regulations could be added and K_{I} and K_{A} values could be estimated. We consider different estimation scenarios, with artificial data, experimental data from one metabolic state (data from [31]), or experimental data from three metabolic states (data from [16]). The same algorithm settings (such as priors or bounds) were used in all tests (with artificial or experimental data). For details on model structure, kinetic and metabolic data, and priors see appendix D.

3.1 E. coli metabolic model: tests with artificial data

I first generated artificial parameter sets containing kinetic constants and metabolic data (metabolite levels, enzyme levels, and fluxes). Artificial data were generated by using the same random distributions (means and widths) that were also used as priors in model balancing. Metabolic state variables were generated from the kinetic model (parameterised by artificial kinetic constants) by computing steady states with randomly chosen enzyme levels and external metabolite levels. For details on artificial data, see appendix E. Based on (noise-free or noisy) artificial data for six simulated metabolic states, model balancing was used to reconstruct the true (noisy-free) values. In different scenarios (see Figure 17 in appendix E), data were either fitted (metabolite and enzyme levels,

[8] The preposterior for kinetic constants is given by the posterior obtained from parameter balancing. For more details, see appendix C.2.

and "known" kinetic constants) or predicted based on the other data ("unknown" kinetic constants).

The results of model balancing with artificial data are shown in Figures 3, 4, 5, and 6, where kinetic or state data were either noise-free or noisy. Figure 3 shows the results for noise-free kinetic and state data. Subfigures show different simulation and estimation scenarios (rows) and different types of variables (columns). Each subfigure shows a scatter plot between true and fitted variables (metabolite levels, enzyme levels, and different types of kinetic constants). Deviations from the diagonal (in y-direction) indicate estimation errors in the kinetic constants. In the top subfigure row, data for all kinetic constants were given; in the centre row, only data for equilibrium constants were used, and in the bottom row, no kinetic data were used[9]. Depending on the scenario, kinetic constants were then either fitted (red dots) or predicted from data (magenta dots). The quality of the fit or prediction is assessed by geometric standard deviations[10] and linear (Pearson) correlations (for logarithmic values, except for the case of enzyme levels). For comparison, I also estimated k_{cat} values by maximal apparent k_{cat} values [16], based on the same artificial data (Figure 7).

The first scenario (top row) shows ideal conditions: we assume noise-free, complete kinetic data and state data. Not surprisingly, the reconstruction errors are very small, arising from small conflicts between data and priors. The other rows show the estimation results based on equilibrium constants only (centre row), or using no kinetic data at all (bottom row). With noise-free data, the reconstructions in these two rows have a similar quality. To assess the effect of noisy data, I generated artificial metabolic data (metabolite levels, enzyme levels, and fluxes) with a relative noise level of 20 percent. With noisy kinetic and/or metabolic data, the estimation results become worse (Figures 4, 5, and 6), and especially the reconstruction of K_{M} values becomes very poor. Using data on equilibrium constants improves the results and k_{cat} values can still be partially reconstructed (Figure 6). Even in the case without any kinetic data (nor equilibrium constants), model balancing yields better k_{cat} estimates than the "maximal apparent catalytic rate" method.

The tests with artificial data show that model balancing can adjust noisy data sets, yielding complete, consistent model parameters and states, and that it can extract information about k_{cat} values from metabolic data. The results are better than with the "maximal apparent k_{cat}" method, and known equilibrium constants improve the results. This is good news, because equilibrium constants are not enzyme-dependent and can be estimated from molecule structures [35, **?**]. K_{M} values are harder to reconstruct: the estimates are in realistic ranges (probably due to the priors), but they appear to be randomly distributed unless noise-free metabolite and enzyme data are used.

3.2 E. coli metabolic model with experimental data

As a next test, I balanced the *E. coli* model with experimental data. As kinetic data, I used *in-vitro* kinetic constants collected in [31] for the same model ("original kinetic data"), as well as a completed, balanced version of this data set ("balanced kinetic data"). For details on model and data, see appendix D.

Figures 8 and 9 show estimation results for a single metabolic state, aerobic growth on glucose; see appendix D. Since the "true" metabolic data *in-vivo* kinetic constants, are not known, the reconstructed kinetic constants, metabolite levels and enzyme levels are compared to the data used for the reconstruction. In a first test, I used a set of kinetic data obtained by parameter balancing (Figure 8). If all kinetic data are used (top row), a good fit to these data is achieved. On the contrary, even with noisy kinetic constants slight adjustments suffice to obtain a consistent kinetic model agreeing with all data available. However, the kinetic constants were fitted and not predicted (as indicated by red dots). In the centre row, where equilibrium constants were used as the only kinetic data, there is no such bias. This time, the kinetic constants are actually predicted (as indicated by magenta dots)

[9] The bottom subfigure rows (estimation without kinetic data) are repeated between Figures 3 and 4, and accordingly between other figures.

[10] The geometric standard deviation is defined as $\exp(\sigma)$, where σ is the root mean square of the residuals on (natural) log scale.

and show correlations to *in-vitro* values. However, there may still be some bias because the kinetic constants (to which I compare the predictions) had been balanced using the same network model and the same priors as used in model balancing. To avoid this bias, I next ran model balancing with the original *in-vitro* kinetic data (which contain much fewer data points for comparison). As shown in Figure 9, the predicted k_{cat} estimates still capture a trend in the *in-vitro* data (Pearson correlation 0.64 with usage of K_{eq} data and 0.29 without K_{eq} data). Again, in comparison to the method of maximal apparent catalytic constants (see Figure 10) model balancing performs better.

A single metabolic state contains too little information to estimate the kinetic constants[11]. Therefore, I repeated the estimation, no using metabolic data from three different states (growth on glucose, glycerol, and acetate) and assuming that the kinetic constants do not change between these states (see appendix D). Figures 11, 12, and 13 show the results. Just like before, a consistent model was obtained by moderate changes in the data. An estimation using equilibrium constants predicted k_{cat} values more reliably than the "maximal apparent k_{cat} value" method. Unexpectedly, using three states instead of one did not considerably improve the estimation results.

3.3 Parameter identifiability and choice of priors

To see how much information can be extracted from our data, we need to think about parameter identifiability and about the choice of priors.

In parameter estimation, parameters or parameter ratios may be non-identifiable, that is, their values cannot be inferred from the given model and data. In our Bayesian method, Gaussian priors guarantee a uniquely determined posterior mode, but if parameters are non-identifiable, their values will only reflect the priors (which means that high values will be underestimated and low values will be overestimated). This problem must arise if there are fewer data values than variables to be estimated. For example, metabolic data from a single metabolic state may not suffice to reconstruct the kinetic constants; if more metabolic states are used, the kinetic constant may become well-defined. In practice, we are faced with several questions: is the algorithm able to find the posterior mode? Can we improve the result by using more data (e.g., metabolite levels from more metabolic states)? If no kinetic data are given, how many metabolic states are needed to identify all kinetic parameters? Which parameters are hard to reconstruct? And are there kinetic constants that remain non-identifiable, no matter how much metabolic data we use?

If an enzyme is always saturated with a metabolite, that is, if the metabolite level is always much larger than the K_{M} value, the K_{M} value is hard to estimate because it has practically no effect on measurable variables. In the reconstructed parameter set, such K_{M} are likely to carry large errors (i.e. posterior variances). A similar problem occurs if the K_{M} value of a unimolecular reaction is always much *smaller* than the metabolite level; in this case, the enzyme works in its linear range, and only the ratio $k_{\mathrm{cat}}/K_{\mathrm{M}}$ is identifiable, while the k_{cat} and K_{M}, individually, are not. If an enzyme in question is *always* saturated or always in the linear range, this is less of a problem, because then parameters that are non-identifiable are also irrelevant for model predictions. However, predictions for other experiments, in which the enzyme *does* behave differently, may be poor. Of course, the identifiability problem is not specific to model balancing; other estimation methods would face the same problem.

In model balancing, like in other estimation methods, priors and measurement error bars must be carefully chosen. In the tests with artificial data, realistic statistical distributions (for kinetic constants, metabolic variables, and their measurement errors) were used to generate data, and the same distributions were used as priors when reconstructing the true values. This is an ideal situation. In real-life applications, if our priors and assumed noise

[11] We can see this by considering possible parameter variations: if a single state is considered, a change in a K_{M} value can be compensated by a simultaneous change in k_{cat} values (yielding the same flux at given metabolite and enzyme levels). Therefore, K_{M} values and k_{cat} values are, individually, non-identifiable. Nevertheless, using priors we may still obtain reasonable estimates of all parameters.

levels are wrong, the reconstruction would be worse than suggested by our tests with artificial data. To obtain the realistic distributions of kinetic constant mentioned before, I started from known (or suspected) distributions (from [31, 14], which relied on [8]), and adjusted them based on data. By visual inspection during parameter balancing, I noticed that some priors had to be changed, probably because kinetic constants in central metabolism are differently distributed than kinetic constants in metabolism in general.

4 Discussion

Various methods and modelling tools have been developed to parameterise kinetic models. They use different types of knowledge (*in-vitro* kinetic constants, omics data, and physical parameter constraints) and different ways estimation approaches (including machine learning, regression models, calculations based on rate laws, and model fitting). A comparison to model balancing highlights some advantages and limitations of these methods.

1. **Parameter estimation or optimisation by random sampling.** In theory, parameter fitting and optimisation can be performed by random screening or by Monte-Carlo methods for optimisation, such as genetic algorithms or simulated annealing. For example, one may generate a large ensemble of possible parameter sets, compute for each of them the likelihood or posterior density values, and choose the one that performs best (see [25] for an example). Such optimisation methods are generic and easy to implement, but with large parameter spaces and complicated objective functions the search for optimal solutions becomes highly inefficient. Moreover, without an analytical grasp of the optimality problem, it is hard to assess how good the solutions actually are. Proving an objective function to be convex, as done here, makes numerical problems more transparent. Another question concerns the usage of priors. In sampling kinetic constants, one may employ realistic priors obtained from parameter balancing, which also account for constraints. However, putting priors on state variables as well would be difficult. In model balancing, priors for all variables are directly integrated into the optimality problem. Thus, compared to simple sampling methods, convex model balancing has two advantages: first, the estimation problem is formulated in a transparent way, and second, instead of numerical sampling, possibly with local optima, we directly obtain an optimality problem for the maximum posterior (and posterior sampling can be done, too).

2. **Structural kinetic modelling and elasticity sampling** An alternative method for model parameterisation is Structural Kinetic Modelling (SKM) [26], in which parameters are not fitted but randomly chosen to create model ensembles. A consistent model state is constructed in two steps: first, a metabolic state is defined by choosing fluxes and metabolite levels. Then, kinetic constants are chosen at random, but in agreement with the predefined metabolic state. In practice this is achieved by randomly sampling the saturation values of enzymes and then reconstructing the corresponding kinetic constants. Elasticity sampling [32], a variant of this method, considers reversible rate laws and guarantees thermodynamically consistent results. In the first step, it requires thermodynamically consistent fluxes, metabolite levels, and thermodynamic forces. In the second step, thermodynamic forces are used to convert saturation values into correct reaction elasticities. SKM and elasticity sampling can be adapted to account for priors or data of K_{M} values. However, including data or priors about k_{cat} values and enzyme levels remains difficult, and the method cannot be used to used to match kinetic constants simultaneously to several metabolic states.

3. **Fitting kinetic constants to complete omics data in single reactions** If fluxes, metabolite levels, and enzyme level are known for several steady states, the kinetic constants can be fitted theoretically, reaction by reaction[12] [36, 17]. However, this approach has a number of limitations: for each reaction considered, complete omics data are required; and if kinetic constants are estimated separately for each reaction,

[12] In the SIMMER method [17], a Markov chain Monte Carlo approach is used for the optimisation. The estimation can be reformulated as a model balancing problem, and be solved by convex optimisation.

these constants may violate thermodynamic constraints (unless a safe parameterisation scheme, e.g. with predefined equilibrium constants, is used).

4. **Maximal apparent k_{cat} method** A comparison of model balancing to the "maximal apparent k_{cat}" method showed that model balancing estimates k_{cat} values more reliably, and thus extracts more information from the available data. Of course, the "maximal apparent k_{cat}" method is not expected to work very well if only few metabolic states are considered. But this also holds for model balancing! The problem with model balancing is that the calculations become harder for larger numbers of states, where the "maximal apparent k_{cat}" method remains the method of choice.

Parameter estimation in kinetic models can easily lead to non-convex optimisation. It may be surprising that a simple convex estimation method exists. Model balancing relies on two insights: all fluxes must be predefined[13], and logarithmic kinetic constants and metabolite concentration are the right variables for optimisation[14]. Model balancing builds on two other methods that share the same features and lead to convex optimality problems: Parameter Balancing (PB) for the estimation of kinetic constants and Enzyme Cost Minimisation (ECM) the estimation of optimal metabolic states (see Figure 2).

1. **Parameter balancing.** Parameter balancing is an estimation method to obtain consistent kinetic and thermodynamic constants from kinetic and thermodynamic data. It resembles model balancing, but without detailed information on rate laws and fluxes. All "multiplicative" constants (such as Michaelis-Menten constants or catalytic constants) are described by logarithmic values. To account for parameter dependencies, all other kinetic constants are computed from a subset of kinetic constants[15][16], the free parameters in our linear regression model. With Gaussian priors and measurement errors (on logarithmic scale), likelihood loss and posterior loss terms are quadratic and convex. Parameter balancing can also be applied to kinetic and thermodynamic constants ("kinetic parameter balancing"), to metabolite concentrations and thermodynamic forces in one or more metabolic states ("state balancing"), or to kinetic constants and metabolic states together ("state/parameter balancing"). Known flux directions can be used as additional data, to define the signs of thermodynamic forces. Thus, parameter balancing can predict thermodynamically feasible kinetic constants and metabolite levels and its optimisation takes place on the same set as in model balancing. It provides reasonable ranges for kinetic constants, but in contrast to model balancing it does not consider rate laws or quantitative fluxes[17], and so it cannot be used to fit kinetic constants to metabolite, enzyme, and flux data.

