Bayesian regression facilitates quantitative modelling of cell metabolism

Teddy Groves,*,† Nicholas Luke Cowie,† and Lars Keld Nielsen†

†*DTU Biosustain, DTU, Kongens Lyngby, Denmark*

‡

E-mail: tedgro@biosustain.dtu.dk

Abstract

This paper presents Maud, a command-line application that implements Bayesian statistical inference for kinetic models of biochemical metabolic reaction networks. Maud takes into account quantitative information from omics experiments and background knowledge, as well as structural information about kinetic mechanisms, regulatory interactions and enzyme knockouts. Below, we review the existing options in this area, explain how Maud improves on the state of the art, describe the intended modelling workflow and illustrate its use with an example application.

Keywords

Bayesian inference, kinetic models of cell metabolism

1 Introduction

A kinetic model of cellular metabolism aims to express what is known about a cellular process in the form of an *in silico* representation of the underlying network of chemical reactions.

Kinetic models can be used to improve production of target molecules, determine regulatory networks (*1*) and identify potential drug targets (*2*, *3*). However, the use of kinetic models in practice is hindered by their dependence on noisy and uncertain information sources. Quantitative in vivo measurements of chemical abundances, and in vitro measurements relating to kinetic parameters, both contain vital information but are notoriously inaccurate (REFERENCES). Practically useful kinetic modelling therefore requires a principled statistical approach that encompasses multiple possible model parameterisations.

Bayesian statistical inference can combine the structural information implicit in kinetic models with knowledge about metabolic parameters and information from omics measurements (*4*, *5*). However, kinetic models pose serious computational challenges for Bayesian inference (*6*, *7*).

The scope of a kinetic model is defined by a stoichiometric matrix, S, in which rows represent metabolites, columns represent reactions, and matrix elements s_{ij} represent the stoichiometric coefficient of metabolite i in reaction j. The change in metabolite concentrations is:

$$\frac{dC}{dt} = S \cdot v - \mu \cdot C \tag{1}$$

In equation (1), C represents a vector of metabolite concentrations, v is a vector of reaction rates, and μ is the growth rate. The second term represents the dilution due to cell growth.

In a kinetic model, the rates, v, are expressed as a function of the enzyme concentrations, E, the metabolite concentrations, C, and a set of parameters, θ as shown in equation (2)

$$v = f(C, E, \theta) \tag{2}$$

The parameters must include sufficient boundary concentrations and fluxes to solve (1).

It is common to assume pseudo-steady state for metabolites, i.e., the rate of fluxes towards

any metabolite is much greater than the rate of change in concentration, $v \gg \frac{dC}{dt}$. Moreover, the dilution effect is assumed minimal, $\mu \cdot C \ll \vec{v}$ (true unless the concentration is very high). Finally, the enzyme concentration is assumed constant for the period considered and hence part of the parameters.

Given these assumptions and a set of values for , a set of steady state metabolite concentrations and fluxes can be found by solving for C the algebraic equation:

$$S \cdot f(C; \theta) = 0 \tag{3}$$

In a fermentation context, (3) captures the rapid kinetics inside the cell, while another set of ODEs would be used to describe the external substrate and product concentrations, which could act as boundary parameters to (3).

In the context of kinetic modelling, Bayesian inference is appealing because it allows uncertainty to be represented appropriately without sacrificing mechanistic accuracy. Measurement uncertainty can naturally be represented in a Bayesian measurement model, whereas the prior model can represent quantitative uncertainty about kinetic parameters. Finally, kinetic rate laws can be represented in Bayesian data generation models with arbitrarily high fidelity. See Gelman et al. (*8*) for more about Bayesian inference and Gelman et al. (*9*) for a discussion of practical Bayesian workflow.

Another advantage is that Bayesian inference problems are well-posed even when not all parameters are strongly identified. Sloppy models in which measurable quantities are sensitive to combinations of parameters but not to individual marginal parameter values are ubiquitous in models of biological systems (*6*, *10*). The parameter correlation structure represents the set of potential models that describe the observed data. As we demonstrate in our case study below, capturing this correlation structure is difficult outside a Bayesian context.

Previous Bayesian kinetic models have either sacrificed mechanistic accuracy or have at-

tempted to fit realistic kinetic models using obsolete or unreliable computational methods.

The most popular algorithm for fitting Bayesian statistical models is Markov Chain Monte Carlo (MCMC). Modern MCMC algorithms allow exploration of high- dimensional posterior distributions, have robust failure diagnostics (*11*) and can incorporate fast numerical solvers, thereby making inference feasible for Bayesian kinetic models. Nonetheless, the kinetic modelling literature reports an aversion to MCMC, rooted mainly in concerns about sampling time and the presumed difficulty of implementing the required statistical model (*4*, *12*). We are only aware of two recent attempts to implement a Bayesian kinetic modelling approach using MCMC. Stapor et al. (*13*) fitted detailed kinetic models using relatively inefficient MCMC algorithms that do not scale well to high dimensional parameter spaces limiting the scope of modelling. Conversely, St. John et al. (*14*) utilises an efficient sampling algorithm but uses approximate kinetics, namely lin-log kinetics Visser and Heijnen (*15*), limiting the scope of interpreting parameters and inferring cellular behaviour in experimental conditions outside the reference dataset.

There have also been efforts to implement Bayesian inference for kinetic models without the use of MCMC. Examples of alternative inference methods include variational inference as in St. John et al. (*14*), rejection sampling and approximate Bayesian computation Saa and Nielsen (*4*) and Laplace approximation, in which the Fisher information matrix is used to calculate a normal approximation around the maximum a posteriori parameter configuration (*5*, *12*, *13*, *16*). Non-MCMC-based Bayesian kinetic models have limited utility because they lack reliable diagnostic tools for verifying that their results approximate the target posterior distribution. This is a problem because realistic kinetic models tend to induce highly correlated, non-Gaussian, joint probability distributions (*6*, *13*).

Our application Maud is the first Bayesian kinetic model to combine biologically realistic mechanistic accuracy—including accurate rate laws, post- translational modification and thermodynamics—with fast, robust MCMC sampling using adaptive Hamiltonian Monte

Carlo. Further, Maud is a general-purpose application that can be used to fit a wide range of Bayesian kinetic models.

2 Results and Discussion

To demonstrate our application's capabilities, we used Maud to analyse an artificial dataset based on the human methionine cycle. We first used Maud to generate simulated training and validation measurements based on plausible parameter values, then performed posterior sampling. Next, we used Maud to predict the validation data.

To show that Maud is robust to missing measurements we compared the results of fitting the full dataset with an intentionally incomplete dataset. To demonstrate why a full Bayesian approach is preferable to an approach based on a Laplace approximation of the posterior distribution, we also fit our methionine cycle model using the latter method and compared the results with MCMC sampling.

Finally, we investigated our results to find out what our model learned about the contributions of different regulatory factors to the flux through GNMT, an important reaction. This analysis illustrates how Maud can be used to generate actionable insights about metabolism without the need for further statistical analysis.

