Contents lists available at ScienceDirect

Chemical Engineering Science

journal homepage: www.elsevier.com/locate/ces

Machine learning uncovers analytical kinetic models of bioprocesses

Tim Forster[a], Daniel Vázquez[b], Claudio Müller[a], Gonzalo Guillén-Gosálbez[a,*]

[a] Department of Chemistry and Applied Biosciences, Institute for Chemical and Bioengineering, ETH Zurich, Vladimir-Prelog-Weg 1, 8093 Zurich, Switzerland
[b] IQS School of Engineering, Universitat Ramon Llull, Via Augusta 390, 08017 Barcelona, Spain

ARTICLE INFO

Keywords:
Bioprocess
Symbolic regression
Optimization

ABSTRACT

Identifying suitable kinetic models for bioprocesses is a complex task, particularly when interpretable models are sought. Classical machine learning algorithms are gaining wide interest to simulate complex bioprocesses that are hard to describe via first principles. However, they often rely on *a priori* assumptions of the model structure and lead to mathematical expressions that are hard to interpret. In this work, we apply an alternative approach based on symbolic regression to identify bioprocess models without assuming a pre-defined model structure. We obtain algebraic expressions for the kinetic rates from data consisting of concentration profiles. The model training was performed following a two-step approach that allows avoiding the iterative integration of differential equations for the parameter estimation step. The proposed procedure was found from numerical examples to slightly outperform neural network benchmarks. Moreover, the obtained algebraic expressions for the rate equations facilitate the model interpretation and enable the direct application of optimization algorithms.

1. Introduction

In recent years, modelling has gained significant attention in the bioprocesses industry, spearheaded by the improvements in mathematical tools that can be used for analysis and optimization (Mowbray et al., 2023; Narayanan et al., 2021). Mathematical modelling can support scientists, engineers, or other subject matter experts in designing experiments (Sadino-Riquelme et al., 2020), predicting and monitoring processes (Del Rio-Chanona et al., 2019; Rivera et al., 2007), and reducing development and production costs (Narayanan et al., 2021, 2020). Modelling complex bioprocesses, however, is a challenging task, particularly when first principles formulations are sought (Mercier et al., 2014; Petsagkourakis et al., 2020; Zhang et al., 2020). These models are nevertheless being increasingly demanded by the market, in which the number of new products originating from bioprocesses is increasing very rapidly (Narayanan et al., 2023).

Bioprocess modelling requires experimental measurements to calibrate an *in-silico* model by minimizing the mismatch between experimental observations and *in-silico* predictions. A common approach relies on well-established mathematical formalisms derived from first principles, such as mass or energy balances. Kroll et al. (2017) provide a workflow for the generation of mechanistic process models, where the authors start from material balances for a certain target variable and expand the models in a mechanistic manner with new states and interactions. They used their method in a mammalian cell culture process to model the viable cell count. A more recent work by Sha et al. (2018) provides stoichiometric and kinetic models and some commonly used mathematical approaches to describe cell systems.

An alternative to purely mechanistic modelling approaches are data-driven strategies. These methods enable model building without relying on expert knowledge (Kahrs and Marquardt, 2007; Taylor et al., 2021). Here, the structure of the model is given by the surrogate modelling approach of choice. For example in the area of process control, Willis et al. (1995) applied an artificial neural network (ANN) to model the biomass concentration in a fermentation process. In a more recent work, Tonner et al. (2017) used Gaussian process models to describe the microbial growth in bioprocesses and interrogated the obtained models to investigate perturbation effects in the systems under study. As a bridge between purely deterministic and purely data-driven methods, hybrid modelling approaches, where mechanistic knowledge is combined with a surrogate component, have also gained popularity (von Stosch et al., 2014). This approach has been applied to a wide range of problems in science and engineering. For example, Zhang et al. (2013) proposed a hybrid kinetic mechanism where quasi-steady-state species are separated from the kinetic ODEs. Gnoth et al. (2010, 2008, 2007) integrated ANNs in kinetic models to approximate unknown behaviours of the microorganisms. More recently, hybrid frameworks for modelling bioprocesses have been put forward by Zhang et al. (2019), and Mowbray (2023) and colleagues. Moreover, in earlier works (Forster et al.,

[*] Corresponding author.
E-mail address: gonzalo.guillen.gosalbez@chem.ethz.ch (G. Guillén-Gosálbez).

https://doi.org/10.1016/j.ces.2024.120606
Received 6 February 2024; Received in revised form 8 July 2024; Accepted 7 August 2024
Available online 8 August 2024

Nomenclature

Abbreviations

ANN	Artificial neural network
BMS	Bayesian machine scientist
LHS	Latin hypercube sampling
ODE	Ordinary differential equation
SR	Symbolic regression

Sets

E	$\{e\mid$ Set of mathematical expressions$\}$
I, J	$\{i, j\mid$ Set of components$\}$
U	$\{u\mid$ Set of discrete sample points$\}$

Parameters

t_0, t_f	Initial and final time
$X_{0,i}$	Initial concentration of metabolite/species i
μ	Mean of a particular property
σ^2	Variance of a particular property
γ_e	Mathematical expression identified by the BMS
θ	Generic model parameters