2. **Enzyme cost minimisation.** Enzyme cost minimisation [31] predicts optimal enzyme and metabolite levels in a kinetic model with known parameter values. Unlike parameter balancing, this method uses kinetic rate laws with given kinetic constants, and it is a biological cost, not a fit to data, that is optimised. ECM assumes predefined metabolic fluxes and determines metabolite and enzyme levels that realise these desired fluxes at a minimal cost, where cost functions can be a linear or convex function of the enzyme levels, plus a convex function of the metabolite levels. The optimisation is carried out in (log-)metabolite space. With given rate laws, the enzyme levels can be written as functions of metabolite levels and fluxes and the cost function (scoring enzyme and metabolite levels) is convex on the feasible metabolite polytope.

[13] Measurement errors in metabolic fluxes will distort our estimation results, but model balancing remains applicable, i.e., the estimation problem is still convex. However, fluxes must be thermodynamically consistent, that is, without thermodynamically infeasible flux cycles.

[14] Accordingly, kinetic constants and metabolite concentration must be described with log-normal distributions for measurement errors and priors while enzyme levels must be described on non-logarithmic scale (assuming normal distributions for measurement errors and priors).

[15] Mathematically, parameter balancing resembles the component contribution method, which component contribution method [37] used to determine thermodynamic constants in eQuilibrator [35].

[16] The equilibrium constants were not parameterised by standard chemical potentials μ° (as proposed in [14] for parameter balancing), but by independent equilibrium constants. This is convenient because we use a smaller set of independent variables and avoid non-identifiability (while the standard chemical potentials themselves are not in the centre of interest), and the same choice could be applied in parameter balancing.

[17] As a practical workaround, balanced kinetic constants can be further adjusted to match quantitative fluxes, but this only works if a single metabolic state is considered.

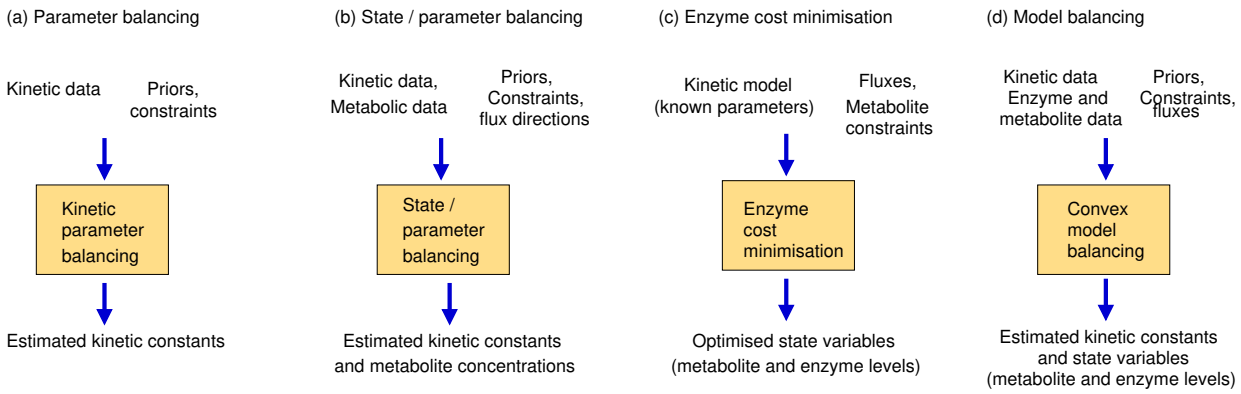

Figure 2: Model balancing and similar methods for parameter estimation and optimal metabolic states. The methods differ in their purpose (parameter estimation versus prediction of biologically optimal states), the choice of free variables (kinetic constants and/or metabolite and enzyme levels), and data used, but they all share some mathematical features: kinetic constants and metabolite levels are described on logarithmic scale (such that all dependencies become linear); thermodynamic and physiological constraints are imposed; and fluxes are predefined. In each of these methods, the search space is a convex polytope and the objective function is convex (either quadratic or derived from kinetics), leading to convex optimality problem.

Model balancing combines elements from both methods. As in parameter balancing, the free variables are log-kinetic constants and log-metabolite levels (forming a feasible parameter/concentration polytope), and the prior and likelihood terms of kinetic and metabolic variables are convex functions. And, as in enzyme cost minimisation, we assume that the fluxes are given and use the fact that the enzyme levels are convex functions of the (logarithmic) metabolite levels. This is combined with two additional insights: it uses the fact that enzyme levels are convex functions in the *combined space* of kinetic and metabolic variables, and the fact that in this space the prior and likelihood terms for enzymes are convex functions just like the enzyme levels themselves.

In all three methods, the feasible region is a high-dimensional polytope (for the vector of logarithmic kinetic constants, metabolite levels, or both). Each dimension refers to one variable, a box is defined by upper and lower bounds, and linear constraints defined by dependencies are added. The feasible polytope for Model Balancing is obtained from the polytopes of the other methods by taking their Cartesian product and removing infeasible regions, in which constraints between kinetic constants and metabolite levels would be violated (shown in Figure 14). Since all variables are estimated at the same time, information about one variable can improve the estimates of other variables. In parameter balancing, a data value for one kinetic constant may improve the estimates of all others. Similarly, in model balancing additional metabolite and enzyme data improve the estimation of all kinetic constants.

Depending on data available, model balancing can be applied in different ways.

1. **Infer a missing data types** Let us assume that data for two of our data types (kinetic constants, metabolite levels, and enzyme levels) are available, while the third type of data is missing. There are three cases: we may estimate *in-vivo* kinetic constants from fluxes, metabolite levels, and enzyme levels; we may estimate metabolite levels from fluxes, enzyme levels, and a kinetic model; or we may estimate enzyme levels from fluxes, metabolite levels, and a kinetic model. If the given data were complete and precise, the third type of variables could be directly computed. But since we assume that the given data are uncertain and incomplete, our aim is to infer the missing data while completing and adjusting the others.

2. **Obtain complete, consistent metabolic states** If all kinetic constants are known, and if metabolite and enzyme have been measured, we can translate these incomplete and uncertain data into consistent and plausible metabolic states. As in all the other cases, fluxes must be given and their directions must agree with the

assumed equilibrium constants and metabolite bounds. Even in the worst case, without any enzyme or metabolite data, we can still guess plausible metabolic states based on fluxes and on the kinetic model and relying on priors for enzyme or metabolite levels.

3. **Ensure thermodynamic constraints and bounds** To obtain a consistent model, we may colect data for kinetic and state variables and translate them into parameters and state variables for our kinetic model. These values will satisfy the rate laws, agree with physical and physiological constraints, and resemble the data and prior values. As in all other cases, posterior sampling could be used to decrease and assess uncertainties about the model parameters.

4. **Sampling from the posterior** Instead of maximising the posterior density, we may sample from the posterior to obtain marginal distributions and covariances of kinetic constants and state variables, and parameter sets can be sampled to obtain a model ensemble. Sampling is facilitated by the fact that the posterior loss function is convex (and thus, the posterior itself has a single mode). To simplify this process, the posterior may be approximated by a multivariate Gaussian distribution, obtained from the posterior mode and the Hessian matrix in this point.

Model balancing can use various types of knowledge (network structure, data, priors, and constraints), handles different types of variables (as defined by the dependence scheme used), and makes relatively few assumptions. For example, many metabolic modelling methods, such as FBA, assume stationary flux distributions. Model balancing does not make this assumption. Like ECM it applies to non-stationary fluxes, e.g. fluxes appearing in dynamic time courses. However, the assumed fluxes must be thermodynamically correct. Here I focused on maximum-posterior estimation. Of course, the posterior can also be sampled (by Monte-Carlo Markov chain methods) or be approximated by a multivariate Gaussian, inside the feasible polytope, defined by the posterior mode and the Hessian matrix in this point. Model balancing extracts information from heterogeneous data. Even if almost no data are available, it can be used to obtain plausible models or model ensembles. In the tests with articificial data, model balancing performed well when precise data were given, and even with imprecise data it performed better than estimation by maximal catalytic rates. Usage of equilibrium constants improves the results, which confirms the importance of known equilibrium constants for constructing reliable kinetic models. Currently, the main limitation seems to be model size, which impacts memory requirements and calculation time (results not shown). Thus, for large models, posterior sampling based on the posterior defined here – may be be the method of choice.

Acknowledgements

I thank Elad Noor for a interesting and enjoyable discussions. Most ideas for this work were developed in the European Commission 7th Framework project BaSysBio (LSHG-CT-2006-037469).

References

[1] B. Teusink, J. Passarge, C.A. Reijenga, E. Esgalhado, C.C. van der Weijden, M. Schepper, M.C. Walsh, B.M. Bakker, K. van Dam, H.V. Westerhoff, and J.L. Snoep. Can yeast glycolysis be understood in terms of in vitro kinetics of the constituent enzymes? Testing biochemistry. *European Journal of Biochemistry*, 267:5313–5329, 2000.

[2] K. Smallbone et al. A model of yeast glycolysis based on a consistent kinetic characterisation of all its enzymes. *FEBS Letters*, 587:2832–2841, 2013.

[3] W. Liebermeister and E. Klipp. Bringing metabolic networks to life: convenience rate law and thermodynamic constraints. *Theor. Biol. Med. Mod.*, 3:41, 2006.

[4] W. Liebermeister, J. Uhlendorf, and E. Klipp. Modular rate laws for enzymatic reactions: thermodynamics, elasticities, and implementation. *Bioinformatics*, 26(12):1528–1534, 2010.

[5] F. Büchel et al. Path2Models: large-scale generation of computational models from biochemical pathway maps. *BMC Systems Biology*, 7:116, 2013.

[6] B. Du, D.C. Zielinski, E.S. Kavvas, A. Dräger, J. Tan, Z. Zhang, K.E. Ruggiero, G.A. Arzumanyan, and B.Ø. Palsson. Evaluation of rate law approximations in bottom-up kinetic models of metabolism. *BMC Systems Biology*, 10(40), 2016.

[7] I. Schomburg, A. Chang, C. Ebeling, M. Gremse, C. Heldt, G. Huhn, and D. Schomburg. BRENDA, the enzyme database: updates and major new developments. *Nucleic Acids Research*, 32:D431–433, 2004.

[8] A. Bar-Even, E. Noor, Y. Savir, W. Liebermeister, D. Davidi, D.S. Tawfik, and R. Milo. The moderately efficient enzyme: evolutionary and physicochemical trends shaping enzyme parameters. *Biochemistry*, 21:4402–4410, 2011.

[9] D. Heckmann, C.J. Lloyd, N. Mih, Y. Ha, D.C. Zielinski, Z.B. Haiman, A. Amer Desouki, M.J. Lercher, and B.Ø. Palsson. Machine learning applied to enzyme turnover numbers reveals protein structural correlates and improves metabolic models. *Nature Communications*, 9:5252, 2018.

[10] W. Liebermeister and E. Klipp. Bringing metabolic networks to life: integration of kinetic, metabolic, and proteomic data. *Theor. Biol. Med. Mod.*, 3:42, 2006.

[11] T. Lubitz, M. Schulz, E. Klipp, and W. Liebermeister. Parameter balancing for kinetic models of cell metabolism. *J. Phys. Chem. B*, 114(49):16298–16303, 2010.

[12] P. Saa and L.K. Nielsen. A general framework for thermodynamically consistent parameterization and efficient sampling of enzymatic reactions. *PLoS Computational Biology*, 11(4):e1004195, 2015.

[13] J.C. Mason and M.W. Covert. An energetic reformulation of kinetic rate laws enables scalable parameter estimation for biochemical networks. *Journal of Theoretical Biology*, 461:145–156, 2019.

[14] T. Lubitz and W. Liebermeister. Parameter balancing: consistent parameter sets for kinetic metabolic models. *Bioinformatics*, 35:3857–3858, 2019.

[15] N.J. Stanford, T. Lubitz, K. Smallbone, E. Klipp, P. Mendes, and W. Liebermeister. Systematic construction of kinetic models from genome-scale metabolic networks. *PLoS ONE*, 8(11):e79195, 2013.

[16] D. Davidi, E. Noor, W. Liebermeister, A. Bar-Even, A. Flamholz, K. Tummler, U. Barenholz, M. Goldenfeld, T. Shlomi, and R. Milo. Global characterization of in vivo enzyme catalytic rates and their correspondence to in vitro k_{cat} measurements. *PNAS*, 113(12):3401–3406, 2016.

[17] S.R. Hackett, V.R.T. Zanotelli, W. Xu, J. Goya, J.O. Park, D.H. Perlman, P.A. Gibney, D. Botstein, J.D. Storey, and J.D. Rabinowitz. Systems-level analysis of mechanisms regulating yeast metabolic flux. *Science*, 354(6311):aaf2786, 2016.

[18] M. Ashyraliyev, Y. Fomekong-Nanfack, J.A. Kaandorp, and J.G. Blom. Systems biology: parameter estimation for biochemical models. *FEBS Journal*, 276(4):886–902, 2009.

[19] A.F. Villaverde, F. Froehlich, D. Weindl, J. Hasenauer, and J.R. Banga. Benchmarking optimization methods for parameter estimation in large kinetic models. *Bioinformatics*, 35:830–838, 2018.

[20] S. Srinivasan, W.R. Cluett, and R. Mahadevan. A scalable method for parameter identification in kinetic models of metabolism using steady state data. *Bioinformatics*, page btz445, 2019.