The methionine cycle, illustrated in Figure 1, is a fundamental pathway in human metabolism, whose intermediate metabolites participate in a variety of mechanisms which must compete for the same resources. Due to this competition, as well as the fact that all the functions occur simultaneously, the methionine cycle is highly regulated, with 6 known allosteric effectors (REFERENCE). This complex regulation means that quantitative modelling of the methionine cycle requires a detailed kinetic model: this is why we chose it as a case study for Maud.

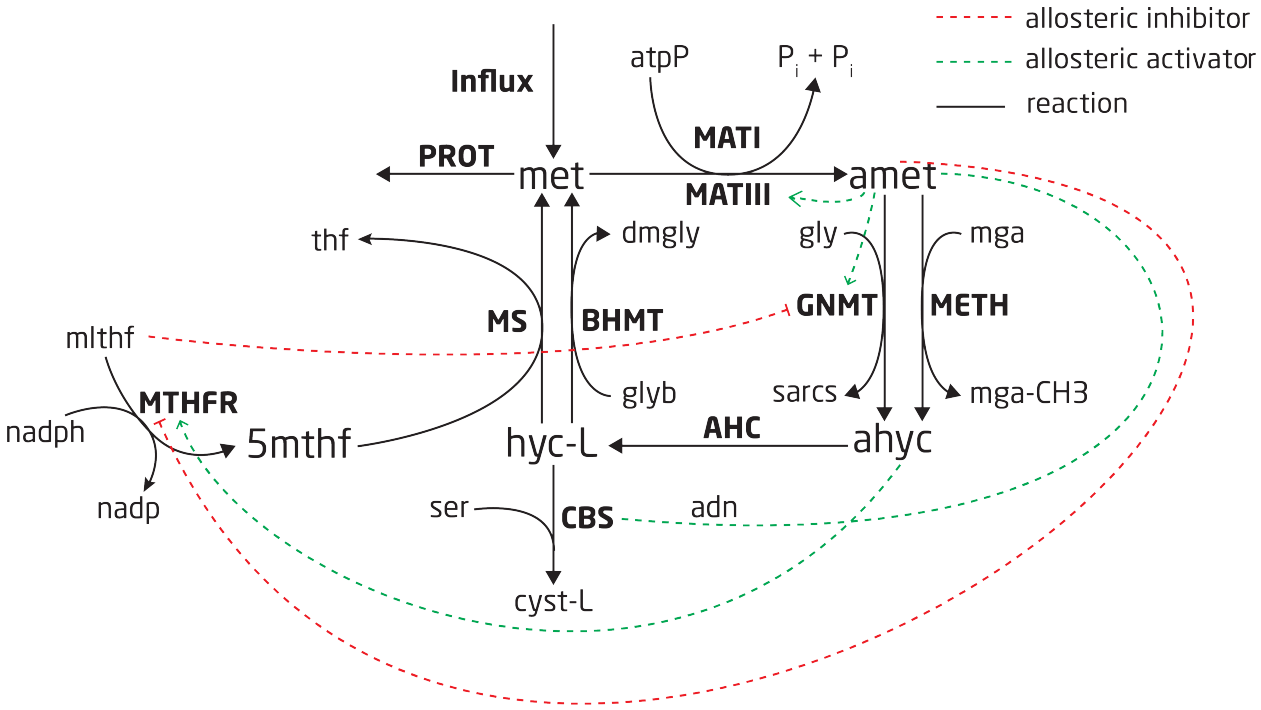

Figure 1: The methionine cycle as modelled, with the solid black lines representing the reactions, the green lines representing allosteric interaction, and the red lines representing allosteric inhibition. The bold fonts are the reaction names and the regular font represents the metabolites.

2.1 Dataset and model specification

The simulated dataset and underlying kinetic model that we used for our analysis can be found at https://github.com/biosustain/Methionine_model/tree/main/data/methionine and is described in supplementary information section [REFERENCE].

We constructed a kinetic model of the methionine cycle in Maud's format using the description in Korendyaseva et al. (*17*). The ordinary differential equation system describing this model is shown in Equation (4).

$$\begin{aligned}
\frac{d[met]}{dt} &= v_{Influx} - v_{PROT} - v_{MAT} + v_{MS} + v_{BHMT} \\
\frac{d[amet]}{dt} &= v_{MAT} - v_{GNMT} - v_{METH} \\
\frac{d[ahyc]}{dt} &= v_{GNMT} + v_{METH} - v_{AHC} \\
\frac{d[hyc-L]}{dt} &= v_{AHC} - v_{CBS} - v_{MS} - v_{BHMT} \\
\frac{d[5mthf]}{dt} &= v_{MTHFR} - v_{MS}
\end{aligned} \tag{4}$$

After specifying the qualitative aspects of the kinetic model, we selected parameter values to use as ground truth by Monte Carlo sampling using a previous model of the methionine cycle as a starting point (see Saa and Nielsen (*4*) for this model).

We used these parameters to simulate steady states in a range of plausible experimental conditions, again using Saa and Nielsen (*4*) as a starting point. These steady states were then used to generate simulated measurements using the measurement model described below in Section 2.1.1.

2.1.1 Measurement model

Generating an artificial dataset required a specification of the true measurement error. For enzyme and metabolite concentration measurements we specified a standard deviation of 0.1 on natural logarithmic scale, corresponding to approximately 10% measurement error. For each reaction measurement a measurement standard deviation of approximately 10% of the simulated value.

These measurement error specifications are somewhat optimistic considering the many sources of variation and uncertainty affecting quantitative proteomics, metabolomics and fluxomics analyses, but are a reasonable first approximation to a realistic set of measurements.

For our main model run, we assumed that all metabolite and enzyme concentrations were measured, and that there was a reaction measurement for each of the network's elementary flux modes.

2.1.2 Training/validation split

The training testing split was selected to achieve a large difference between the fluxes of the training and testing dataset. The split was determined as we are interested in showing how our model can fit to varied conditions, and conditions closer to the training set are likely to be predicted well without necessarily learning the system.

2.1.3 Additional dataset with missing measurements

To gain insight into our model's robustness to missing measurements, we also performed a model run with the same 6 experimental datasets, but with measurements of the metabolite S-Adenosyl-L-homocysteine, or "ahcys" removed. Since ahcys regulates three enzymes in the methionine cycle, including one enzyme which is also thermodynamically regulated, we expected the removal of these measurements to yield interesting results.

2.1.4 Maud input specification

We constructed inputs in Maud's format for each of the analysed datasets, based on the scenario that the true kinetic model was known except for parameter values, which needed to be inferred from the training data and priors. These inputs can be found at https://github.com/biosustain/Methionine_model/tree/main/data.

The prior distributions and corresponding true parameter values used in our case study are shown in supplementary materials section [REFERENCE]. The first two columns show the 1% and 99% quantiles of each marginal prior distribution. True parameter value are shown in column three, and the last column shows the z-score on log scale of the true parameter value according the marginal prior distribution. As can be seen from the table, there are 7 parameters for which the true value is outside the 1%-99% range. This is desirable, making the case study more realistic, because extreme deviance from the prior distribution is likely to occur in practice due to in vivo to in vitro measurement differences.