Variables

BMS_i and ANN_i	BMS or ANN models for species i
p	Probability
Rxn_i	Generic reaction term (production or consumption of species i)
X_i	Concentration of metabolite/species i (used as continuous variables in ODE expressions)
$X_{i,u}$	Concentration of metabolite/species i at time t_u
$\widehat{X}_{i,u}$	Model predictions of the concentration of metabolite/species i at time t_u
$\dot{X}_{i,u}$	Derivatives of metabolite/species i at time t_u
$\widehat{\dot{X}}_{i,u}$	Model predictions of the derivatives of metabolite/species i at time t_u
$\overline{X}_{i,u}$	Mean of the experimental data points of species i at time t_u
t and t_u	Time and sampled time point
ζ_i	Function to smooth noisy concentration profile for species i
$\dot{\zeta}_i$	Derivative of function to approximate derivative profile for species i
$\mathscr{DL}$	Description length (objective function of the BMS)

2023a), a method for building models that are based on canonical kinetic representations (i.e., S-system (Savageau, 1970, 1969a, 1969b)) was studied, where observed concentration data and a pre-defined canonical form for the rate expression were used to identify a suitable model structure and simultaneously estimate its parameters.

A key point in all the modelling approaches above is to define the model structure whose parameters will be calibrated via parameter estimation methods. Ideally, the model structure and its parameters should be simultaneously determined, since the choice of a specific model structure limits the accuracy of the model. However, in practice the model structure is first defined, hopefully through a mechanistic derivation of first principles, but sometimes through a surrogate formalism. Once the structure is chosen, its parameters are calibrated by solving a parameter estimation problem where the parameter values are the decision variables, and the objective function is often given by the mismatch between *in-silico* and experimental observations. Works that optimize both the model structure and its parameters are quite scarce. A well-known example in the Process Systems Engineering (PSE) literature is the ALAMO approach for the automated learning of algebraic models (Wilson and Sahinidis, 2017). This algorithm creates closed-form surrogate models by solving a mixed-integer programming (MIP) problem where binary variables model the selection of specific algebraic terms from a set of allowable functions and continuous ones the associated parameters. Designed for dynamic systems, Brunton et al. (2016) proposed the SINDy (Sparse Identification of Nonlinear Dynamics) algorithm, which was successfully applied to different systems. By using sparse regression techniques, SINDy provides the user with an appropriate rate model for the ODE. Sun and Braatz (2020) developed an algorithm that combines nonlinear feature generation followed by sparse regression to learn interpretable nonlinear models, called algebraic learning via elastic net (ALVEN). Other works, such as those by Willis and von Stosch (Willis and von Stosch, 2017), use a problem-tailored approach for extracting ODEs from process data by formulating a hybrid semi-parametric modelling framework using mixed integer programming and multivariate rational functions. These modelling methods have the advantage of only requiring data and, therefore, can be set up without any expert knowledge about the system. Nonetheless, they assume a set of basis functions that must be combined linearly to form the algebraic expressions sought, which constrains the feasible set of plausible mathematical models that could explain given data.

Another approach for identifying closed-form expressions is symbolic regression (SR), which is based on the principles of genetic programming (Keane et al., 1993; Koza, 1994). In contrast to the main tools mentioned above, such as ALAMO, SINDy, or ALVEN, SR methods represent mathematical equations as expression trees (Cozad and Sahinidis, 2018). Employing a defined search procedure (i.e., mainly stochastic algorithms (Diveev and Shmalko, 2021) like a genetic algorithm (Cranmer et al., 2020) or Markov-Chain Monte Carlo (Guimerà et al., 2020)), SR simultaneously identifies the tree structure and involved parameters in order to optimally represent observed data (Cozad and Sahinidis, 2018; Neumann et al., 2020). While previous approaches specified the basis functions, SR only requires a pool of allowed operators, and the functions are created from the available pool and given data. SR has been successfully applied in various fields, for example, McKay et al. (1997) used an SR approach to model a vacuum distillation column and a chemical reactor system. In a later work, the authors applied SR to develop a model of a food extrusion process (McKay et al., 1999). Vladislavleva et al. (2013) used an available software package named DataModeler (2023) to predict energy outputs of wind farms by considering weather data. Schmidt and Lipson (2009) discovered physical laws from experimental data using SR to identify nonlinear relationships. In recent contributions, researchers used SR to discover new perovskite catalysts (Weng et al., 2020) and to recover a variety of physical expressions (Udrescu and Tegmark, 2019). Other works resulted in commercially available SR software, such as Eureqa (Schmidt and Lipson, 2009) or TuringBot (2023). Cranmer et al. (2020) implemented an open-source SR algorithm named PySR (Cranmer, 2020) in Python that was applied to cosmology problems. Similarly, Guimerà et al. (2020) developed the Bayesian machine scientist (BMS), a SR algorithm based on a Markov-Chain Monte Carlo approach. These approaches were applied in kinetic modelling for heterogeneous catalysis (de Servia et al., 2023), process design (Ferreira et al., 2019a, 2019b; Negri et al., 2022), process optimization (Forster et al., 2023b; Forster et al., 2023c) or to model links between energy-related impacts and socioeconomic drivers (Vázquez et al., 2022).

Here, we apply SR techniques for kinetic model building in bioprocesses. In contrast to previous works that developed fully black-box or hybrid models based on standard surrogates (e.g., ANN and GPs) (Del Rio-Chanona et al., 2019; Gnoth et al., 2010), here we apply SR to find a suitable kinetic expression and associated parameters. Specifically, our approach combines the BMS with a two-step decomposition algorithm inspired by the works of Miró (2014), Voit and Almeida

(2004), Michalik et al. (2009), and Brendel et al. (2006). The goal is to identify reaction rates from observed concentration profiles of species, where the rate equation is determined via SR. de Servia et al. (2023) recently applied also SR using pySR (Cranmer, 2020) for heterogeneously catalyzed reactions. However, we here focus on bioprocesses and instead use the BMS for SR (Guimerà et al., 2020). Numerical examples show that the BMS can identify closed-form surrogate rate expressions and lead to a similar performance relative to ANN-benchmark models. Following the successful application of the BMS in other problems, including the approximation of process simulations (Negri et al., 2022), process optimization (Forster et al., 2023b), and the investigation of energy-related impacts and socioeconomic drivers in macro-economic studies (Vázquez et al., 2022), here we show that it can also be used to find kinetic expressions that explain given data precisely.