[21] N. Jamshidi and B.Ø. Palsson. Formulating genome-scale kinetic models in the post-genome era. *Molecular Systems Biology*, 4:171, 2008.

[22] A. Khodayari and C.D. Maranas. A genome-scale Escherichia coli kinetic metabolic model k-ecoli457 satisfying flux data for multiple mutant strains. *Nature Communications*, 7(13806), 2016.

[23] R.W. Smith, R.P. van Rosmalen, V.A.P. Martins dos Santos, and C. Fleck. DMPy: a Python package for automated mathematical model construction of large-scale metabolic systems. *BMC Systems Biology*, 12:72, 2018.

[24] C.J. Foster, S. Gopalakrishnan, M.R. Antoniewicz, and C.D. Maranas. From Escherichia coli mutant ^{13}C labeling data to a core kinetic model: A kinetic model parameterization pipeline. *PLoS Computational Biology*, 15(9):e1007319, 2019.

[25] D. Christodoulou, H. Link, T. Fuhrer, K. Kochanowski, L. Gerosa, and U. Sauer. Reserve flux capacity in the pentose phosphate pathway enables Escherichia colis rapid response to oxidative stress. *Cell Systems*, 6:569578, 2018.

[26] R. Steuer, T. Gross, J. Selbig, and B. Blasius. Structural kinetic modeling of metabolic networks. *PNAS*, 103(32):11868–11873, 2006.

[27] W. Liebermeister. Predicting physiological concentrations of metabolites from their molecular structure. *J. Comp. Biol.*, 12(10):1307–1315, 2005.

[28] A. Bar-Even, E. Noor, A. Flamholz, J.M. Buescher, and R. Milo. Hydrophobicity and charge shape cellular metabolite concentrations. *PLoS Computational Biology*, 7(10):e1002166, 2011.

[29] N. Tepper, E. Noor, D. Amador-Noguez, H.S. Haraldsdóttir, R. Milo, J. Rabinowitz, W. Liebermeister, and T. Shlomi. Steady-state metabolite concentrations reflect a balance between maximizing enzyme efficiency and minimizing total metabolite load. *PLoS ONE*, 8(9):e75370, 2013.

[30] A. Flamholz, E. Noor, A. Bar-Even, W. Liebermeister, and R. Milo. Glycolytic strategy as a tradeoff between energy yield and protein cost. *PNAS*, 110(24):10039–10044, 2013.

[31] E. Noor, A. Flamholz, A. Bar-Even, D. Davidi, R. Milo, and W. Liebermeister. The protein cost of metabolic fluxes: prediction from enzymatic rate laws and cost minimization. *PLoS Computational Biology*, 12(10):e1005167, 2016.

[32] W. Liebermeister. Elasticity sampling links thermodynamics to metabolic control. *Preprint on arXiv.org: arXiv:1309.0267*, 2013.

[33] E. Noor, A. Flamholz, W. Liebermeister, A. Bar-Even, and R. Milo. A note on the kinetics of enzyme action: a decomposition that highlights thermodynamic effects. *FEBS Letters*, 587(17):2772–2777, 2013.

[34] A. Gelman, J. B. Carlin, H. S. Stern, and D.B. Rubin. *Bayesian Data Analysis*. Chapman & Hall, New York, 1997.

[35] A. Flamholz, E. Noor, A. Bar-Even, and R. Milo. eQuilibrator – the biochemical thermodynamics calculator. *Nucleic Acids Research*, 40(D1):D770–D775, 2012.

[36] J. Bruck, W. Liebermeister, and E. Klipp. Exploring the effect of variable enzyme concentrations in a kinetic model of yeast glycolysis. *Genome Informatics Series*, 20, 2008.

[37] E. Noor, H.S. Haraldsdottir, R. Milo, and R.M.T. Fleming. Consistent estimation of Gibbs energy using component contributions. *PLoS Computational Biology*, 9:e1003098, 2013.

[38] T. Lubitz, J. Hahn, F.T. Bergmann, E. Noor, E. Klipp, and W. Liebermeister. SBtab: A flexible table format for data exchange in systems biology. *Bioinformatics*, 32(16):25592561, 2016.

[39] 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.J. Hodgman, J.H. Hofmeyr, P.J. Hunter, N.S. Juty, J.L. Kasberger, A. Kremling, U. Kummer, N. Le Novère, L.M. Loew, D. Lucio, P. Mendes, E. Minch, E.D. Mjolsness, Y. Nakayama, M.R. Nelson, P.F. Nielsen, T. Sakurada T J.C. Schaff, B.E. Shapiro, T.S. Shimizu, H.D. Spence, J. Stelling, K. Takahashi, M. Tomita, J. Wagner, J. Wang, and the SBML Forum. The Systems Biology Markup Language (SBML): A medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531, 2003.

[40] B.R.B.H. van Rijsewijk, A. Nanchen, S. Nallet, R.J. Kleijn, and U. Sauer. Large-scale 13c-flux analysis reveals distinct transcriptional control of respiratory and fermentative metabolism in escherichia coli. *Mol. Syst. Biol.*, 7(477):477, 2011.

[41] A. Schmidt, K. Kochanowski, S. Vedelaar, E. Ahrné, B. Volkmer, L. Callipo, K. Knoops, M. Bauer, R. Aebersold, and M. Heinemann. The quantitative and condition-dependent Escherichia coli proteome. *Nature Biotechnology*, doi:10.1038/nbt.3418, 2015.

[42] L. Gerosa, B.R.B.H. van Rijsewijk, D. Christodoulou, K. Kochanowski, T.S.B. Schmidt, E. Noor, and U. Sauer. Pseudo-transition analysis identifies the governing regulation of microbial nutrient adaptations from steady state data. *Cell Systems*, 1:270–282, 2015.

E. coli model with artificial data (noise-free kinetic data, noise-free metabolic data)

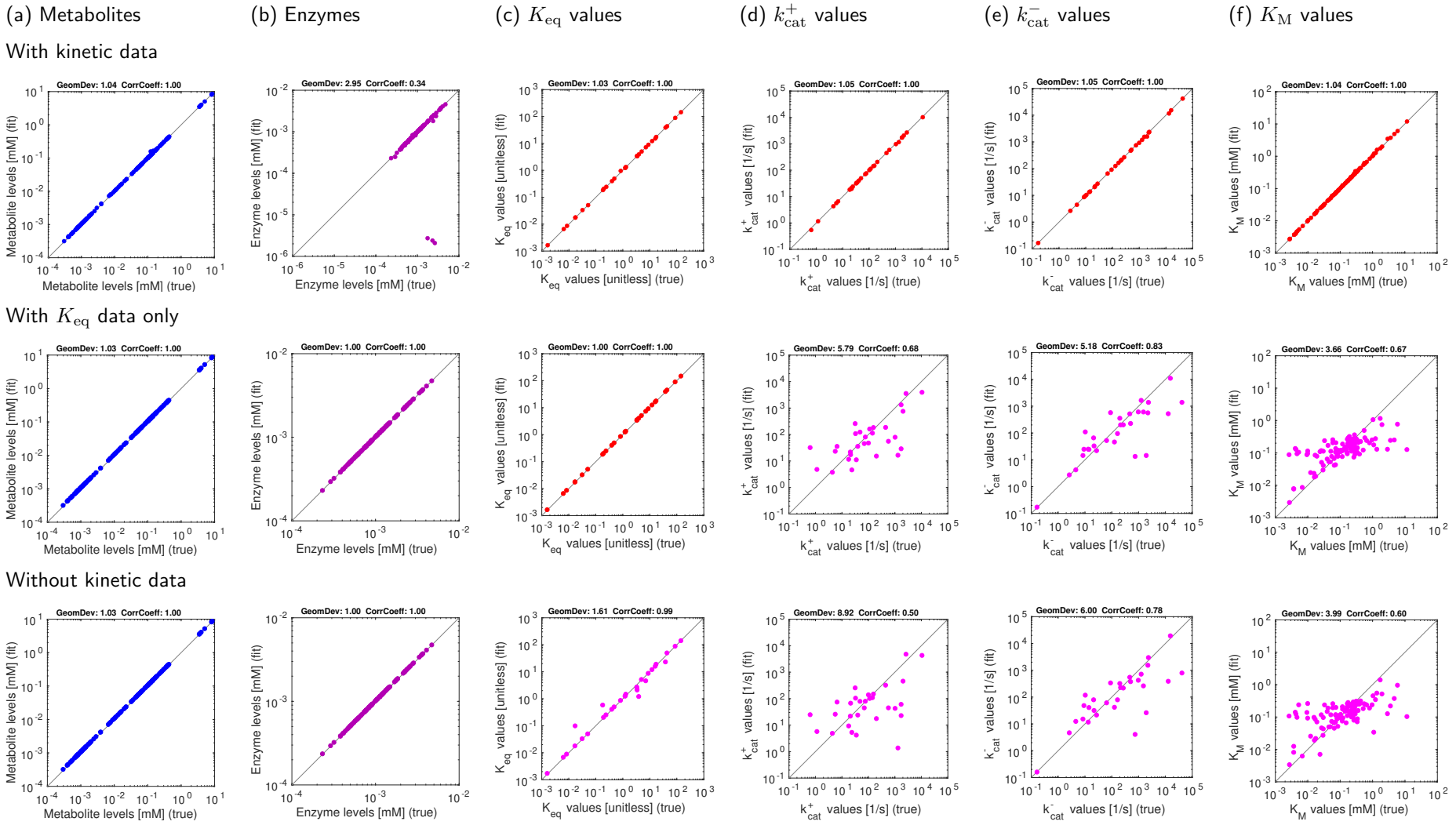

Figure 3: Model balancing results for *E. coli* central metabolism model with artificial data. The model structure is shown in Figure 16. Each subfigure shows "true" values (x-axis) versus reconstructed values (y-axis). Similarities are quntified by geometric standard deviations ("GeomDev") and Pearson correlation coefficients ("CorrCoeff"). (a) Metabolite levels. (b) Enzyme levels. (c)-(f) Different types of kinetic constants. Rows show different estimation scenarios (see Figure) Upper row: simple scenario S1 (noise-free artificial data, data for kinetic constants). Centre row: scenario S1K (noise-free artificial data, kinetic data given only for equilibrium constants). Lower row: scenario S2 (noise-free artificial data, no data for kinetic constants). Depending on the scenario, kinetic constants are either fitted (red dots) or predicted (magenta dots).