2.2 Findings

2.2.1 Posterior inference

Running standard diagnostic checks indicated that the samples we generated were from the target posterior distribution. The improved $\hat{R}$ statistic (*11*) for every variable of interest was within 2% of 1, indicating appropriate mixing within and between Markov chains. Additionally, the number of effective samples was high, indicating that we generated enough posterior samples to support inferences about the bulks of the distributions of the sampled parameters. Furthermore, we observed no post warm-up divergent transitions, indicating that the sampler was able to transform the log-posterior distribution, avoiding any regions with excessive curvature that might inhibit exploration via HMC.

Posterior predictive checking indicated that our model achieved a good fit to the simulated reaction and metabolite concentration measurements, as shown by the graphs in the top row

of figure Figure 2.

Analysis of the posterior distributions for kinetic parameters indicated that these are highly correlated. The marginal posterior distributions for most kinetic parameters did not shrink significantly compared with the corresponding marginal prior distributions, even though these parameters' joint posterior distribution contained enough information to make accurate out of sample predictions. In some cases, there were two-dimensional correlations such as the one shown in the bottom left of figure Figure 2; in this case the marginal distribution of the two parameters is roughly banana shaped. More commonly, however, two-dimensional pair plots were insufficient to reveal the underlying correlation structure, as seen in the bottom-right plot in figure Figure 2. This does not mean that the parameters were uncorrelated, but rather that the correlations involve more than two parameters.

Overall, our results show that Maud can fit a realistic pathway-sized dataset. This was achieved without fixing the marginal values of kinetic parameters: the information required to make good predictions was contained in the correlation structure of the joint posterior distribution. This finding is consistent with previous analyses of biological systems that found they are "sloppy", that is, sensitive to parameter combinations rather than marginal parameter values, with important combinations, scales and regions of sensitivity being difficult to ascertain in advance (*6*, *18*).

The question naturally arises whether the crucial high-dimensional parameter correlations are linear or non-linear. This is relevant to the question of model performance, as linear correlations are easier to correct for. A linearly correlated posterior space would also be easier to summarise. We address this question below in Section 2.2.2.

This case study illustrates the type of kinetic model and dataset that Maud can fit. The model we analysed has 10 reactions, 5 state variables and 212 parameters. Generalising from our ability to fit this model in a reasonable time using Maud, we expect that Maud can be used to fit realistic Bayesian models of approximately the same size, but not, for example,

genome-scale kinetic models. To fit larger models, faster steady state solving methods or alternative inference algorithms will be required. Section 2.2.2 addresses whether Laplace approximation is a suitable candidate.

2.2.2 Comparison with Laplace approximation

We found that the Laplace method was not able to produce an accurate posterior approximation for our model and dataset.

Our overall strategy was to compare samples generated using MCMC, which we were confident could be treated as draws from the true posterior distribution, with approximate posterior samples generated using the Laplace method. We were unable to generate approximate posterior samples for our main methionine cycle case study using Laplace approximation, as the algorithm could not recover from solver failures caused by unrealistic parameter configurations. We therefore made a comparison for a simpler model. This input can be found at https://github.com/biosustain/Methionine_model/tree/main/data/example_ode.

MCMC sampling for our comparison input yielded 200 samples in 625 minutes; Laplace sampling yielded the same number of samples in only one minute. The diagnostics indicated that our algorithm was able to find the maximum a posteriori parameter configuration, approximate the Hessian and use these quantities to generate approximate posterior samples. The results can be found at https://github.com/biosustain/Methionine_model/tree/main/results.

Figure 3 summarises the results of comparing the samples generated using each method. As can be seen from the top left plot, the Laplace method does not provide a good approximation to the true posterior distribution in this case, as the marginal distribution of the total log probability density is clearly different. This impression was confirmed using the Kolmogorov-Smirnov test, which is a test to differentiate two empirical univariate distributions. The two distributions were significantly different with a p-value indistinguishable from zero.

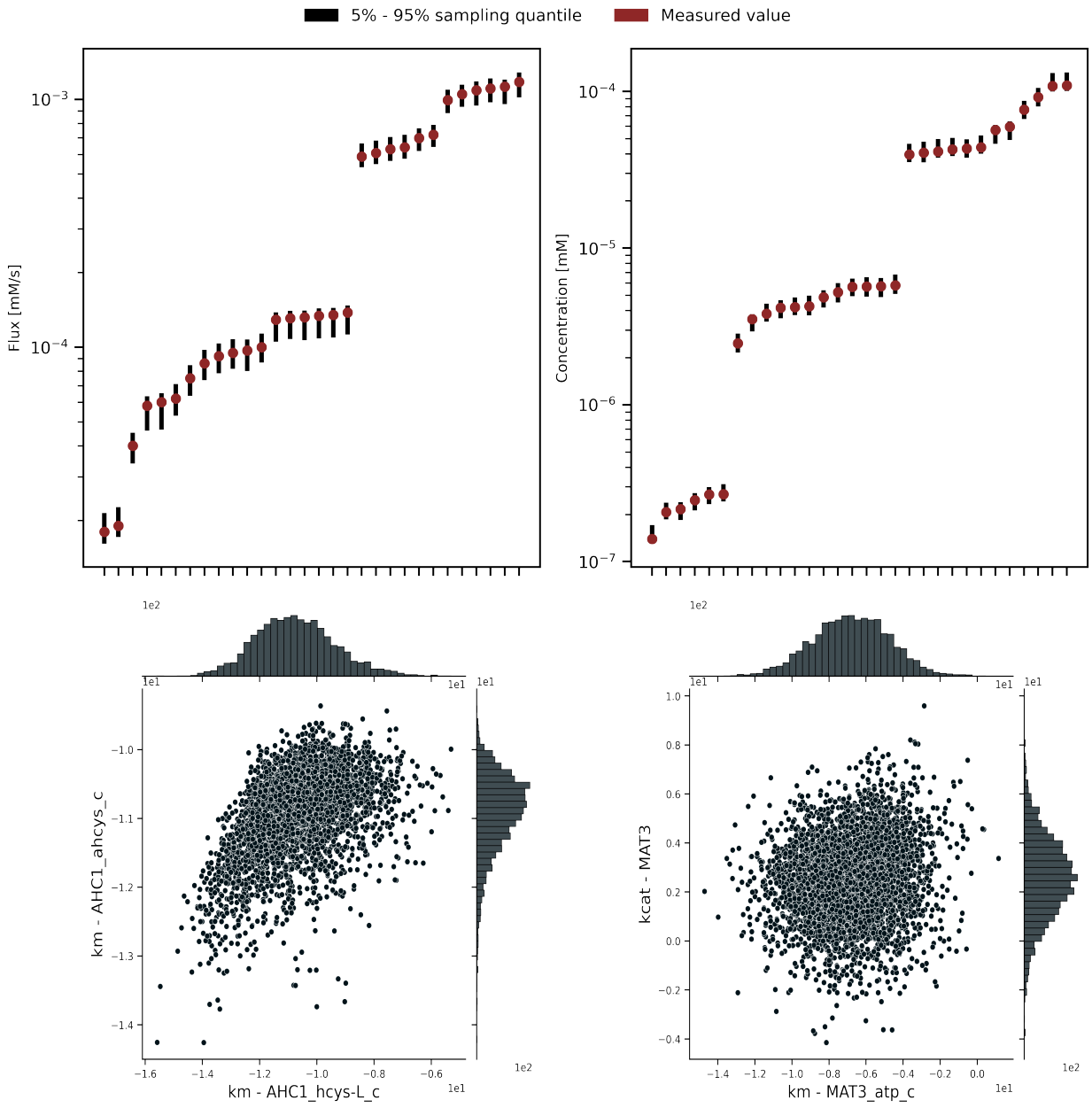

Figure 2: Marginal posterior distributions from our main model run. (**Top left**) Comparison of posterior predictive intervals with simulated flux measurement values. All the flux measurements are within the predictive intervals, indicating a good fit. (**Top right**) Comparison of posterior predictive intervals with simulated concentration measurement values. These also show a good fit. (**Bottom left**) Pairwise marginal posterior distribution for two uncorrelated parameters, namely $K_m^{AHC1,hcys-L}$ and $K_m^{AHC1_a hcys}$. (**Bottom right**) Pairwise marginal posterior distribution for two correlated parameters, namely $K_m^{MAT3,atp}$ and K_{cat}^{MAT3}.