The remainder of this article is organized as follows: First, the problem statement is described in detail. Subsequently, the proposed methodology is discussed. Afterward, the case studies are introduced, and the results are summarized. Finally, the conclusions of the work are drawn.

2. Problem statement

Without loss of generality, in this work, we consider a generic ideal batch reactor with constant volume V and different species $i \in I$ taking part in some reactions. The mass balance of such a system can be described by expression (1). In this description, X_i might be the concentration of microbial cells or of a given species in the bioreactor, and $X = [X_1, X_2, \cdots, X_i]$ represents the vector of all metabolite concentrations. On the left-hand side of the equation, dX_i/dt (or $\dot{X}$), refers to the accumulation term.

$$\frac{d}{dt}X_i = \dot{X}_i = Rxn_i(X), \quad \forall i \in I \tag{1}$$

The $Rxn_i(X)$ term represents an expression that is unknown to the modeler and that depends on the concentration of all the species (state variables) collected in vector X. This is a common situation arising in bioprocess development, because the underlying metabolic pathways in such systems can be very complex (Guillén-Gosálbez et al., 2013; Mercier et al., 2014; Petsagkourakis et al., 2020; Zhang et al., 2020). This complexity is given by the potentially large feedback loops between a wide range of species and the nonlinear nature of these interactions. In this work, we will approximate $Rxn_i(X)$ using a symbolic regression method that generates an algebraic expression without assuming any pre-defined structure of that reaction rate. Hence, here we do not rely on any canonical formalism to derive the kinetic model.

The goal, then, is to find a suitable expression for $Rxn_i(X)$ in Eq. (1) such that the mismatch between the model predictions and the experimental observations is minimized. Note that in this work, we assume that neither the structure of $Rxn_i(X)$ nor the involved parameters are known, unlike in a standard parameter estimation problem as discussed by Voit and Almeida (2004) or Brendel et al. (2006). Therefore, herein, we aim to find both, the rate expressions and their parameters simultaneously by only using the available concentration measurements. It is worth to mention that in the subsequently proposed approach, the modelling of a rate in the form $Rxn_i(X)$ for a species i is only possible for species that can be measured in the sampled data. If no data is available for species i, a parameter estimation and, therefore, a model building for such a species is not directly possible. Such a case might be encountered if some species have a shorter lifetime than the sampling frequency. Consequently, our modelling approach focuses on species that can be sampled, not on non-sampled or hidden species. The section that follows introduces our approach.

3. Methodology

In time-series-related problems, the concentrations (subsequently also called states) X_i are often considered to be continuous in time, i.e., $X_i(t)$. However, usually only discrete concentration values are available at the sampling times. Therefore, we consider a discrete notation based on a series of time points $u \in U$. The complete profile of one species i can therefore be described by expression (2).

$$X_{i,u} \in \left[X_{i,0}, X_{i,1}, X_{i,2}, \cdots, X_{i,|U|}\right], \quad \forall i \in I \tag{2}$$

From such a sampled array, we are interested in searching for a suitable model for the rate expression that can predict the time-dependent evolution by using the initial conditions at time t_0. This model-building task is typically formulated as a general dynamic optimization problem. In such an optimization problem, the sum of squared residuals (SSR) between the observed data point $X_{i,u}$ and the model prediction $\widehat{X}_{i,u}$ is minimized, by optimizing the values of some unknown model parameters θ. The problem can therefore be formulated as given in (3):

$$\begin{aligned} \min_{\beta} SSR &= \sum\nolimits_{i\in I}\sum\nolimits_{u\in U}\left(X_{i,u} - \widehat{X}_{i,u}\right)^2 \\ s.t. \widehat{\dot{X}}_{i,u} &= \mathscr{M}_i\left(X_{j,u}, \theta\right), \forall i \in I, j \in I, u \in U \\ \widehat{X}_{i,0} &= X_{i.0}, \forall i \in I \\ &= \widehat{X}, \widehat{\dot{X}} \in \mathbb{R}^+ \end{aligned} \tag{3}$$

In (3), the predicted derivative $\widehat{\dot{X}}_{i,u}$ of species i at time point u is calculated by a model $\mathscr{M}_i(X, \theta)$ with some trainable parameters θ that well approximate the underlying reaction rate $Rxn_i(X)$. The model building process to approximate $Rxn_i(X) \approx \mathscr{M}_i(X, \theta)$ is discussed below. The initial conditions $X_{i,0}$ are usually known values. However, finding the concentration profiles $X_{i,u}$ for a given system requires solving the ODEs, either simultaneously or sequentially. In this context, stiff ODEs can often make numerical integration very difficult and inefficient (Tjoa and Biegler, 1991). Moreover, effectively handling the existence of binary variables in this approach would remain challenging.