E. coli model with artificial data (noisy kinetic data, noise-free metabolic data)

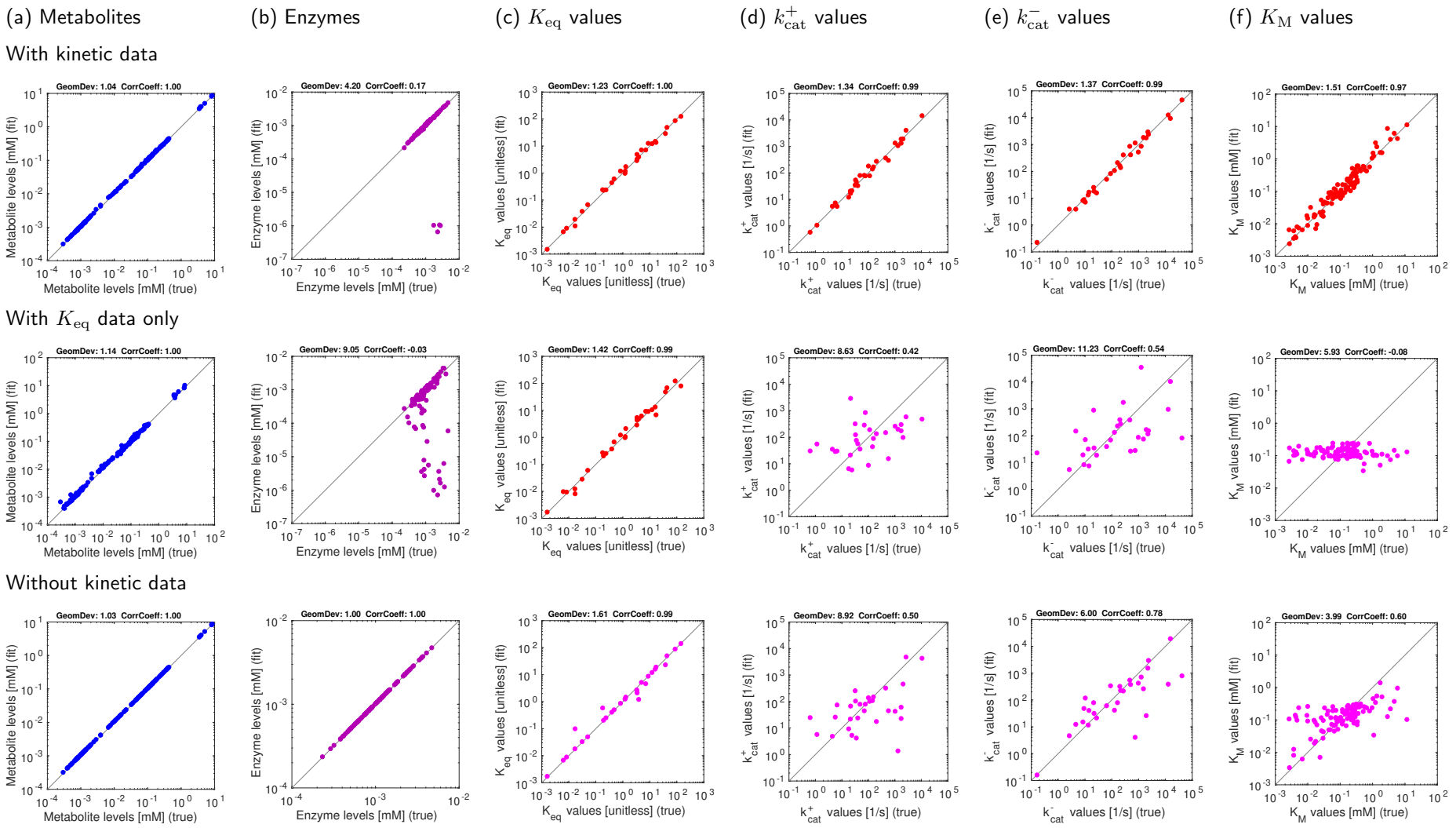

Figure 4: Same as Figure 3, with noisy kinetic data

E. coli model with artificial data (noise-free kinetic data, noisy metabolic data)

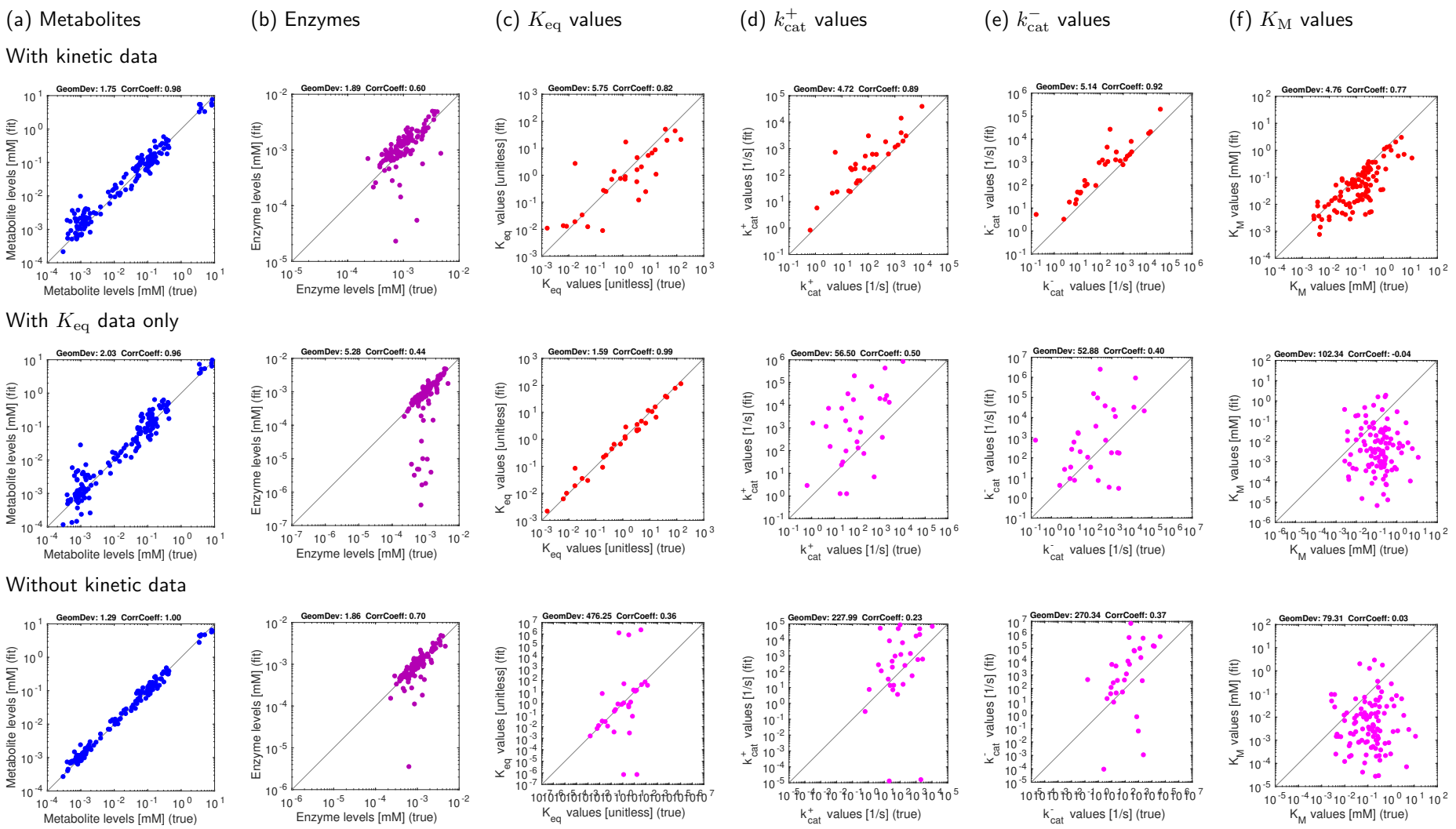

Figure 5: Results for *E. coli* central metabolism with noisy artificial data. Top row: estimation scenario S3 (noisy artificial data, data used for kinetic constants). Centre row: estimation scenario S3K (noisy artificial data, data for equilibrium constants only). Bottom row: estimation scenario S4 (noisy artificial data, no data for kinetic constants).

E. coli model with artificial data (noisy kinetic data, noisy metabolic data)

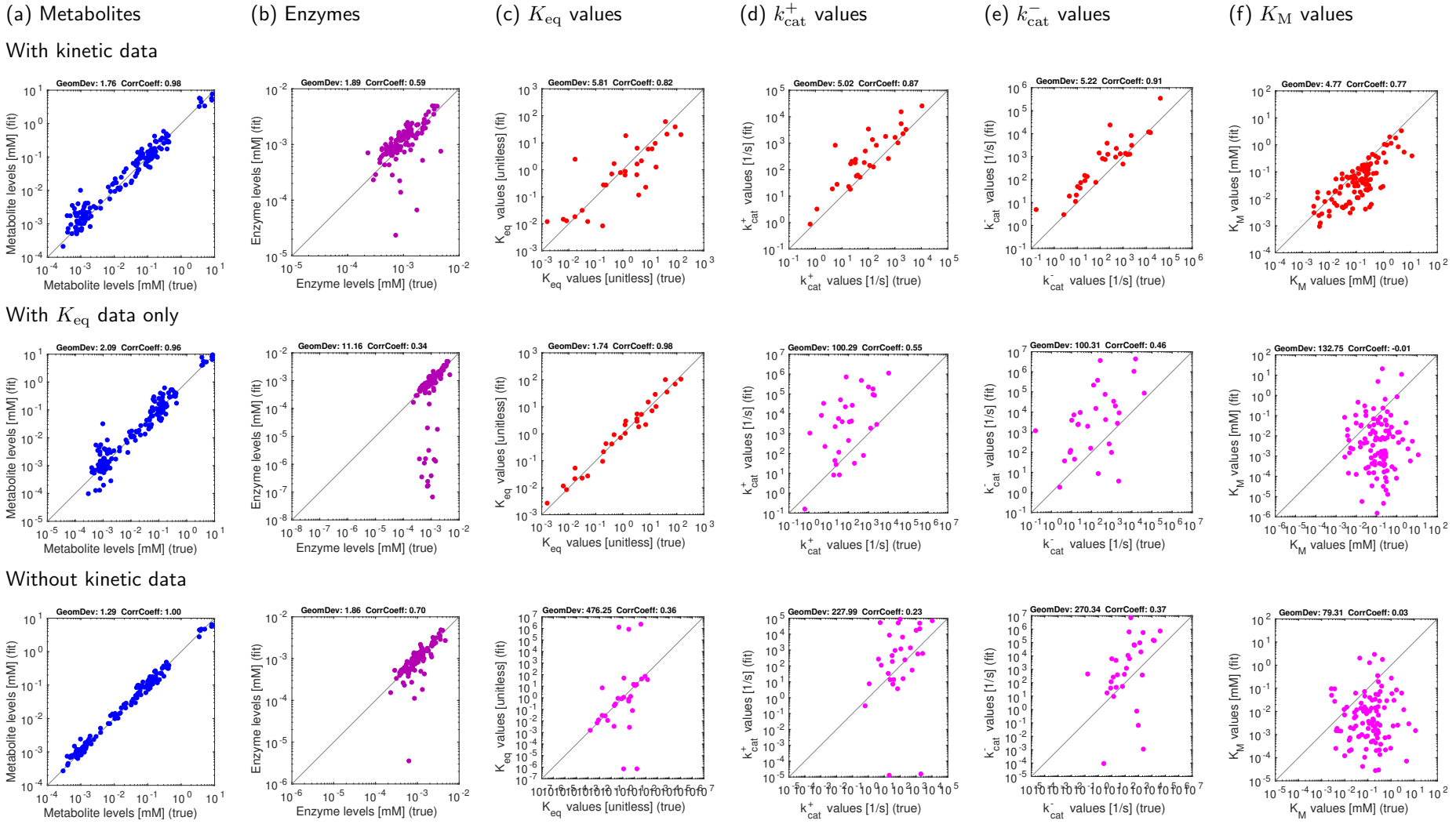

Figure 6: Same as Figure 5, with noisy kinetic data

Figure 7: Catalytic constants in *E. coli* central metabolism (artificial data), estimated by maximal apparent catalytic rates [16]. Note that k_{cat} values can only be estimated in the direction of fluxes (e.g. k^+_{cat} for reactions with forward fluxe).

E. coli model (aerobic growth on glucose), balanced kinetic data

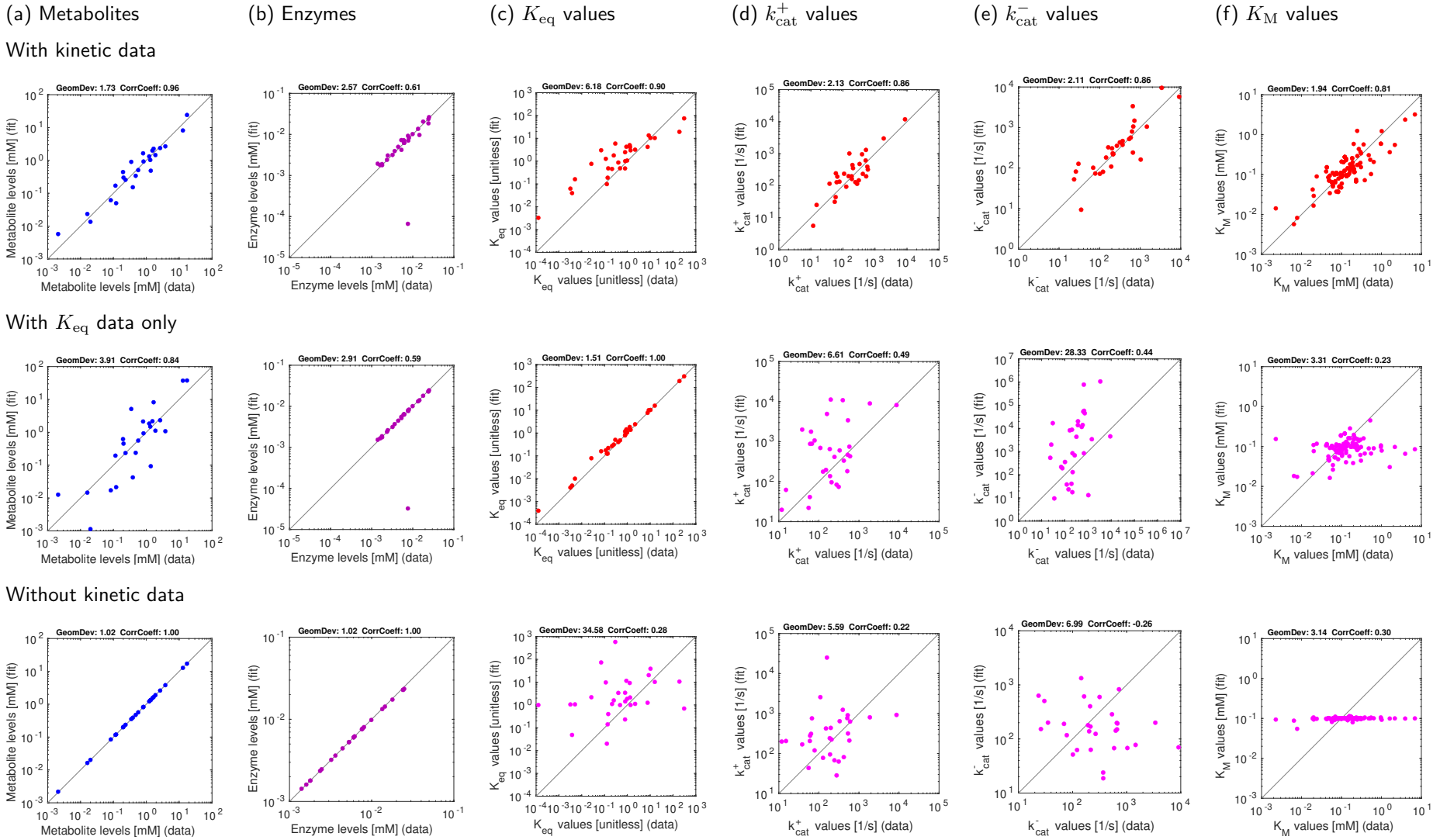

Figure 8: Results for *E. coli* central metabolism with experimental data (aerobic growth on glucose). The kinetic data stem from previous parameter balancing based on *in-vitro* data. Top: estimation using kinetic data. Centre: estimation using equilibrium constants as the only kinetic data. Centre: estimation using equilibrium constants as the only kinetic data. Bottom: estimation without usage of kinetic data. The same metabolite, enzyme, and kinetic data were used in [31].