The difference between the Laplace approximation output and the true posterior distribution manifests not only in the parameter space, but also in the measurement space. Figure Figure 3 frame B compares the 5%-95% interval for flux measurement log likelihoods in the true posterior with the Laplace approximation; lower log likelihood values indicate that the modelled and measured values are further away. The graph shows that the Laplace approximation yielded significantly worse predictions than the true posterior, even for the training data.

To further explore why this is the case we compared samples from the true posterior and the Laplace approximation for the pairwise marginal distributions of two Michaelis-Menten constants $K_m^{A,r1}$ and K_{cat}^{r1}: see the bottom right cell of figure Figure 3. This comparison demonstrates that the Laplace method is not able to capture the correct relationships between parameters' distributions.

This result shows that MCMC, while slower than Laplace approximation, is unfortunately preferable for posterior inference in this case. We expect that the Laplace method will produce worse approximations the more complex the target model. Since the approximation is already unacceptable for our simple test model, we recommend that Maud users use MCMC sampling in preference to Laplace approximation if possible when fitting realistic Bayesian kinetic models.

Our results here also provide circumstantial evidence that the parameter correlations in Bayesian kinetic model posteriors tend to be non-linear, as a posterior with only linear correlations would likely be more germane to Laplace approximation. A conclusion that we drew from this analysis was that the results of fitting our model cannot be summarised simply, for example by fitting a multivariate normal distribution to the posterior draws. We therefore recommend that Maud users store the full set of MCMC draws rather than using such an approximation. This does not preclude the possibility that there is an alternative, more compact, way to summarise the results of Bayesian kinetic model inference; we leave

research into this topic to future work.

2.2.3 Effect of missing metabolite concentration measurements

Comparing model runs with and without the ahcys measurements showed that Maud can produce sensible results even from incomplete metabolomics data.

Figure 4 shows that, as might be expected, the model with missing measurements did not correctly infer the missing ahcys concentrations. Nonetheless, the remaining measured metabolites were still well predicted, suggesting that information about the network is still preserved despite the missing measurements. Comparison of flux measurements in both models also indicated that removing the ahcys measurement did not result in catastrophic model failure.

The missing measurements did affect Maud's ability to infer parameter values correctly. The lower left plot of Figure 4 shows that the model with full metabolomics learned the true value for the displayed dissociation constant, despite this value being far from the mean of the corresponding marginal prior distribution. In contrast, the model with missing measurements stayed in the neighbourhood of the prior.

This result is reassuring because not having access to all measurements is a common situation in multi-omics studies. For instance, measuring all metabolites in a pathway can be infeasible because of limitations of mass spectrometers, availability of standards, column effects, and compartmentalisation. However, provided that sufficient information is available from other sources, our approach can produce sensible results from incomplete metabolomics data.

2.2.4 Application to regulatory understanding

To demonstrate how Maud's output can be used to yield useful metabolic insights we used the results of our case study to explain why the flux of the enzyme $GNMT$ is higher in dataset 1 than in dataset 12. GNMT is an irreversible enzyme that is homotropically activated by its substrate, competitively inhibited by its product and heterotropically inhibited by 5,10-