Michalik et al. (2009) and Voit and Almeida (2004) proposed alternative approaches to simplify the dynamic problem shown above based on a reformulation of the original model in the derivative space instead of the state space. This reformulation is given in (4).

$$\begin{aligned} \min_{\beta} SSR &= \sum\nolimits_{i\in I}\sum\nolimits_{u\in U}\left(\dot{X}_{i,u} - \widehat{\dot{X}}_{i,u}\right)^2 \\ s.t. \widehat{\dot{X}}_{i,u} &= \mathscr{M}_i\left(X_{j,u}, \theta\right), \forall i \in I, j \in I, u \in U \\ \widehat{X}_{i,0} &= X_{i.0}, \forall i \in I \\ &= \widehat{X}, \widehat{\dot{X}} \in \mathbb{R}^+ \end{aligned} \tag{4}$$

To solve the problem given in (4), the derivatives $\dot{X}_{i,u}$ have to be obtained from the discrete time profiles of the observed state variables $X_{i,u}$. Such derivates can then be subsequently used to train a suitable kinetic model $\mathscr{M}_i(X, \theta)$. This strategy avoids integrating the dynamic system in (3), at the expense of performing the regression in the space of reaction rates, which poses some challenges concerning the computation of derivatives leading to low errors in the original dynamic space of state variables. This is because the derivatives determined experimentally can be affected by experimental errors, which may lead to good predictions in the reaction rates space but poor in the original states variables space.

The method of choice follows an incremental approach for building the surrogate model, as shown in Fig. 1, where the details of the steps are given below in section 3.1.

The discussed procedure starts with collecting noisy concentration data $X_{i,u}$ for different species i and times u. To smooth out the noise in the measurements, a univariate function in time $\zeta_i(t)$ is fitted to the data. In the second step, this identified function $\zeta_i(t)$ can be derived analytically and the derivatives $\dot{\zeta}_{i,u}$ can be evaluated at the experimental time

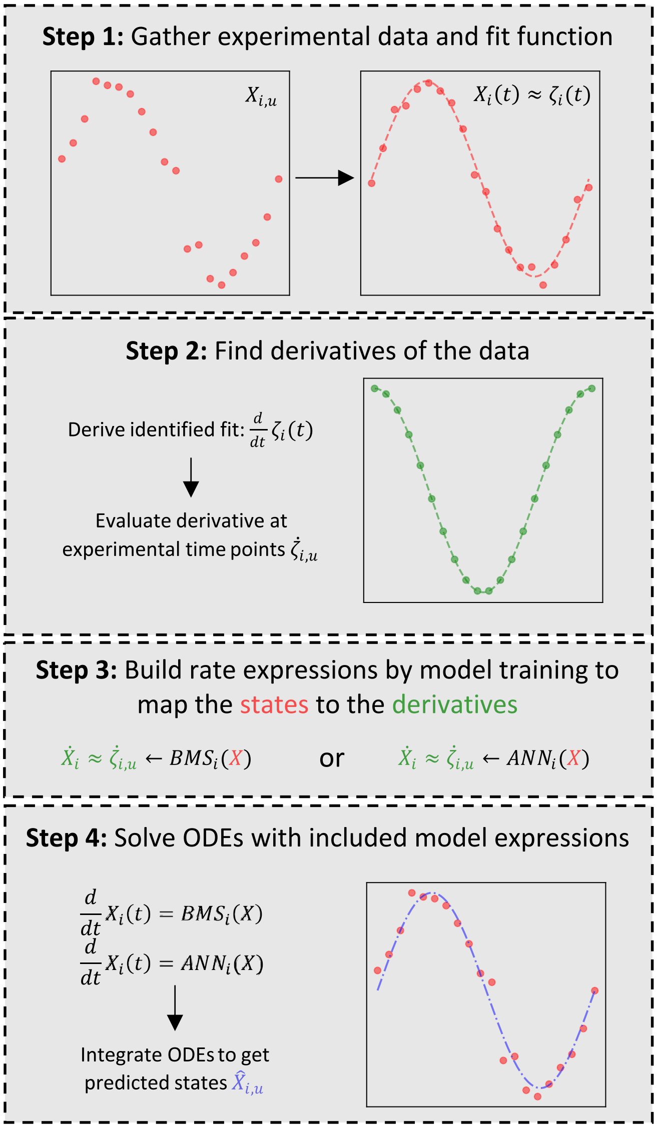
Step 1: Gather experimental data and fit function
$X_{i,u}$
$X_i(t) \approx \zeta_i(t)$
Step 2: Find derivatives of the data
Derive identified fit: $\frac{d}{dt}\zeta_i(t)$
Evaluate derivative at experimental time points $\dot{\zeta}_{i,u}$
Step 3: Build rate expressions by model training to map the states to the derivatives
$\dot{X}_i \approx \dot{\zeta}_{i,u} \leftarrow BMS_i(X)$
or
$\dot{X}_i \approx \dot{\zeta}_{i,u} \leftarrow ANN_i(X)$
Step 4: Solve ODEs with included model expressions
$\frac{d}{dt}X_i(t) = BMS_i(X)$
$\frac{d}{dt}X_i(t) = ANN_i(X)$
Integrate ODEs to get predicted states $\hat{X}_{i,u}$

Fig. 1. Overview of the approach for building a rate expression. In the first step, a function $\zeta_i(t)$ is fit to data points for each species i. The functions $\zeta_i(t)$ are then derived analytically (step 2). In step 3, models (BMS or an ANN) are trained to map the states to the calculated derivatives $\dot{\zeta}_{i,u}$. Last, in step 4, the models are incorporated into a system of ODEs which can be solved with appropriate initial conditions.

points. Third, the state values $X_{i,u}$ are linked to the calculated derivatives $\dot{\zeta}_{i,u}$ by an appropriate model found via SR. The model, therefore, approximates the $Rxn_i(X)$ term given in Eq. (1). Last, the trained models can be incorporated into a system of ODEs, which is solved using specific initial conditions. In the following subsection, these four steps are discussed in more detail.