E. coli central metabolism model (aerobic growth on glucose), in-vitro kinetic data

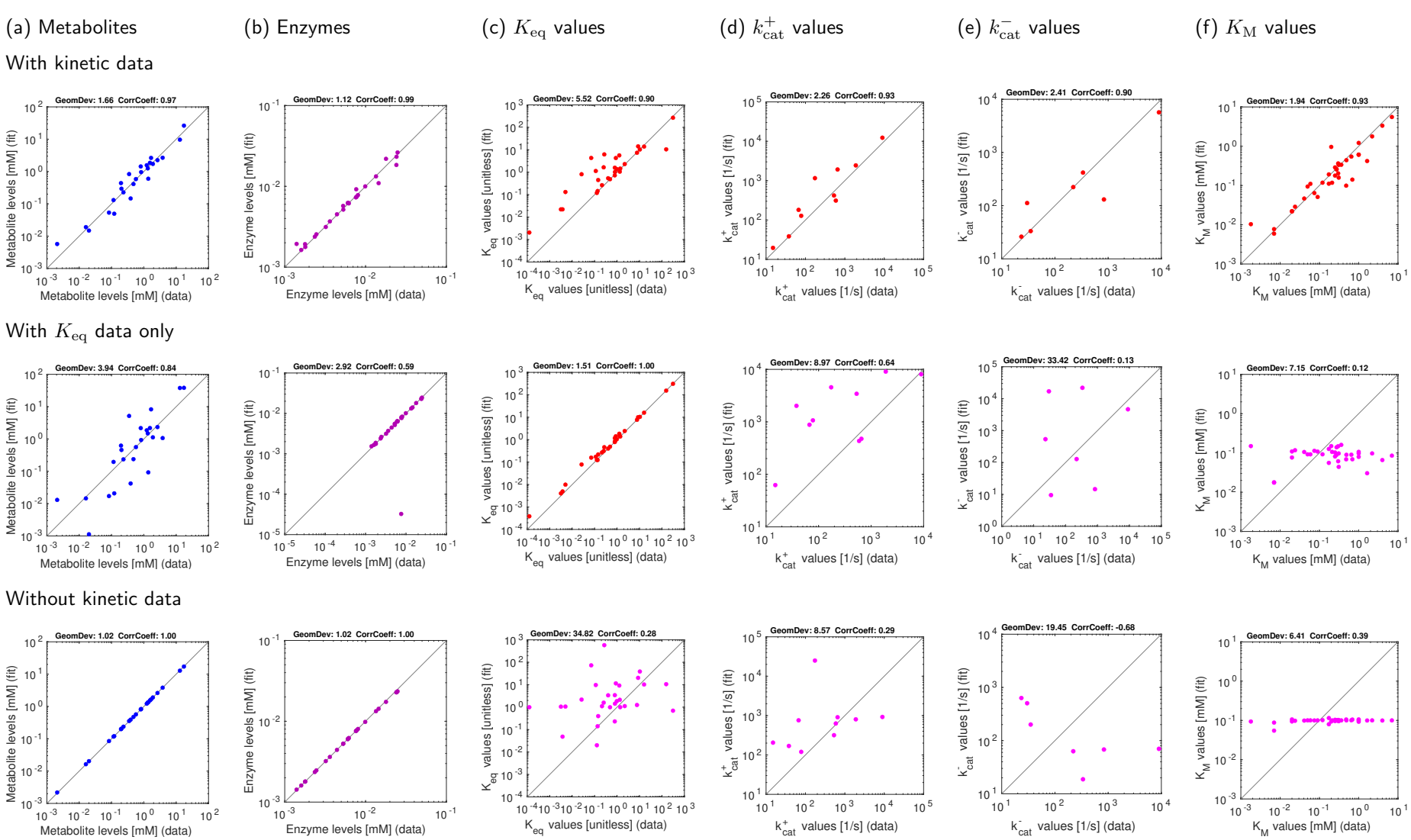

Figure 9: Results for *E. coli* central metabolism with experimental data (aerobic growth on glucose). Same as Figure 8, but based on original kinetic *in-vitro* data instead of balanced kinetic data.

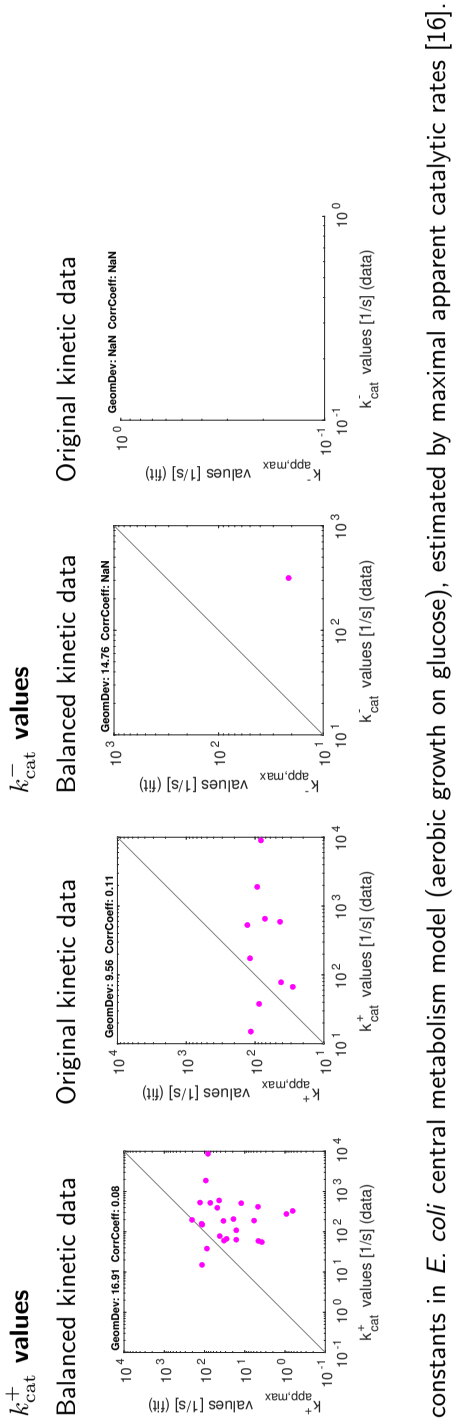

Figure 10: Catalytic constants in *E. coli* central metabolism model (aerobic growth on glucose), estimated by maximal apparent catalytic rates [16].

E. coli central metabolism model, three conditions (glucose, glycerol, acetate), kinetic data balanced

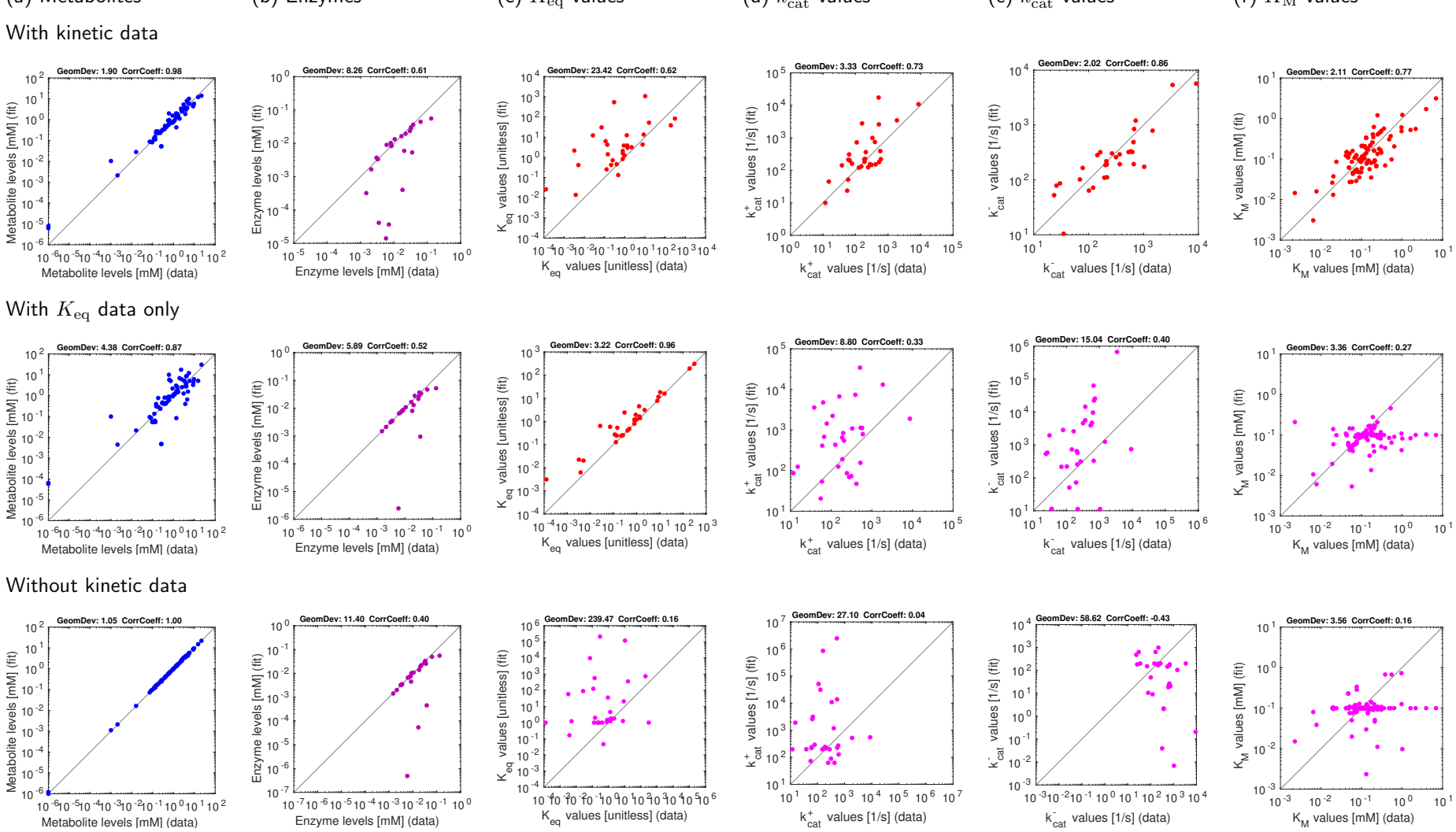

Figure 11: Results for *E. coli* central metabolism with experimental data (aerobic growth on glucose, glycerol, or acetate). Balanced kinetic data used. Top: estimation with kinetic data used. Centre: estimation using equilibrium constants as the only kinetic data. Centre: estimation using equilibrium constants as the only kinetic data. Bottom: estimation without usage of kinetic data.

E. coli central metabolism model, three conditions (glucose, glycerol, acetate), in-vitro kinetic data

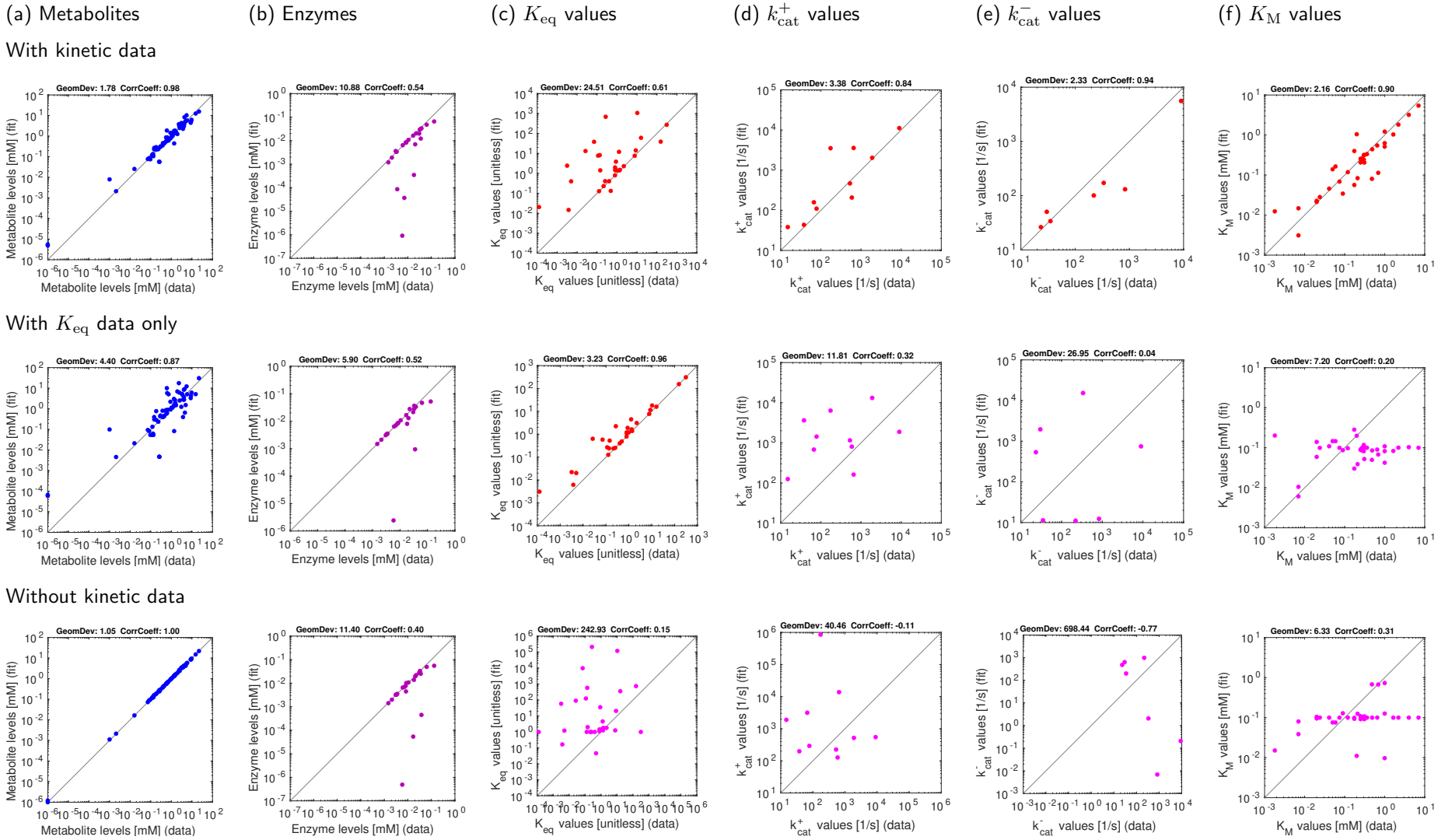

Figure 12: Results for *E. coli* central metabolism with experimental data (aerobic growth on glucose, glycerol, or acetate). Original kinetic data used. Top: estimation with kinetic data used. Centre: estimation using equilibrium constants as the only kinetic data. Centre: estimation using equilibrium constants as the only kinetic data. Bottom: estimation without usage of kinetic data.