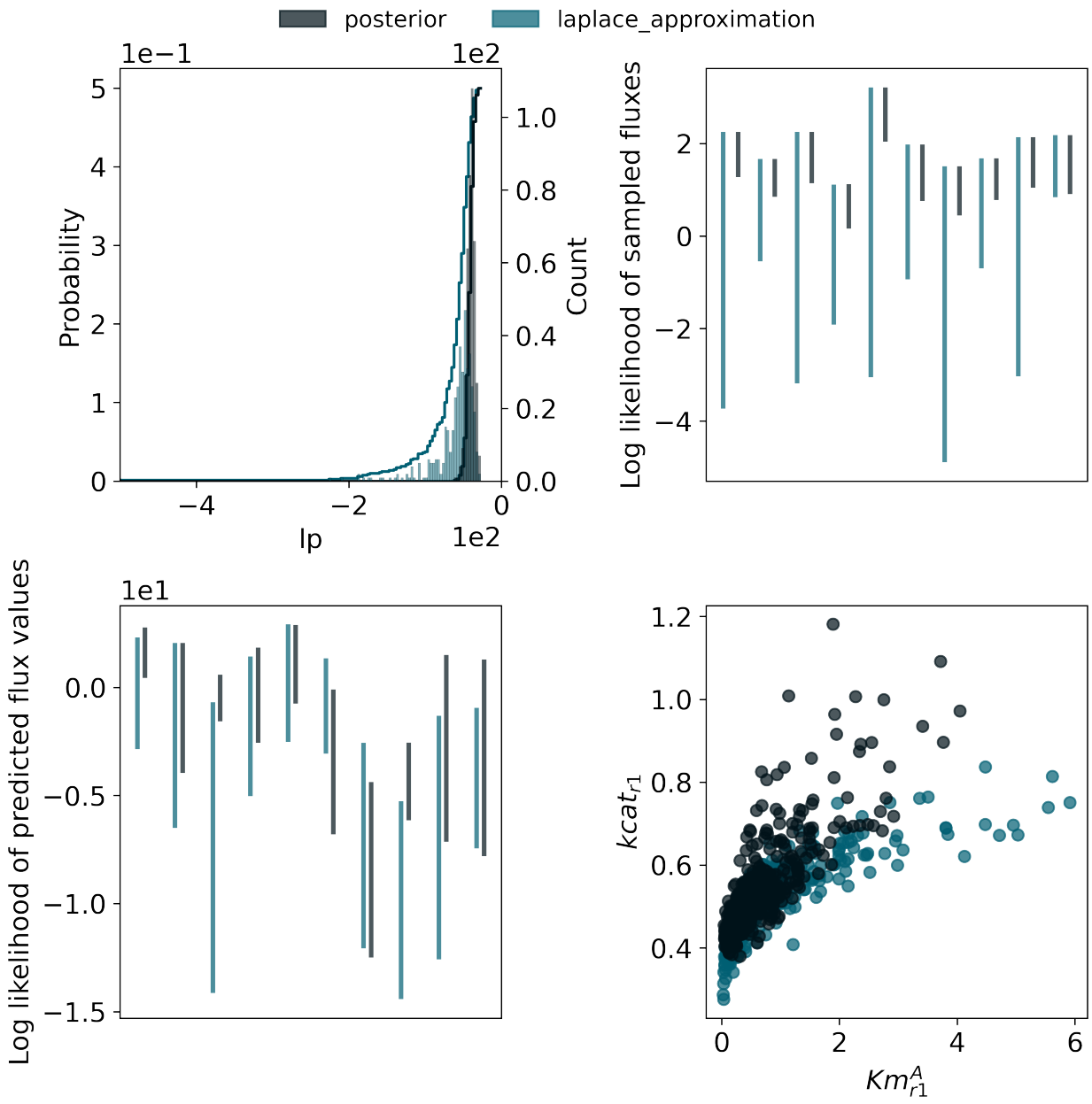

Figure 3: Graphical comparison of approximate posterior samples generated using Laplace sampling (blue-green) with posterior samples generated using MCMC (dark grey). (**Top left**) the two sets of samples clearly have different marginal distributions for the overall log probability variable, indicating that the Laplace samples do not accurately approximate the target distribution. (**Top right**) The distribution of marginal posterior predictive log likelihood values for training data flux measurements shows that the Laplace method tended to yield much worse predictions compared with the true model. (**Bottom left**) The Laplace method also tended to produce worse flux predictions for held-back test measurements. (**Bottom right**) The marginal joint distribution of two parameters: $K_m^{A,rl}$ and K_{cat}^{r1}. The Laplace method is not able to track the correct joint distribution for this pair of parameters. This is unsurprising given that the target distribution has position-dependent scales which are difficult for a linear approximation to capture.

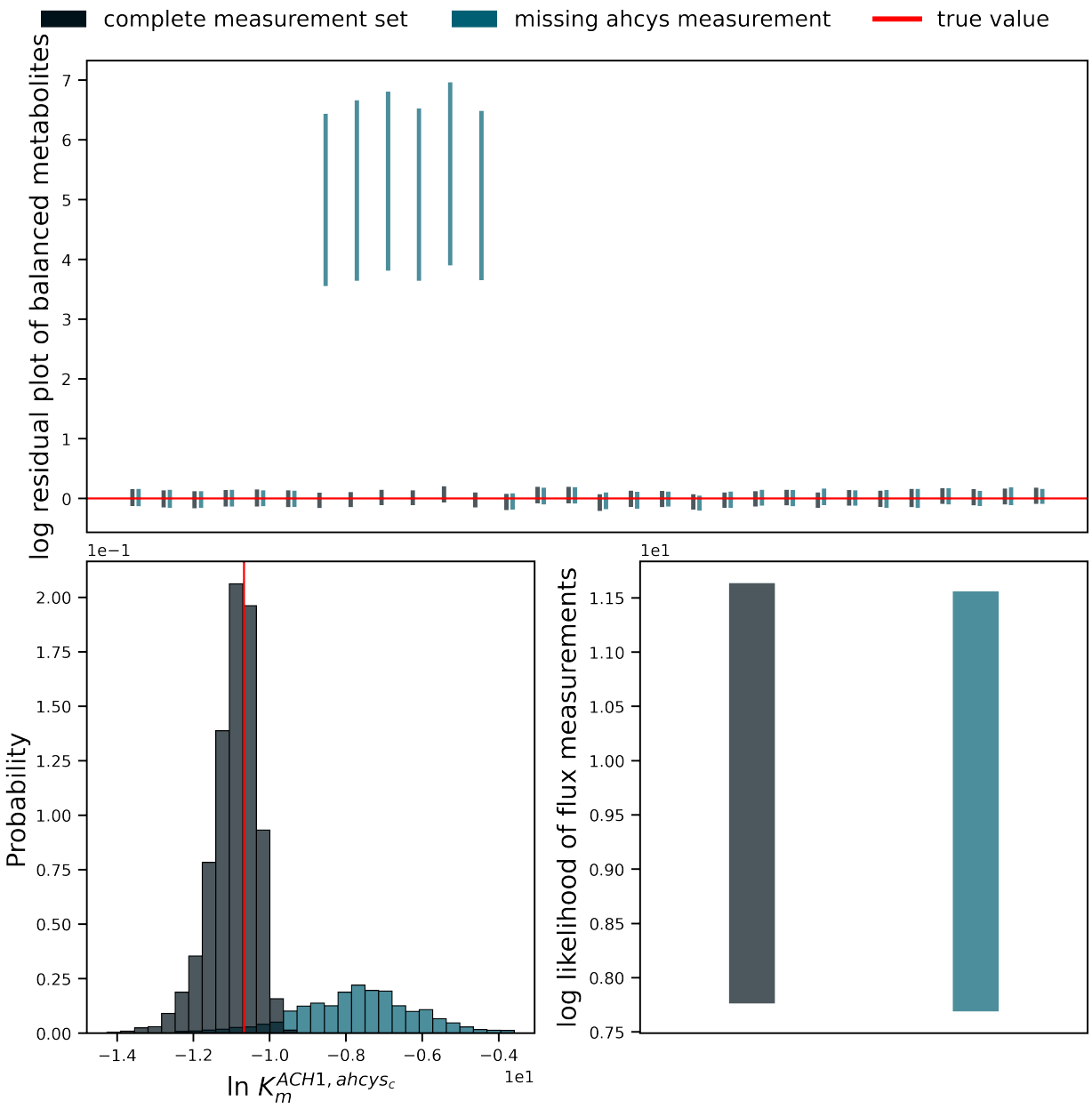

Figure 4: Results of removing concentration measurements for the metabolite $ahcys_c$ from our case study dataset. (**Top**) A comparison of metabolite concentration residuals between the full measurement dataset (blue-green) and the missing-data dataset (grey), displayed on natural logarithmic scale. The missing-data model was unable to predict the withheld $ahcys_c$ concentrations. (**Bottom Left**) The marginal posterior distribution for the Michaelis constant $K_m^{AHC1,ahcys_c}$ in each model, alongside the true parameter value used to generate both datasets. The true value is recovered by the complete-data model but not by the missing-data model. (**Bottom Right**) The distribution of total log-likelihood for out-of-sample flux measurements in both models. There is a significant overlap between the two distributions, suggesting that removing the $ahcys_c$ measurement did not cause catastrophic prediction failure. However, overall the complete-data model tended to make better predictions.

methylenetetrahydrofolate (mlthf). The diverse regulation makes it the ideal test case to elucidate regulatory changes.

Figure 5 shows the regulatory description of GNMT, according to the results our main case study analysis. Each curve shows the marginal posterior distribution of the ratio of the corresponding regulatory component in dataset 1 compared with dataset 12. A positive value indicates that the component was increased in dataset 1 relative to dataset 12, with 0 indicating no difference. The probability, according to our model, of the component acting in each direction is given by the relative area under the curve on each side of the 0 point.

Our model correctly inferred that saturation and allosteric effects were the main drivers of regulation between the two datasets in this case, with the curves for each component aligning with the ground truth shown in red.

Importantly, this form of analysis takes into account all modelled sources of uncertainty, including uncertainty about the true values of the flux in each dataset, and propagates this uncertainty into the final conclusion. Our result shows that Maud could be used in this realistic case not only to provide a user with an explanation for an observed difference in fluxes, but also a reasonable judgement as to the explanation's robustness.