3.1. Incremental approach for model building

The procedure is schematically shown in Fig. 1. There are several possible ways to derive data numerically. A comparison of three possible methods to derive a noisy sinusoidal signal is given in Fig. 2. The simplest method is the differentiation via forward finite differences. The main disadvantage of this approach is the amplification of noise during the derivation process. Therefore, a smoothing step is preferred before differentiating noisy data, for example using a Savitzky-Golay filter (Savitzky and Golay, 1964). Here, however, we used instead a polynomial or a univariate BMS to fit a function $\zeta_i(t)$ to the noisy data, as given in (5). The polynomial approach was successfully demonstrated in an earlier work by the authors (Forster et al., 2023a). The symbolic fit using the univariate BMS was inspired by de Servia et al. (2023), where the authors demonstrated an approach for fitting and deriving the observed data. In the present work, we adapted this approach and use a different toolbox. The methods that are discussed in here are implemented in Python and available on GitHub (https://github.com/forstertim/udiff).

$$X_i(t) \approx \zeta_i(t) = \begin{cases} p_{i,1} + p_{i,2}t + \cdots + p_{i,q+1}t_u^q \\ BMS_i(t) \end{cases}, \quad \forall i \in I \tag{5}$$

In the case of the polynomial approach, the unknown parameters p have to be regressed to the noisy data, while when using the BMS the structure and parameters are both to be found. Both, the polynomial and the algebraic expression identified by the BMS are univariate in time. In both cases, the resulting expressions can subsequently be derived analytically, as given in (6). The derivatives can be evaluated at the experimental time points $t_u, u \in U$.

$$\dot{X}_i(t) \approx \dot{\zeta}_i(t) = \frac{d}{dt}\zeta(t), \quad \forall i \in I \tag{6}$$

Steps 1 and 2: Fitting univariate function and estimating derivatives. In the case of the polynomial approach, we defined a set of polynomial degrees $q \in Q$. The different polynomials are fit to the noisy data and the corresponding Bayesian information criteria (BICs) are calculated. The polynomial with the lowest BIC is subsequently differentiated analytically as given above. In the case of the univariate BMS, we defined a threshold for the coefficient of determination (R^2). The BMS is trained with a given number of steps (discussed in more detail below). If the R^2-threshold is not reached, then the training steps are doubled. This procedure is repeated for a given number of times at most. After that, the identified algebraic expression can be derived analytically. As shown in Fig. 2 (b), the approximated derivatives are more accurately calculated by the smoothing methods given in (5) and (6) compared to forward finite difference differentiation. However, the first and last sample points might still comprise some error even after applying such smoothing techniques. To reduce this noise impact further, one possibility is to disregard the initial and last sample points for the subsequently discussed model training, as done in other works (Willis and von Stosch, 2017).

An in-depth analysis of how the derivative approximation methods perform under different noise levels and data set sizes is given in the supplementary information section S1. The results summarized in Fig. S1 show that the symbolic estimation approach given in (5) seems to work well suited even in presence of noise and scarce data sets.

Step 3: Building the rate expression. Rate expressions map some discrete states $X_{i,u}$ to the obtained derivatives $\dot{\zeta}_{i,u}$. The identified model, therefore, is intended to predict the changes in concentration of the species at a given time. To identify this model, we use an SR tool, the BMS. Upon model training, the BMS identifies an algebraic expression that approximates the reaction term as given in (7). To benchmark our results, we compare them to those from an ANN, as shown in (8). The reason for this choice is that ANNs are generally regarded as good approximators (Psichogios and Ungar, 1992). Additional details on the symbolic regression tool are discussed below in section 3.2.

$$Rxn_i(X) \approx \mathscr{M}_i(X) = BMS_i(X), \quad \forall i \in I \tag{7}$$

$$Rxn_i(X) \approx \mathscr{M}_i^{benchmark}(X) = ANN_i(X), \quad \forall i \in I \tag{8}$$

Step 4: Solving the ODE model with the built rate expressions. The fully trained models can be incorporated into the ODE in (1), resulting in the ODEs given in (9) for the BMS approach, and in (10) for the ANN approach.