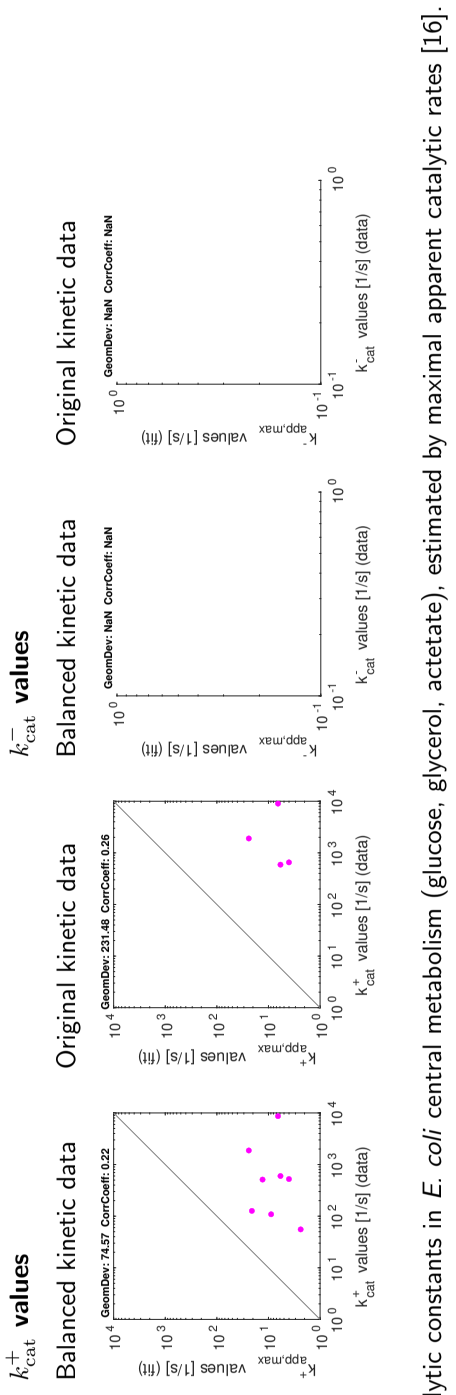

Figure 13: Catalytic constants in *E. coli* central metabolism (glucose, glycerol, actetate), estimated by maximal apparent catalytic rates [16].

A The model balancing problem

A.1 Model variables and constraints

To define a model balancing problem, we need to consider all model parameters and state variables (as "model variables") and figure out their dependencies. We split the model variables into "independent" (or "free") variables and "dependent" variables based on the following thoughts. (i) To describe dependencies between kinetic constants, we treat some of them as free variables (independent log-equilibrium constants, log-Michaelis-Menten constants, and log-velocity constants), while all others are linearly dependent on them (dependent log-equilibrium constants, log-catalytic constants). (ii) For each metabolic state, we consider a metabolite log-concentration vector, an enzyme concentration vector, and a flux vector. Vectors from different metabolic states (usually given as columns of a matrix) are concatenated into a large vector. (iii) Since enzyme levels follow from kinetic constants, metabolite levels, and fluxes, they are treated as dependent variables. (iv) Thermodynamic driving forces follow from equilibrium constants and metabolite concentrations, and are therefore dependent variables. The kinetic constants and metabolite levels remain the only free variables. (v) The predefined flux directions determine the signs of driving forces, implying linear constraints between logarithmic equilibrium constants and metabolite concentrations. Altogether, we obtain the following variables and dependencies (see Figure 1 (b)).

1. **Independent variables** Our free variables comprise (i) the independent kinetic constants on logarithmic scale (independent equilibrium constants $\ln K_{\mathrm{eq}}^{\mathrm{ind}}$, Michaelis-Menten constants $\ln K_{\mathrm{M}}$, allosteric activation constants $\ln K_{\mathrm{A}}$, allosteric inhibition constants $\ln K_{\mathrm{I}}$, and velocity constants $\ln K_{\mathrm{V}}$), collected in a vector

$$\mathbf{q}^{\mathrm{ind}} = \begin{pmatrix} \ln \mathbf{k}_{\mathrm{eq}}^{\mathrm{ind}} \\ \ln \mathbf{k}_{\mathrm{V}} \\ \ln \mathbf{k}_{\mathrm{M}} \\ \ln \mathbf{k}_{\mathrm{A}} \\ \ln \mathbf{k}_{\mathrm{I}} \end{pmatrix}, \tag{7}$$

   and (ii) the metabolite log-concentrations from one or more metabolic states s, contained in metabolite vectors $\mathbf{x}^{(s)} = \ln \mathbf{c}^{(s)}$. We obtain a vector of free variables

$$\mathbf{y} = \begin{pmatrix} \mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \\ .. \\ \mathbf{q}^{\mathrm{ind}} \end{pmatrix}. \tag{8}$$

   With n_p independent kinetic constants, n_m metabolites, and n_s metabolic states, the vector has the length $n_p + n_m\, n_s$.

2. **Dependent variables** We consider three types of dependent variables: dependent kinetic constants, enzyme concentrations, and thermodynamic forces. (i) The dependent kinetic constants on logarithmic scale, (dependent equilibrium constants $\ln K_{\mathrm{eq}}^{\mathrm{dep}}$, forward catalytic constants $\ln k_{\mathrm{cat}}^{+}$), and reverse catalytic constants $\ln k_{\mathrm{cat}}^{-}$), form in a vector

$$\mathbf{q}^{\mathrm{dep}} = \begin{pmatrix} \ln \mathbf{k}_{\mathrm{eq}}^{\mathrm{dep}} \\ \ln \mathbf{k}_{\mathrm{cat}}^{+} \\ \ln \mathbf{k}_{\mathrm{cat}}^{-} \end{pmatrix}. \tag{9}$$

   This vector can be computed from $\mathbf{q}^{\mathrm{ind}}$ by a linear function $\mathbf{q}^{\mathrm{dep}} = \mathbf{M}^{\mathrm{dep}}\, \mathbf{q}^{\mathrm{ind}}$. The dependency matrix $\mathbf{M}$ follows from the stroichiometric matrix as described in [11]. Similarly, the vector $\mathbf{q}$ of all kinetic constants is

given by the linear formula

$$\mathbf{q} = \begin{pmatrix} \mathbf{q}^{\mathrm{ind}} \\ \mathbf{q}^{\mathrm{dep}} \end{pmatrix} = \begin{pmatrix} \mathbf{I} \\ \mathbf{M}_{\mathrm{ind}}^{\mathrm{dep}} \end{pmatrix} \mathbf{q}^{\mathrm{ind}} = \mathbf{M}_{\mathrm{ind}}^{\mathrm{all}}\, \mathbf{q}^{\mathrm{ind}}. \tag{10}$$

(ii) The thermodynamic forces are computed by the linear formula

$$\boldsymbol{\theta}^{(s)} = \ln \mathbf{k}_{\mathrm{eq}} - \mathbf{N}^{\top}\, \mathbf{x}^{(s)} \tag{11}$$

or briefly $\boldsymbol{\theta} = \mathbf{M}^{\theta}\, \mathbf{y}$ with a matrix $\mathbf{M}^{\theta}$ obtained from the network structure. (iii) Based on rate laws and using Eq. (1), the enzyme concentration vectors $\mathbf{e}^{(s)}$ are given by

$$e_l^{(s)} = \frac{v_l^{(s)}}{k_l(\mathbf{q}, \mathbf{x}^{(s)})}. \tag{12}$$

3. **Feasible region** The feasible region for our free variables is defined by two types of constraints. First, lower and upper bounds on all variables aside from enzyme levels[18]

$$\mathbf{q}^{\min} \le \mathbf{q} \le \mathbf{q}^{\max}, \qquad \mathbf{x}^{\min} \le \mathbf{x}^{(s)} \le \mathbf{x}^{\max}, \qquad \boldsymbol{\theta}^{\min} \le \boldsymbol{\theta}^{(s)} \le \boldsymbol{\theta}^{\max}, \tag{13}$$

where s denotes metabolic states. Second, the driving forces must be positive along the fluxes, and the given flux directions define the signs of all driving forces. For all reactions with non-zero fluxes $v_l^{(s)} \neq 0$, this yields the thermodynamic constraints

$$v_l^{(s)}\, \theta_l^{(s)} > 0, \tag{14}$$

which translate into linear constraints for the variable vector $\mathbf{y}$. In reactions with zero flux, driving forces are unconstrained (unless for some reasons reactions are assumed to be in chemical equilibrium). Together these constraints can be written as

$$\mathbf{A}\, \mathbf{y} \le \mathbf{b}, \tag{15}$$

with a matrix $\mathbf{A}$ and a vector $\mathbf{b}$ obtained from reaction stoichiometries and the flux directions. These constraints define a convex feasible polytope $\mathcal{P}$. Each polytope point defines a feasible vector $\mathbf{y}$, i.e. a feasible set of model parameters and metabolic states (i.e. states with positive forward driving forces). Conversely, any feasible set of kinetic constants and metabolic states (respecting all bounds) corresponds to a point in the polytope.

4. **Priors and likelihood terms** The posterior is obtained from prior and likelihood terms. For metabolite levels, we assume uncorrelated normal priors for the values $x_i^{(s)}$ (i.e. log-normal priors for concentrations). The data values $x_{i,\mathrm{data}}^{(s)}$, appearing in the likelihood, are assumed to be independent and normally distributed. For the absolute enzyme levels, we assume *normal*, independent priors and data values. For logarithmic kinetic constants, we assume normally distributed data values. For the independent kinetic constants, we use a correlated prior, obtained from a prior term for each independent kinetic constant and from pseudo values for dependent kinetic constants. Formally, pseudovalues are invoked to define a correlated prior, but in practice they are treated like additional data points (see [11]).

In contrast to similar modelling methods (Parameter Balancing and ECM), model balancing determines $\mathbf{q}$ and $\mathbf{x}$ at the same time. The resulting vector $\mathbf{y}$ lives in a high-dimensional polytope whose geometric structure is

[18] Positivity is ensure by the other formulae. With thermodynamically feasible rate laws, the enzyme levels $e_l^{(s)}(\mathbf{q}, \mathbf{x}^{(s)})$ for active reactions, Eq. (12), are positive and convex on the entire polytope $\mathcal{P}$ (see appendix B.1).

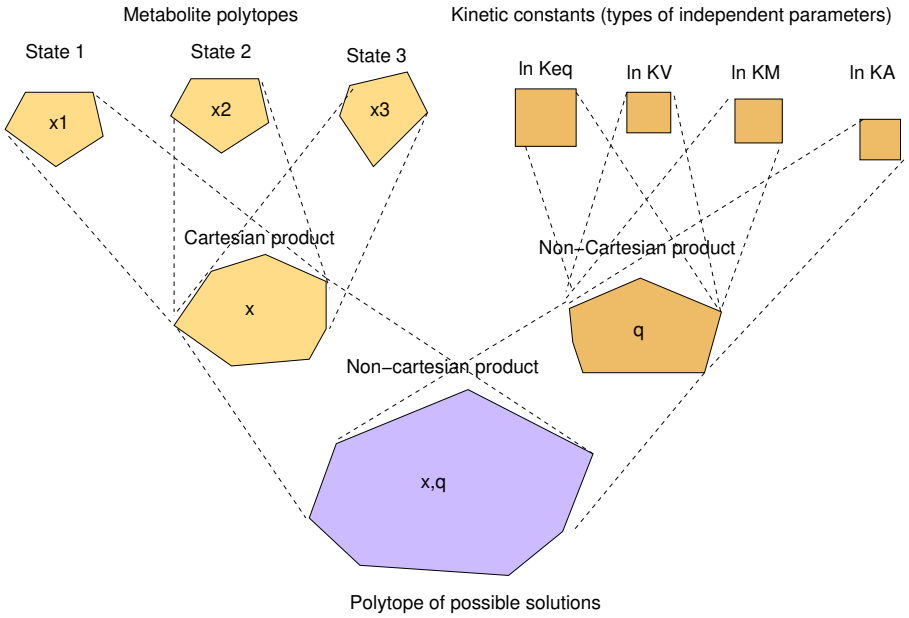

Figure 14: Search space used in model balancing. The free model variables (metabolite levels and kinetic constants, all on logarithmic scale) are constrained by physiological ranges and thermodynamic constraints, dependent on flux directions. Together, these inequality constraints define a feasible region in the space of logarithmic variables (bottom).This high-dimensional polytope arises from a "non-Cartesian" product between a metabolite polytope and a kinetic constant polytope (centre), a Cartesian product from which some parts are removed due to constraints. The metabolite polytope itself is a Cartesian product of the metabolite polytopes for single metabolic states; the kinetic constant polytope is a (non-Cartesian) product of polytopes (boxes) for the different types of kinetic constants (top).

schematically shown in Figure 14. Since each state vector $\mathbf{y}$ consists of a vector $\mathbf{q}$ and a number of vectors $\mathbf{x}^s$, the polytope resembles a Cartesian product of the polytopes for these single vectors. However, thermodynamic constraints between kinetic constants and metabolite levels require that some parts of this Cartesian product must be removed.