3 Methods

Maud is a command line application implementing Bayesian inference for a wide range of realistic kinetic models. Maud is written in Python (*19*), designed for use on Windows, macOS and Linux, registered on the Python Package Index as `maud-metabolic-models`, documented at https://maud-metabolic-models.readthedocs.io and actively developed and maintained at https://github.com/biosustain/Maud/.

To use Maud, a user must first collate appropriate input information, represent it in files with Maud's required formats (see section (REFERENCE)below). Maud's command line

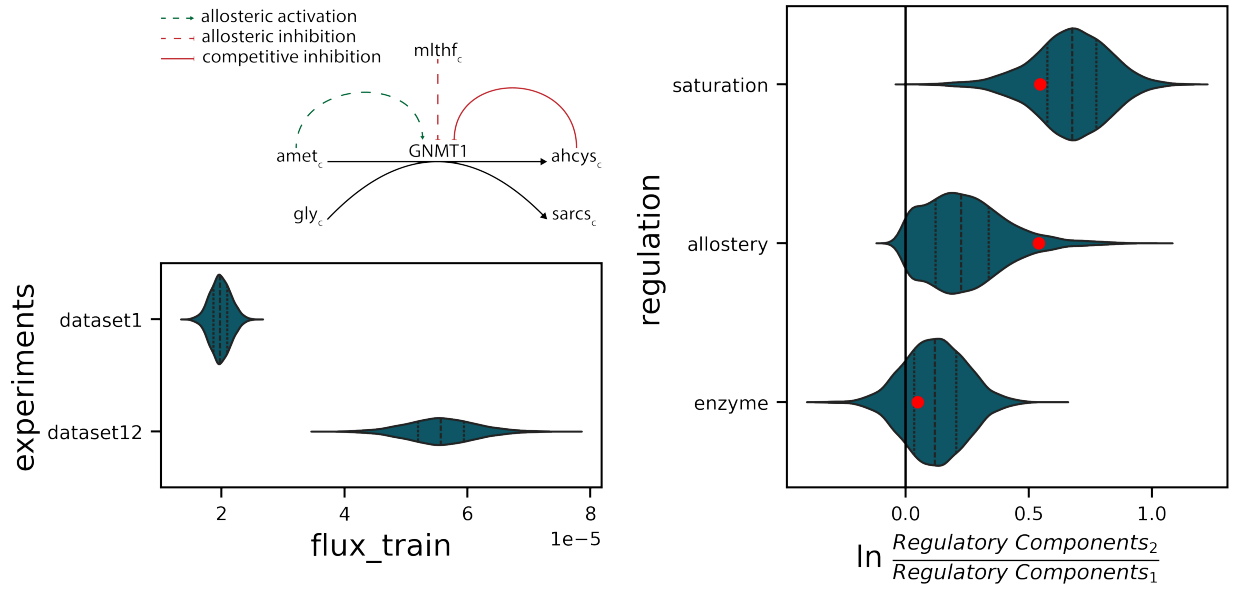

Figure 5: Illustration of how analysing a system with Maud can yield actionable insights about the underlying metabolic network. (**Top Left**) A schematic of the regulatory interactions associated with the enzyme $GNMT1$. Dashed green lines represents allosteric activation, dashed red lines indicate allosteric inhibition and solid red lines represent competitive inhibition. (**Bottom Left**) Comparison of marginal posterior distributions for $GNMT$ flux in datasets 1 and 2. (**Right**) Log-scale ratios of the regulatory elements defined in equation (5). Note that the *Reversibility* and k_cat components are excluded: this is because this reaction was modelled as irreversible, and k_{cat} was modelled as constant across datasets. These plots identify why flux in dataset 12 is higher than in dataset 1: the flux increase is due to allostery and saturation with no control from enzyme concentration changes.

interface provides commands for inference, simulation and making out-of-sample predictions. Results are stored in files, using a structured, interoperable format.

3.1 Input format

Maud inputs are structured directories, somewhat inspired by the PEtab format (*20*). A Maud input directory must contain a toml (*21*) file called `config.toml` which gives the input a name, configures how Maud will be run and tells Maud where to find the other files, allowing these to have custom names. It must also include a file containing a kinetic model definition, a file specifying information about parameters and a file with information experiments. The required structure of these files is documented at <https:// maud-metabolic-models.readthedocs.io/en/latest/inputting.html>. The input is validated against a Pydantic (*22*) data model which can be inspected at https://github.com/biosustain/Maud/tree/main/maud/data_model.

We chose to implement a custom input format despite the existence of standard formats in similar areas, including SBML (*23*) and PEtab (*20*). This choice was partly motivated by the need to ensure flexibility as Maud was developed, but there are also features of SBML and PEtab that make them structurally unsuitable in this context. Our requirements for an input format included that it be mathematics-free, so that all mathematical details are encapsulated in source code, and that it has a detailed, verifiable structure. These requirements made toml more attractive than SBML: toml is easier for humans to read and edit and can straightforwardly be validated using tools like Pydantic. Further, an SBML representation of our desired input would not contain differential equations. It would therefore not be interoperable with most SBML targeting software, which typically assumes that differential equations are available and does not know about Maud's structure.

3.2 Kinetic model

Maud's kinetic model decomposes into factors as shown in equation (5).

$$F(C;\theta) = Enzyme \cdot k_{cat} \cdot Reversibility \cdot Saturation \cdot Allostery \tag{5}$$

Each of the terms on the right-hand side of (5) is a function of C and θ. This idea is taken from Noor et al. (*24*). The terms usefully gather physically meaningful and conceptually distinct factors contributing to reaction fluxes. *Enzyme* captures the effect of enzyme concentration, k_{cat} that of enzyme efficiency, *Reversibility* quantifies thermodynamic effects, *Saturation* the effect of enzyme availability and *Allostery* the effect of post translational modifications.

We used the model of enzyme saturation from Liebermeister et al. (*25*) and the generalised Monod-Wyman-Changeux model of Allosteric regulation introduced in (*26–29*) and used more recently in Matos et al. (*30*). To capture the effect on enzyme activity of coupled phosphorylation and dephosphorylation processes we developed a new mathematical model inspired by the generalised MWC model of allosteric regulation. Full details of all mathematical aspects of Maud's kinetic model can be found in supplementary material section (REFERENCE).

3.3 Statistical model

Maud represents information from measurements using generalised linear regression models that probabilistically connect realised measurements with true values of the measurable quantities. Information from other sources is represented using a prior distribution over a set of latent parameters. The parameters and the measurable quantities are connected by a generative model encompassing Maud's kinetic model as well as the steady state equation (3). Together, the measurement model, prior model and generative model determine a

joint probability function that assigns a probability density to any possible combination of measurements and parameters.

Below we describe the prior and measurement models, as the generative model has already been discussed above.

3.3.1 Prior model

Maud's prior model includes unknown parameters corresponding to quantities in the kinetic model that are assumed to be unknown, other than steady state metabolite concentrations and fluxes, which are derived from the values of other parameters by solving the steady state problem. See table (REFERENCE) in this paper's supplementary materials for a description of all these parameters and their dimensions. Note that some quantities in Maud's kinetic model are not treated as parameters: for example, temperatures, compartment volumes and the formation energy of water. Maud treats these quantities as if they were known precisely: they can be configured by the user or default values can be used. Although in practice there can be considerable uncertainty regarding these quantities, we chose to disregard this uncertainty in the interest of simplicity.