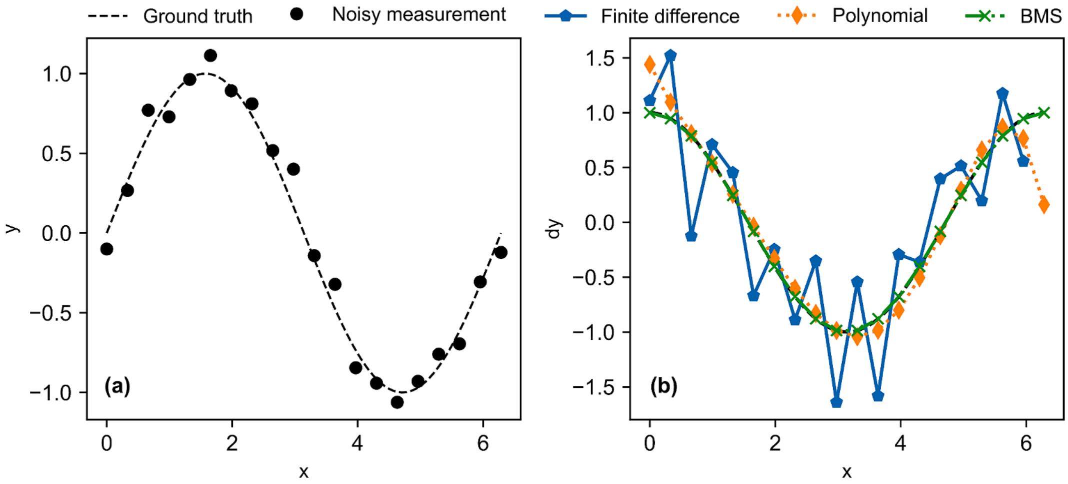

Fig. 2. (a) Noisy measurements (circles) together with the underlying sinusoidal ground truth (dashed line). (b) Comparison of numerical differentiation methods. The dashed black line represents the cosinusoidal ground truth of the derivative (covered up by the other approaches). The blue pentagons with the solid line represent the derivatives by forward finite differences. The orange diamonds with the dotted line represent the derivatives obtained by the polynomial approach discussed above. The green crosses with the dashed-dotted line represent the derivatives of the BMS approach. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$\frac{d}{dt}X_i = BMS_i(X), \quad \forall i \in I \tag{9}$$

$$\frac{d}{dt}X_i = ANN_i(X), \quad \forall i \in I \tag{10}$$

These ODEs can be solved for the initial conditions $X_{i,0}$, $i \in I$ and considering an integration period $t = [t_0, t_f]$.

3.2. Background to the Bayesian machine scientist

In this work, we do not assume any pre-defined model structure to search for suitable rate expressions. Upon model training, the BMS returns an algebraic closed-form expression, which can subsequently be incorporated into the system of ODEs to be integrated. We now provide an overview of how the BMS works. For further information, the reader is referred to the original paper (Guimerà et al., 2020). The algorithm identifies a suitable mathematical expression by searching through a space of expressions represented as symbolic trees. To perform the search through this space of expressions, several allowable moves from an initial tree can be done by the algorithm.

The space of possible mathematical expressions γ is described by E. Starting from one symbolic representation $\gamma_e, e \in E$, we perform changes in the tree leading to different mathematical expressions. One example of such a tree evolution is shown in Fig. 3 (a). The addition of the two main terms in γ_1 is replaced by a multiplication, which leads to the expression γ_2. A further replacement of the addition in γ_2 leads to the expression γ_3, which explains the observed data points (black circles) better than γ_1 or γ_2. Another adaptation would be the elementary tree replacement (i.e., exchanging the complete sub-tree $(\beta + \delta)$ by another sub-tree). For each resulting expression, a goodness-of-fit metric can be calculated. The SR algorithm then proceeds to search the space of expressions, seeking the expression with the best goodness of fit. This search is stochastic, as in other evolutionary algorithms (Costa and Oliveira, 2001; Guimerà et al., 2020).

The BMS can provide closed-form algebraic expressions from data based on a set of user-defined mathematical operations (i.e., addition, subtraction, multiplication, etc.). In the algorithm, a conditional probability $p(\gamma_e|D)$ is assigned to each expression $\gamma_e, e \in E$ (the space of symbolic trees shown schematically in Fig. 3 (b)) used to fit some data D,

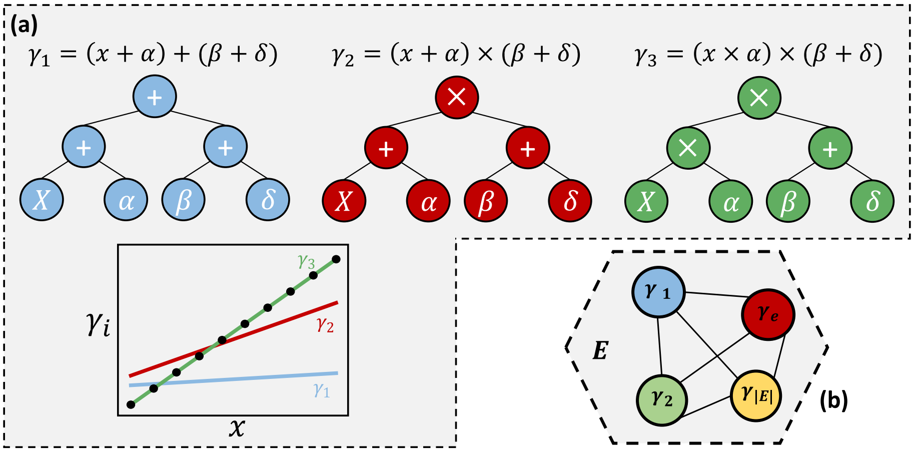
(a)
$\gamma_1 = (x + \alpha) + (\beta + \delta)$
$\gamma_2 = (x + \alpha) \times (\beta + \delta)$
$\gamma_3 = (x \times \alpha) \times (\beta + \delta)$
+
×
X
α
β
δ
γ_i
x
γ_3
γ_2
γ_1
E
γ_1
γ_e
γ_2
$\gamma_{|E|}$
(b)