To see how the metabolite spaces for several states are combined, let us return to our simplified model balancing problem from section 2.3. We can solve this problem separately for each of the states, and this is in fact the easiest thing to do. But we can also fit all metabolic states simultaneously by one big regression model, combining all metabolite profiles $\mathbf{x}^{(s)}$. Each of these profiles must lie in a metabolite polytope $\mathcal{P}_{\mathbf{x}}^{(s)}$, and if the flux directions in all metabolic states are the same, these polytopes are identical. In contrast, if fluxes change their directions, the metabolite polytopes $\mathcal{P}_{\mathbf{x}}^{(s)}$ will differ. If we merge all vectors $\mathbf{x}^{(s)}$ into a vector $\mathbf{x}$, the feasible polytope for this vector will be higher-dimensional and will be given by the Cartesian product $\prod_s \mathcal{P}_{\mathbf{x}}^{(s)}$. As before, we can consider the prior, likelihood, and posterior (for all metabolic states) as functions on this higher-dimensional polytope, and the problem remains strictly convex. Since the metabolic states are independent, the prior, likelihood, and posterior functions can be split into products of priors, likelihoods, and posteriors for the single states, confirming again that the estimation problems can be separately solved.

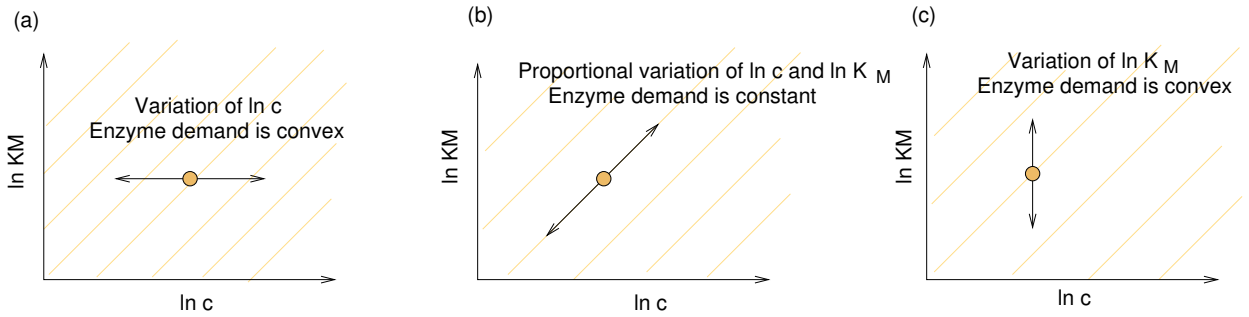

Figure 15: If enzyme demand is convex in in log metabolite levels, it is also convex in the log kinetic constants. The graphics illustrates this by showing variations of model variables (logarithmic kinetic constants and metabolite levels) and their effects on enzyme demand (symbolised by contour lines). (a) The enzyme demand (for each reaction and each metabolic state, at given fluxes) is convex in the (logarithmic) metabolite levels (proof in [31]). (b) A variation of a K_{M} value will change the enzyme demand, but since K_{M} values always appear in term of the form c/K_{M}, this change can be compensated by also varying the corresponding metabolite level, and can therefore also be mimicked by an opposite variation of this metabolite level. (c) It follows that the enzyme demand is convex in $\ln K_{\mathrm{M}}$, and is therefore a convex function in the space of $\ln c$ and $\ln K_{\mathrm{M}}$.

B Convexity proof

B.1 The reciprocal catalytic rate is a convex function of log-metabolite levels, K_{M} values, and k_{cat} values

In our models, we assume reaction rates of the form $v_l = e_l\, k_l$, with catalytic rates k_l depending on metabolite concentrations c_i and kinetic constants (in a vector $\mathbf{p}$, containing all forward and reverse catalytic constants $k^{\pm}_{\mathrm{cat,l}}$, Michaelis-Menten constants $K_{M,li}$, and possibly activation and inhibition constants K_{A} and K_{I}). In particular, we assume that enzyme kinetics k_l follow modular rate laws (which ar so general that this means hardly any restriction):

$$k_l = \frac{k^{+}_{\mathrm{cat,l}} \prod_j (c_i/K_{M,li})^{m^{\mathrm{S}}_{li}} - k^{-}_{\mathrm{cat,l}} \prod_j (c_i/K_{M,li})^{m^{\mathrm{P}}_{li}}}{D_l(\mathbf{c}, \mathbf{k}_{\mathrm{M}})} \tag{16}$$

with the molecularities m_{li}. The denominator D depend on the rate law chosen and must be a polynomial with positive prefactors (or "posinomial"), consisting of terms of the shape $c_i/K_{M,li}$ and possibly $c_i/K_{I,li}$ or $K_{A,li}/c_i$.

Proposition 1 *(Reciprocal rate laws are convex in the logarithmic metabolic concentrations and kinetic constants) For all rate laws of the form 16, the reciprocal catalytic rate $1/k_l$ is a convex function of the logarithmic metabolite concentrations $\ln c_i$, the logarithmic Michaelis-Menten constants $\ln K_{M,li}$, and the logarithmic catalytic constants $\ln k^{\pm}_{\mathrm{cat,l}}$.*

Corollary: Since the logarithmic kinetic constants are related by linear dependencies, the reciprocal catalytic rate $1/k_l$ is also a convex function of the metabolite log-concentrations $\ln c_i$ and the logarithmic *independent* kinetic constants considered in model balancing.

Proof (alternative 1) For this proof, we note that $1/k(\mathbf{x})$ is convex in $\mathbf{x}$ if the kinetic constants are fixed and if $\mathbf{x}$ is restricted to the feasible metabolite polytope given these kinetic constants and the predefined flux direction. This has been shown in [31]. Moreover, we note that in the rate laws considered, concentrations and kinetic constants always appear in the form of product terms (e.g. $k^{+}_{\mathrm{cat}} \cdot c/K_{\mathrm{M}}$). On log-scale, these terms are sums (e.g. $\ln k^{+}_{\mathrm{cat}} + \ln c - \ln K_{\mathrm{M}}$). Therefore, if changes of logarithmic concentrations have a certain effect (namel a "convex" variation of 1/r), then changes of logarithmic kinetic constants should the same type of effects (see

Figure 15). To see this in detail, we first show that $1/k$ is convex in the combined space of $\mathbf{x} = \ln \mathbf{c}$ (relevant metabolites) and $\mathbf{q}_\mathrm{M} = \ln K_\mathrm{M}$ (relevant Michaelis-Menten values). Since concentrations and Michaelis-Menten values always appear as ratios, any linear variation of a $\ln K_\mathrm{M}$ value can be mimicked by a variation in $\mathbf{x}$-space: instead of increasing a Michaelis-Menten value, we can decrease the corresponding metabolite level, with the same effect on the catalytic rate. Therefore, any linear variation in $(\mathbf{x}, \mathbf{q}_\mathrm{M})$-space can be mimicked by a linear variation in $(\mathbf{x})$-space alone as far as changes in $1/k$ are concerned. Therefore, convexity of $1/ratelaw$ in $\mathbf{x}$-space implies convexity in $(\mathbf{x}, \mathbf{q}_\mathrm{M})$-space. Next, we consider variations of the catalytic constants k_cat and use the same trick: we know that $1/k$ is convex in $(\mathbf{x}, \mathbf{q}_\mathrm{M})$-space, and describe changes of the catalytic constants as variations in $\mathbf{q}_\mathrm{cat}$-space. Again, any linear variation can be mimicked by a linear variation in $(\mathbf{x}, \mathbf{q}_\mathrm{M})$, and so $1/k$ must be convex in $(\mathbf{x}, \mathbf{q}_\mathrm{M}, \mathbf{q}_\mathrm{cat})$-space. So far, we considered only k_cat and K_M values and neglected the activation constants K_A and inhibition constants K_I. In our rate laws these constants appear in similar mathematical terms as the Michaelis-Menten constants. For example, a rate law with competitive inhibition contains similar terms with K_I values and K_M values in its denominator. The terms with K_A values, on log scale, carry a minus sign, but since this term (on log-scale) is linear, the minus sign does not change the convexity. Finally, since (logarithmic) kinetic constants depend linearly on (logarithmic) independent kinetic constants, the enzyme level is also convex in the (logarithmic) independent kinetic constants and (logarithmic) metabolite levels.

There is a also a shorter proof. Without loss of generality, we assume that the flux v_l is positive. To show that $1/k_l$ is a convex function of $\binom{\mathbf{q}}{\mathbf{x}}$, we rewrite (see [4], Eq. 26)

$$\frac{1}{k_l} = \frac{D_l(\mathbf{c}, \mathbf{p})}{k_l^V \sqrt{\prod_i (c_i/K_{M,li})^{m_{li}}}\, 2\sinh(\frac{h_l}{2}\theta_l)} \tag{17}$$

with molecularities m_{li}, the vector $\mathbf{p}$ of kinetic constants, and driving force $\theta_l = -\sum_i n_{il}(\mu_i^\circ/RT + \ln c_i)$. Since e^x is a convex function, expression (17) will be convex in $(\ln \mathbf{c}, \ln \mathbf{p})$ if its logarithm

$$\ln D_l(\mathbf{c}, \mathbf{p}) - \ln k_l^V - \frac{1}{2}\ln(\prod_i (c_i/K_{M,li})^{m_{li}}) - \ln 2 - \ln\sinh(\frac{h_l}{2}\boldsymbol{\theta}_l) \tag{18}$$

is convex in $(\ln \mathbf{c}, \ln \mathbf{p})$. Since the denominator term is a posinomial $D_l(\mathbf{c}, \mathbf{p}) = \sum_a A_{ail}\, c_i^{\alpha_{ail}}\, k_{il}^{\beta_{ail}}$, $\ln D_l$ is convex [31]. Furthermore, $-\frac{1}{2}\ln(\prod_i (c_i/K_{M,li})^{m_{li}})$ is linear in $(\ln \mathbf{c}, \ln \mathbf{p})$ and therefore convex, and $-\ln 2$ is constant and therefore convex. Finally, $-\ln\sinh(\,\cdot\,)$ is convex for any positive arguments, and its argument $\frac{h_l\theta_l}{2}$ is in fact positive (for positive fluxes) and affine in $\binom{\mathbf{q}}{\mathbf{x}}$.

B.2 Model balancing is a convex problem

Based on the proof in section B.1, we can conclude that model balancing is a convex problem. For a proof, we need to show that the likelihood loss for enzyme data, and the negative log priors for enzyme levels are convex functions on the feasible polytope. First, we note that likelihood loss and prior loss are convex functions of the individual enzyme levels $e_l^{(s)}$ and that the concatenation of two convex functions yields a convex function. Thus, it remains to be shown that each enzyme level $e_l^{(s)}$ is a convex function on the feasible polytope. Second, each $e_l^{(s)}$ depends, effectively, only on the kinetic constants of the reaction considered and on the metabolite levels c^s affecting this reaction. Third, given the flux in this state and given the kinetic constants and metabolite levels, the enzyme level $e_l^{(s)}$ is proportional to $1/k_l$ in this state (which, as we saw, is convex in $\ln \mathbf{c}^s$ and in the logarithmic kinetic constants).

C Implementation

A Matlab implementation of Model Balancing, together with example models and data, is available at `https://github.com/liebermeister/cmb`. The file format for models and data (kinetic constants, fluxes, metabolite levels, protein levels) is SBtab [38] and metabolic networks can be defined in SBML [39] or SBtab files. By default, the algorithm starts by running model balancing on an average model state (with metabolic state data given by the geometric mean over the metabolic states). The resulting kinetic constants are then used as initial values for the following full calculation with several metabolic states.

C.1 Possible simplifications and variants of model balancing

Model balancing can be adapted in various ways. (i) If a type of data is not used, likelihood terms for this data type are omitted. Even without any data, priors will keep the results in biochemically plausible ranges. (ii) If certain parameters (e.g. the equilibrium constants) are precisely known, their values can be predefined (e.g. by treating them as data with very small standard errors). (iii) Model balancing also applies to models with irreversible rate laws. In an irreversible rate law, there are fewer kinetic constants (since reverse catalytic constants, equilibrium constants, and velocity constants do not play a role); the forward kinetic constant is a free parameter, and no Haldane relationship is considered. Describing (some or all) rate laws as irreversible changes the structure of the kinetic dependence matrix $\mathbf{M}$. (iv) Different model parameterisations: instead of independent equilibrium constants, standard chemical potentials may be used as independent parameters [11]. (v) A preposterior for kinetic parameters may be obtained by previous parameter balancing, and pseudo values for metabolite and enzyme levels may be obtained by a previous ECM. (vi) To penalise unrealistically high metabolite or enzyme levels, a regularisation term may be added, for example, proportional to the cost function considered in ECM. (vii) Omics data may not contain absolute metabolite and enzyme levels, but relative changes between metabolic states. To account for such data, a variant of the dependence scheme might be considered: for each metabolite, we split the log-concentrations $\ln c_i^{(s)}$ into a reference value $\ln c_i$ and a deviation $\Delta \ln c_i^{(s)}$. Uncorrelated priors for these variables yield a meaningful correlated prior for the metabolite levels, and a similar splitting can be used for enzyme levels.