Except for metabolites' standard condition Gibbs energy changes of formation, Maud uses independent normal prior distributions for parameters that can in principle be both negative and positive. For parameters that are constrained to be positive, Maud uses independent log-normal distributions. Formation energy parameters have a multivariate normal prior distribution. Location, scale and covariance parameters for all these prior distributions can be selected freely by the user.

3.3.2 Measurement model

Maud's measurement model considers three types of measurement: metabolite concentration measurements, enzyme concentration measurements and flux measurements, represented by vectors y^{conc} y^{enz} and y^{flux} respectively.

All measurements are specific to an experimental condition; that is, a case where the true state of the network, including knockouts, boundary conditions and state variables as well as kinetic and thermodynamic parameters, can safely be assumed to be the same. Maud's statistical model allows for arbitrarily many experimental conditions, and for any measurable quantity to be measured any number of times in any condition.

Metabolite and enzyme measurements are intended to represent the results of quantitative metabolomics and proteomics experiments. The likelihood functions for such measurements are shown in equations Equation (6) and Equation (7).

$$y_i^{conc} \sim LN(\ln \hat{y}_i^{conc}, \sigma_i^{conc}) \tag{6}$$

$$y_i^{enz} \sim LN(\ln \hat{y}_i^{enz}, \sigma_i^{enz}) \tag{7}$$

Both equations are log-normal generalised linear models with a standard link function (the natural logarithm ln) and known standard deviation σ_{conc}. The use of this measurement model is motivated by the consideration that concentrations are constrained to be non-negative, so the measurement model should avoid assigning positive probability mass to negative metabolite concentration values. In addition, we expect the precision of most metabolomics and proteomics experiments to be roughly proportional to the value of the true measured quantity, which supports a measurement model with constant coefficient of variation. The measurement standard deviations σ_{conc} and σ_{enz} are assumed to be known exactly for simplicity; plausible values can be elicited by considering the likely coefficient of variation of the measuring apparatus.

Flux measurements, representing the results of quantitative fluxomics analyses, are modelled using a likelihood function from a standard linear regression model, as shown in equation (8). Flux measurements can be obtained from the results of isotope labelling experiments using

metabolic flux analysis, for example as described in (Young 2014). When entering flux measurements, it is important only to specify measurements for a network's free fluxes, as the values of some steady state fluxes in a metabolic network are constrained by others, with the result that dependent fluxes cannot typically be measured separately. If measurements of multiple dependent fluxes are entered, information will inappropriately be double counted.

$$y_i^{flux} \sim LN(\ln \hat{y}_i^{flux}, \sigma_i^{flux}) \tag{8}$$

Our measurement model improves on analyses of metabolomics and proteomics data that assume a regression model with normally distributed errors, whether explicitly using a standard linear model or implicitly using ordinary least squares fitting. This assumption is undesirable because it implies that the measured quantity could in principle be negative, and assumes an additive underlying random process, whereas multiplicative processes tend to better describe real concentration data.

The use of independent measurement models for metabolite, enzyme and flux measurements carries an implicit assumption that there are no systematic correlations in the measurement errors. This choice was motivated by simplicity - it would be better to use a model with potentially correlated measurements. Similarly, it would be preferable to include measurement errors as model parameters, thereby avoiding possible bias due to incorrect assessments of measurement accuracy. However, we chose to use a simpler measurement model to avoid the complexity and potential fitting issues that these changes would entail.

Finally, the reader may wonder why Maud uses a linear regression model for reaction flux measurements even though this creates the potential for erroneous double counting and requires non-trivial upstream modelling, as intracellular fluxes typically cannot be measured directly. Instead, fluxes are typically inferred from isotope labelling experiments using metabolic flux analysis: see Dai and Locasale (*31*) for more about this method. Ideally Maud's measurement

model for fluxes would extend from fluxes to the results of potential labelling experiments, thereby removing the need for upstream analysis and avoiding any double counting. This option has not yet been pursued, again for the sake of simplicity.

3.4 Implementation

Maud uses the Python library click (*32*) to implement a command line interface. The command line interface loads input files as Python dictionaries, which are parsed using the Python library `toml` (*33*) and then validated and converted into structured `MaudInput` objects using Pydantic (*22*). Maud's statistical model is implemented in the probabilistic programming language Stan (*34*) and accessed using the interface cmdstanpy (*35*). For posterior sampling, Maud uses the `MaudInput` to create a Stan input json file and obtain configuration information for cmdstanpy, which it uses to (if necessary) generate a model executable file and then trigger posterior sampling using adaptive Hamiltonian Monte Carlo. When sampling is complete, Maud converts to the output into the standard format `InferenceData` using the Python library arviz (*36*) and saves it as a json file, along with some information for debugging.

Two details of Maud's implementation are important to highlight: the method of posterior sampling and how Maud solves the steady state problem (3).

3.4.1 Posterior sampling

Although integrals of the joint probability model for kinetic models are typically analytically intractable, they can be approximated numerically using Markov Chain Monte Carlo (MCMC) and other methods. Maud uses MCMC primarily because there exist many methods for verifying that MCMC samples really do approximate the target probability distribution: see Vehtari et al. (*11*) and Talts et al. (*37*) for discussion of this point. In addition, there are several examples of successful Bayesian kinetic modelling projects using MCMC including St. John et al. (*14*) and Xing et al. (*38*).

3.4.2 Solving the steady state problem

In order to implement adaptive Hamiltonian Monte Carlo, Maud must repeatedly solve the steady state problem (3) and find its gradients with respect to all parameters. To achieve this, Maud uses a hybrid method involving two numerical solvers from the SUNDIALS suite (*39*): CVODES and IDAS. These solvers are accessed via their interface from Stan. The hybrid method follows that proposed by Margossian (*40*), and involves numerically evolving the ODE system for a short period of time, then using the difference between the evolved and starting concentrations as the target for a numerical algebra solver.

4 Supporting information

The document `supplementary.pdf` contains the following information:

- Description of Maud's kinetic model including parameter dimensions and rate equations.
- Details of the procedure used to generate our case study results, including generating the artificial case study datasets, specifying priors and carrying out computation.
- Specification of the prior distribution used for the case studies

5 Acknowledgements

This research was funded by the Novo Nordisk foundation using grant number NNF14OC0009473.

6 References

1. Christodoulou, D., Link, H., Fuhrer, T., Kochanowski, K., Gerosa, L., and Sauer, U. (2018) Reserve Flux Capacity in the Pentose Phosphate Pathway Enables Escherichia

coli's Rapid Response to Oxidative Stress. *Cell Systems 6*, 569–578.e7.

2. DeBerardinis, R. J., and Chandel, N. S. (2016) Fundamentals of cancer metabolism. *Science Advances 2*, e1600200.

3. Liberti, M. V. et al. (2017) A Predictive Model for Selective Targeting of the Warburg Effect through GAPDH Inhibition with a Natural Product. *Cell Metabolism 26*, 648–659.e8, tex.pmid: 28918937 tex.publisher: Cell Press.