Fig. 3. (a) Several equations are represented as symbolic trees. From $\gamma_1 = (x_1 + \alpha) + (\beta + \delta)$, a node replacement can be performed to reach $\gamma_2 = (x_1 + \alpha) \times (\beta + \delta)$. A further node replacement can be done to obtain the equation $\gamma_3 = (x_1 \times \alpha) \times (\beta + \delta)$. The expression for γ_3 (green line) ends in the best possible model to fit the data (black circles) compared to γ_1 (blue line) and γ_2 (red line) in the lower part of the figure. (b) The space E of all possible expressions γ_e is schematically shown as a dashed polygon. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

which is calculated according to Bayes Theorem (Bishop, 2006; Murphy, 2013), given by (11):

$$p(\gamma_e|D) = \frac{p(D|\gamma_e)p(\gamma_e)}{p(D)} \tag{11}$$

Where $p(D)$ represents the marginal likelihood of some data D. $p(D)$ is independent of γ_e and therefore acts as a normalization constant. Marginalizing over the parameters ϕ_e associated with expression γ_e (Murphy, 2013), the numerator in (11) can be expressed as an integral over the space of all possible parameter values Φ_e (Guimerà et al., 2020). The description length $\mathscr{DL}(\gamma_e)$ then describes the resulting integral (Guimerà et al., 2020; Hansen and Yu, 2001; Murphy, 2013), given in equation (12).

$$\mathscr{DL}(\gamma_e) = -log\left[\frac{1}{p(D)}\int_{\Phi_e} d\phi_e p(D|\gamma_e,\phi_e)p(\phi_e|\gamma_e)p(\gamma_e)\right] \tag{12}$$

Computing the numerical value of the integral included in the description length is challenging (Guimerà et al., 2020; Murphy, 2013). It has been shown (Grünwald, 2007; Murphy, 2013) that the entire metric can be approximated through the Bayesian information criterion (BIC) and the prior of the corresponding symbolic expression γ_e, as shown in (13):

$$\mathscr{DL}(\gamma_e) \approx \frac{BIC(\gamma_e)}{2} - log(p(\gamma_e)) \tag{13}$$

Therefore, the plausibility of observing an expression γ_e conditioned on some data D is quantified by the description length $\mathscr{DL}(\gamma_e)$. In other words, during the stochastic search, the description length (i.e., a metric measuring the plausibility of observing an expression γ_e) serves as objective function which is being minimized. As visible in expression (13), to compute the description length, the prior knowledge about expression γ_e is required as $p(\gamma_e)$. Guimerà et al. (2020) used a predefined corpus of equations from Wikipedia. After parsing the publicly available equations, the number of operations were counted that were present in the expression. Based on this information, they created distributional information about operators in equations, which were subsequently used as the prior distributions $p(\gamma_e)$ (Guimerà et al., 2020).

According to Grünwald (2007), $\mathscr{DL}(\gamma_e)$ can be understood as an encoded length of the expression γ_e (number of natural units). A Markov chain Monte Carlo (MCMC) (Hastings, 1970) algorithm is used to explore the space E of expressions, where the number of MCMC iterations is defined by the user. After evaluating the description length of each expression $\mathscr{DL}(\gamma_e)$ – which represents the goodness-of-fit metric and therefore the objective function – the BMS keeps the most plausible one, representing the expression with the shortest description length (the best goodness-of-fit).

3.3. Model performance metrics

For assessing the performance of the models, an arbitrary set of initial conditions can be used to integrate the ODE, comparing the simulated and experimental profiles values. After training the models on dedicated training runs, separated test runs were used to assess their performance. A detailed description of how the data is generated and split into training and test sets is shown in Section 4.

The performance is assessed via the root mean squared error ($RMSE$) and the coefficient of determination (R^2), defined as given in Eqs. (14) and (15). These metrics can be calculated for both the training and test sets, obtaining the training and test errors, respectively. They can be calculated for the concentration (state) space or the derivative space.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i\in I}\sum_{u\in U}\left(\widehat{X}_{i,u} - X_{i,u}\right)^2} \tag{14}$$

$$R^2 = 1 - \frac{SSR}{SST} = 1 - \frac{\sum_{i\in I}\sum_{u\in U}\left(\widehat{X}_{i,u} - X_{i,u}\right)^2}{\sum_{i\in I}\sum_{u\in U}\left(X_{i,u} - \overline{X}_{i,u}\right)^2} \tag{15}$$

In these relationships, the predictions by the model are described by $\widehat{X}_{i,u}$. The experimental data points and the mean of the experimental data points of the data are described by $X_{i,u}$ and $\overline{X}_{i,u}$, respectively. Variables SSR and SST denote the sum of squares of residuals and the total sum of squares (proportional to the variance of the data), respectively. The error metrics in (14) and (15) can be calculated for the state and derivative variables.

3.4. Implementation details

All calculations were carried out on an Intel®Core™ i7-8700 CPU and 16 GB of RAM. To construct the sampling dataset, we used Python 3.10 with NumPy v1.24.3 and pyDOE v0.3.8. For the BMS training, the hyperparameter values are those given in the original article of the BMS (Guimerà et al., 2020), i.e., 5 % probability of root replacement, 45 % probability of node replacement, and 50 % probability of elementary tree replacement. The allowed unitary operations included $\exp(x)$, $\log(x), x^2, x^3$, and $\sqrt{x}$, while the binary operations consisted of $+, -,$ Ã·, $\times$, and x^a. The maximum number of MCMC steps was chosen to be 10^4. The neural network training was performed with Scikit-learn v1.0.2 (Pedregosa et al., 2011). A grid search with a 3-fold cross-validation was performed to tune and find appropriate hyperparameters for this benchmark model. Parameters considered during the grid search were the hidden layer size, the activation function, the learning rate, and the initial learning rate. Details of this grid search and the settings of the fixed hyperparameters are given in Section S3 of the supporting information.