C.2 Practical computation details

1. **Calculation of the preposterior** To compute the preposterior functions (Eq. 5 for metabolite levels, and similar formulae for enzyme levels and kinetic constants), we need to invert a covariance matrix. This can be numerically expensive. To compute the preposterior of the independent kinetic constants, we need to solve

$$\begin{aligned} \mathbf{C}_{\mathbf{q},\mathrm{ind,pre}} &= [\mathbf{C}^{-1}_{\mathbf{q},\mathrm{ind,prior}} + \mathbf{M}^\top \mathbf{C}^{-1}_{\mathbf{q},\mathrm{data}} \mathbf{M}]^{-1} \\ \bar{\mathbf{q}}_{\mathrm{ind,pre}} &= \mathbf{C}_{\mathbf{q},\mathrm{ind,pre}} \left[\mathbf{C}^{-1}_{\mathbf{q},\mathrm{ind,prior}} \bar{\mathbf{q}}_{\mathrm{ind,prior}} + \mathbf{M}^\top \mathbf{C}^{-1}_{\mathbf{q},\mathrm{data}} \bar{\mathbf{q}}_{\mathrm{data}}\right] . \end{aligned} \tag{19}$$

   The matrix inversion for $\mathbf{C}^{-1}_{\mathbf{q},\mathrm{ind,prior}}$ and $\mathbf{C}^{-1}_{\mathbf{q},\mathrm{data}}$ (covariance matrices for metabolite and enzyme levels) is easy because the original covariance matrices are diagonal and the projector matrices $\mathbf{P}$ select single vector elements. However, inverting the term in brackets may be hard. To speed up the calculation, we set $\mathbf{A} = \mathbf{C}^{-1}_{\mathbf{q},\mathrm{ind,pre}}$ and obtain the similar formulation

$$\begin{aligned} \mathbf{A} &= \mathbf{C}^{-1}_{\mathbf{q},\mathrm{ind,prior}} + \mathbf{M}^\top \mathbf{C}^{-1}_{\mathbf{q},\mathrm{data}} \mathbf{M} \\ \bar{\mathbf{q}}_{\mathrm{ind,pre}} &= \mathbf{A}^{-1} \left[\mathbf{C}^{-1}_{\mathbf{q},\mathrm{ind,prior}} \bar{\mathbf{q}}_{\mathrm{ind,prior}} + \mathbf{M}^\top \mathbf{C}^{-1}_{\mathbf{q},\mathrm{data}} \bar{\mathbf{q}}_{\mathrm{data}}\right] . \end{aligned} \tag{20}$$

   Now the costly matrix inversion in the first equation is avoided, and the right-hand side in the second equation

can be computed without explicitly computing the matrix inverse (e.g. by using the matrix left division operator $\backslash$ in matlab). This calculation is faster and works for sparse matrices.

2. **Reactions with vanishing flux** If reaction flux is non-zero, the flux direction puts a constraint on the driving force, and the predicted enzyme level is positive. If a reaction is always inactive – that is, in *all* metabolic states – the kinetic constants for this reaction are ill-determined, and the reaction can be removed from the model. But what if a reaction fluxes vanish in some of the metabolic states? The vanishing flux can either be caused by a vanishing enzyme level, or by a vanishing thermodynamic force. If the reaction is known to be in chemical equilibrium, we also set the driving force to 0, which leads to an extra equality constraint on metabolite levels. In this case, the enzyme level can be positive and needs to be estimated (although the economical "principle of dispensable enzyme" would suggest a vanishing enzyme level in this case). Otherwise, with a zero flux and a non-zero driving force, the enzyme activity must be zero: for an enzyme without allosteric inhibition, this means that the enzyme concentration must vanish.

3. **Divergence of enzyme levels close to polytope boundaries.** Each thermodynamic constraint defines a boundary of the feasible polytope. Close to this boundary, an enzyme levels goes to infinity and the likelihood function explodes. This steep increase can cause numerical problems during optimisation. To handle them, we may apply the logarithm function once more to the (likelihood or posterior) score, and use the resulting function as our minimisation objective. This new objective function will still go to infinity at polytope boundaries, but less steeply. The new objective function may be non-convex, but since it depends monotonically on a convex function, it will still have a single local minimum. A second way to avoid this problem is to exclude problematic regions close to the boundary by introducing some extra constraints. In practice. we can make all thermodynamic constraints a bit tighter, by requiring small, non-zero thermodynamic forces in every reaction [31].

4. **Starting point for optimisation** To obtain an initial point for our optimisation, we may first run model balancing for an average metabolic state. This yields a first guess of the kinetic constants. Alternatively, we can run model balancing separately for each metabolic state. In each run, we start from the prior mode (or alternatively, from the posterior mode for kinetic constants obtained by Parameter Balancing, and the posterior mode for each metabolite value). The resulting concentration vectors and the state-averaged (arithmic/geometric) kinetic constant vector can be used as initial values for the multi-state problem.

5. **Running parameter balancing as a separate first step** Model balancing can also be run in two steps. The first step, is a simple parameter balancing problem: we consider only kinetic constants and fit them to kinetic data. The result is a multivariate Gaussian posterior for all (logarithmic) kinetic parameters [11] that summarizes all data and prior knowledge about the kinetic constants. In the second step, we use this posterior as a prior for the kinetic constants, and fit kinetic constants and model states (metabolite and enzyme levels) to metabolite and enzyme data. Since the kinetic data have already been used to define the prior, they can be ignored in this part of the estimation. The calculation is equivalent to the method described in this paper. By processing the kinetic data separately in advance, we can learn more clearly what information is contained in the kinetic data alone, before combining them with metabolic data. Moreover, a known kinetic "prior" that includes all information about kinetic data may allow us to further constrain the kinetic constants in order to reduce the feasible search space.

D Example model

The *E. coli* central carbon metabolism model, taken from [31], comprises 40 metabolites and 30 reactions and contains 107 K_{M} values and 167 kinetic constants in total (K_{M} values as well as forward and reverse k_{cat} values)

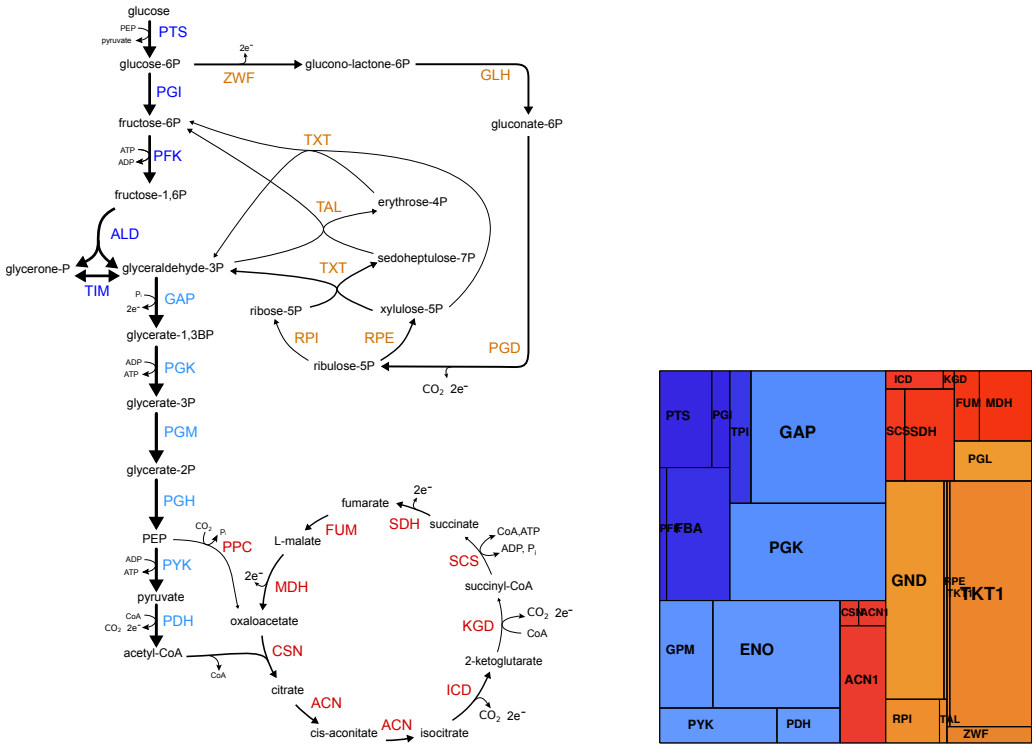

Figure 16: Model of *E. coli* central carbon metabolism and protein data, both taken from [31].

. The model structure is shown in Figure 16 and described at `https://github.com/liebermeister/cmb` (in the file resources/data/data-organisms/escherichia_coli/network/ecoli_noor_2016.tsv).

To model aerobic growth on glucose, I used a data set from [31], which gathered measured flux data from [40], proteomics data from [41], and metabolomics data from [42]. To model several metabolic states, I used a data set from [16], where a larger network model had been considered, proteomics data from different sources were used, and flux data had been computed by FBA. I linearly the flux data onto the *E. coli* model to obtain complete and consistent flux values. A comparison between the two data sets reveals a discrepancy in scaling: the (FBA-derived) fluxes from [16] were smaller than the fluxes taken from [31] by an approximate factor of 10, while enzyme levels were smaller by an approximate factor of 2.

E Prior distributions and artificial data

To define priors, pseudo values, and constraints (for kinetic constants, metabolite levels and enzyme levels), I used the default values from parameter balancing (see `www.parameterbalancing.net`. However, when running parameter balancing as a test, I found that the available k_{cat} values were typically much higher than the prior median value, as expected for enzymes in central metabolism [8]. In line with these data, I changed the prior for k_{cat} values from a median of 10 s^{-1} (geometric standard deviation 100) to a median of 200 s^{-1} (geometric standard deviation 50). Likewise, I changed the prior width for K_{M} values from a geometric standard deviation of 10 to a geometric standard deviation of 20 (while keeping the median 0.1 mM unchanged). A table describing the priors is provided in the github repository, file resources/data/data-prior/cmb_prior.tsv. These values, used in the matlab implementation, can be easily modified.

Artificial kinetic constant data were generated as follows. Given the network structure, true artificial kinetic constants were generated by assigning random (log-normal) values to $\ln K_{\mathrm{eq}}^{\mathrm{ind}}$, $\ln K_M$, and $\ln K_V$ and computing

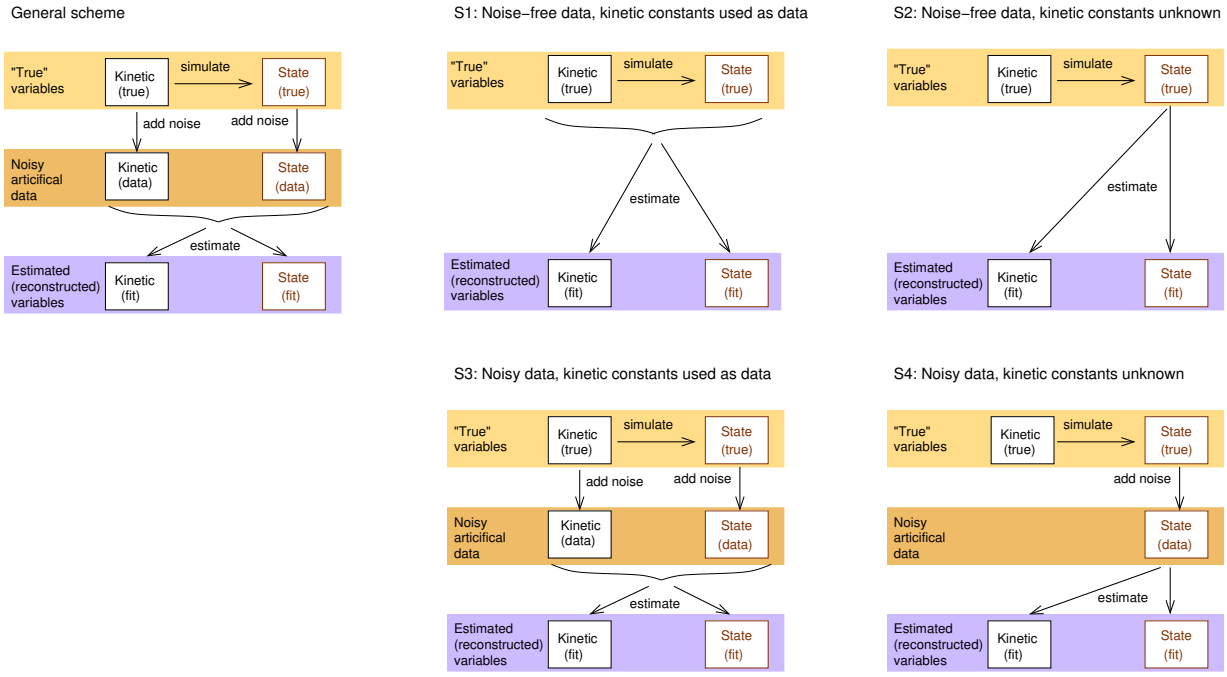

Figure 17: Estimation scenarios with artificial data. Left: general procedure. In a given model, kinetic constants are drawn from random distributions (respecting their interdependencies), and metabolic state data are generated by combining sampling and simulation runs (top row). From these "true" values, artificial (kinetic and state) data are generated by adding uncorrelated noise (centre row). Model balancing is used to estimate the kinetic parameters and metabolic state variables (bottom row), aimed to resemble the true values. Right: I employed four variants of this procedure (called S1-S4) in which noise is either considered or not (in the latter case, the noise level is set to zero), and kinetic data are used or not. In another variant, data for equilibrium constants are used as the only kinetic data.

the other constants. The random values were sampled from the same distributions that are used as priors in model balancing.

To generate artificial metabolic state data, enzyme levels and external metabolite levels were randomly sampled from the same distributions that are used as priors in model balancing. Then the model was parameterised with the "true" artificial kinetic constants and was solved to obtain a steady reference state (steady-state metabolite concentrations and fluxes). Based on this reference state, a number of metabolic states were constructed by randomly varying metabolite and enzyme levels (again, following the prior distribution) and computing the (non-steady) reaction rates[19]. The resulting states are seen as the "true values". To generate noisy state data, uncorrelated random noise was added to the "true values". When generating artificial data, noise was also added to fluxes but the flux signs were kept unchanged, to ensure thermodynamically feasible flux directions as required in model balancing.

[19] Alteratively, one could similate a dynamic time course and take snapshots at different time points.