4. Saa, P. A., and Nielsen, L. K. (2016) Construction of feasible and accurate kinetic models of metabolism: A Bayesian approach. *Scientific Reports 6*, 29635.

5. Gopalakrishnan, S., Dash, S., and Maranas, C. (2020) K-FIT: An accelerated kinetic parameterization algorithm using steady-state fluxomic data. *Metabolic Engineering*

6. Gutenkunst, R. N., Waterfall, J. J., Casey, F. P., Brown, K. S., Myers, C. R., and Sethna, J. P. (2007) Universally sloppy parameter sensitivities in systems biology models. *PLoS Computational Biology 3*, 1871–1878, tex.pmcid: PMC2000971 tex.sciwheelprojects: Relative Metabolomics.

7. Raue, A., Becker, V., Klingmüller, U., and Timmer, J. (2010) Identifiability and observability analysis for experimental design in nonlinear dynamical models. *Chaos: An Interdisciplinary Journal of Nonlinear Science 20*, 045105.

8. Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2020) Bayesian Data Analysis, Third Edition. 656.

9. Gelman, A., Vehtari, A., Simpson, D., Margossian, C. C., Carpenter, B., Yao, Y., Kennedy, L., Gabry, J., Bürkner, P.-C., and Modrák, M. (2020) Bayesian Workflow. *arXiv:2011.01808 [stat]*

10. White, A., Tolman, M., Thames, H. D., Withers, H. R., Mason, K. A., and Transtrum, M. K. (2016) The limitations of model-based experimental design and param-

eter estimation in sloppy systems. *PLoS Computational Biology 12*, e1005227, tex.pmcid: PMC5140062 tex.sciwheel-projects: Relative Metabolomics.

11. Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., and Bürkner, P.-C. (2021) Rank-Normalization, Folding, and Localization: An Improved R^ for Assessing Convergence of MCMC (with Discussion). *Bayesian Analysis 16*, 667–718.

12. Raue, A., Kreutz, C., Theis, F. J., and Timmer, J. (2013) Joining forces of Bayesian and frequentist methodology: a study for inference in the presence of non-identifiability. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 371*, 20110544.

13. Stapor, P., Weindl, D., Ballnus, B., Hug, S., Loos, C., Fiedler, A., Krause, S., Hroß, S., Fröhlich, F., Hasenauer, J., and Wren, J. (2018) PESTO: parameter estimation toolbox. *Bioinformatics 34*, 705–707.

14. St. John, P., Strutz, J., Broadbelt, L. J., Tyo, K. E. J., and Bomble, Y. J. (2018) Bayesian Inference of Metabolic Kinetics from Genome-Scale Multiomics Data. *bioRxiv*

15. Visser, D., and Heijnen, J. J. (2003) Dynamic simulation and metabolic re-design of a branched pathway using linlog kinetics. *Metabolic Engineering 5*, 164–176.

16. Liebermeister, W., and Noor, E. (2021) Model Balancing: A Search for In-Vivo Kinetic Constants and Consistent Metabolic States. *Metabolites 11*, 749.

17. Korendyaseva, T. K., Kuvatov, D. N., Volkov, V. A., Martinov, M. V., Vitvitsky, V. M., Banerjee, R., and Ataullakhanov, F. I. (2008) An Allosteric Mechanism for Switching between Parallel Tracks in Mammalian Sulfur Metabolism. *PLoS Computational Biology 4*, e1000076.

18. Poirier, D. J. (1998) REVISING BELIEFS IN NONIDENTIFIED MODELS. *Econometric Theory 14*, 483–509.

19. Van Rossum, G., and Drake, F. L. *Python 3 Reference Manual*; CreateSpace: Scotts Valley, CA, 2009.

20. Schmiester, L. et al. (2021) PEtab—Interoperable Specification of Parameter Estimation Problems in Systems Biology. *PLOS Computational Biology 17*, 1–10.

21. Preston-Werner, Tom and Gedam, Pradyun TOML Specification 1.0.0-Rc.1. 2020; https://toml.io/en/v1.0.0-rc.1/.

22. Pydantic developers Pydantic. 2022; https://pypi.org/project/pydantic/.

23. Keating, S. M. et al. (2020) SBML Level 3: An Extensible Format for the Exchange and Reuse of Biological Models. *Molecular Systems Biology 16*, e9110.

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

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

26. Monod, J., Wyman, J., and Changeux, J. P. (1965) On the nature of allosteric transitions: a plausible model. *Journal of Molecular Biology 12*, 88–118.

27. Changeux, J.-P. (2013) 50 years of allosteric interactions: the twists and turns of the models. *Nature Reviews. Molecular Cell Biology 14*, 819–829, tex.sciwheel-projects: Yeast.

28. Popova, S. V., and Sel'kov, E. E. (1975) Generalization of the model by Monod, Wyman and Changeux for the case of a reversible monosubtrate reaction SR,TP. *FEBS Letters 53*, 269–273.

29. Popova, S. V., and Sel'kov, E. E. (1979) [Description of the kinetics of the two substrate reactions S1+S2 goes to and comes from S3+S4 by a generalized Monod, Wyman, Changeux model]. *Molekuliarnaia Biologiia 13*, 129–139.

30. Matos, M. R. A., Saa, P. A., Cowie, N., Volkova, S., de Leeuw, M., and Nielsen, L. K. (2022) GRASP: A Computational Platform for Building Kinetic Models of Cellular Metabolism. *Bioinformatics Advances 2*, vbac066.

31. Dai, Z., and Locasale, J. W. (2017) Understanding Metabolism with Flux Analysis: From Theory to Application. *Metabolic Engineering 43*, 94–102.

32. Click Developers Click: Python Composable Command Line Interface Toolkit. Pallets, 2022; https://pypi.org/project/click/.

33. Pearson, W. Toml: Python Library for Tom's Obvious, Minimal Language. 2020; https://pypi.org/project/toml/.

34. Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A. (2017) Stan: A Probabilistic Programming Language. *Journal of Statistical Software 76*, 1–32.

35. Stan Development Team CmdStanPy. 2022; https://github.com/stan-dev/cmdstanpy.

36. Kumar, R., Carroll, C., Hartikainen, A., and Martin, O. (2019) ArviZ a Unified Library for Exploratory Analysis of Bayesian Models in Python. *Journal of Open Source Software 4*, 1143.

37. Talts, S., Betancourt, M., Simpson, D., Vehtari, A., and Gelman, A. (2018) Validating Bayesian Inference Algorithms with Simulation-Based Calibration. *arXiv:1804.06788 [stat]*

38. Xing, Z., Bishop, N., Leister, K., and Li, Z. J. (2010 Jan-Feb) Modeling Kinetics of

a Large-Scale Fed-Batch CHO Cell Culture by Markov Chain Monte Carlo Method. *Biotechnology Progress 26*, 208–219.

39. Serban, R., and Hindmarsh, A. C. CVODES: The Sensitivity-Enabled ODE Solver in SUNDIALS. Volume 6: 5th International Conference on Multibody Systems, Nonlinear Dynamics, and Control, Parts A, B, and C. Long Beach, California, USA, 2005; pp 257–269.

40. Margossian, C. Computing Steady States with Stan's Nonlinear Algebraic Solver. Stan Conference 2018. 2018.