4. Case studies

Subsequently, two different case studies are presented. We employed Latin hypercube sampling (LHS) together with the bounds given in Table 1 to generate different initial conditions, with each set of initial conditions representing a different batch. For each case study shown below, 13 batch runs were simulated in total. From those, 10 batches were used to train the models and 3 were taken as test batches. We added normally distributed noise (NumPy) with a mean of $\mu = 0$ and a variance of $\sigma^2 = 0.2$ to the profiles obtained from integrating the different batches to create more realistic data (more significant noise level in lower numerical ranges to resemble measurement errors). For the two case studies, several scenarios were considered, which are summarized in Fig. 4. To study the influence of the amount of data available, we generated profiles with 40 and 20 time points per batch. It is worth to be mentioned that the time spans of the subsequently introduced case studies are 80 h and 180 h, respectively. A sampling rate of 20 points within this time frame results in one sample every two hours and every approximately 9 h. Indeed, it should be kept in mind that a reduction of the sampling frequency will result in a reduction in accuracy of the derivative approximation, which is discussed in more detail

Table 1
Lower and upper bounds used for generating the training and test sets. With those bounds and a Latin Hypercube Sampling approach, different initial conditions were generated. These were used to solve the systems of ODEs in (16) and (17) to create different batch runs.

	CSI			CSII			
Species	B	S	P	B	C	N	P
Lower bound	0.1	50	0	216	108	450	17
Upper bound	0.4	90	0	264	132	550	21
Unit	g/L	g/L	g/L	mg/L	mg/L	mg/L	mg/L

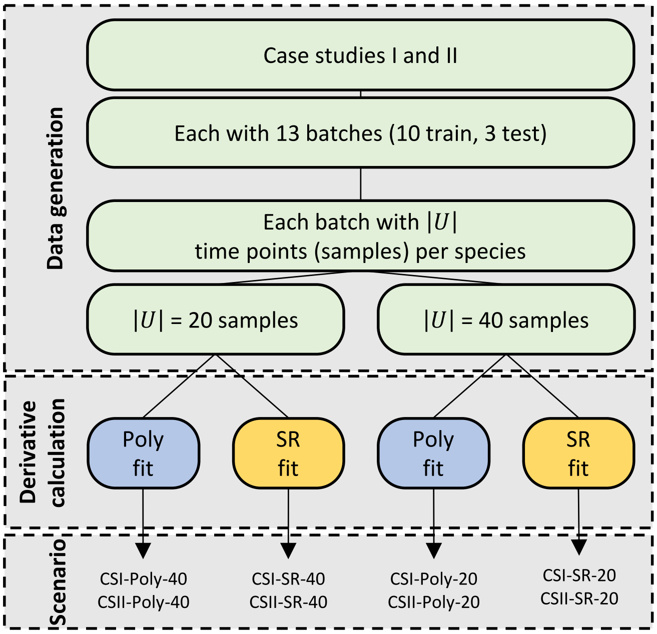

Fig. 4. An overview of the organization of the case studies is shown schematically. For each of the base case studies discussed below, batches with either 20 or 40 samples were generated. Then, either the polynomial or symbolic regression approach was applied to calculate the derivatives.

in the supporting information section S1. To calculate the derivatives from the data, the polynomial fit or symbolic regression fit, both described in (5), were used. Hence, four different scenarios for each case study were explored. The resulting scenarios are described by the abbreviations Poly-20, Poly-40, SR-20, and SR-40, depending on the number of points per batch and the method for derivative approximation (Fig. 4). As an example, CSI-Poly-40 describes the scenario of CSI with the polynomial approach for the derivative calculation and 40 samples per batch and species. The case studies are also collected and published on GitHub (https://github.com/forstertim/insidapy).

4.1. Case study I

A bioprocess is considered where some bacteria produce a specific product while consuming a substrate. The variables B, S, and P (all in g/L) represent the biomass, substrate, and product concentration, respectively. These species are summarized in the vector $X = \{B, S, P\}$. The process is modeled in batch mode and adapted from Turton et al. (2018):

$$\frac{dB}{dt} = \phi \cdot B$$

$$\frac{dS}{dt} = -\frac{1}{Y_{B,S}} \cdot \phi \cdot B$$

$$\frac{dP}{dt} = \frac{Y_{P,S}}{Y_{B,S}} \cdot \phi \cdot B$$

$$\phi = \phi_{max} \cdot \frac{S}{K_S + S} \cdot \frac{\Omega(A_1, T, E_{A,1})}{1 + \Omega(A_2, T, E_{A,2})} \cdot \left(1 - \frac{B}{K_\phi + B}\right) \tag{16}$$

In these mass balances, ϕ (1/h) models the growth rate. $Y_{i,j}$ represents the yield coefficient of species j with respect to species i. The expressions $\Omega(A, T, E_A)$ represent Arrhenius reaction rates that depend on the temperature T and temperature-independent pre-factors and activation energies $E_{A,1}$, $E_{A,2}$, A_1, and A_2, respectively. The parameters K_s and K_ϕ represent half-saturation constants. Data was generated for the interval $t = [0, 80]$ hours. The values of the parameters are given in Table S3 of the supporting information. As mentioned in Section 3.1, the first two and last five points (polynomial approach) or the first two and last two points (BMS approach) were excluded for model training.

4.2. Case study II

Here, we focus on a bioprocess studied by Del Rio-Chanona et al. (2019). The system of ODEs in (17) is based on a Monod model, a Logistic model, and a Luedeking-Piret model (Zhang et al., 2015), where cell growth, cell decay, and substrate uptakes are considered. For a detailed description, the reader is referred to the work of Del Rio-Chanona et al. (2019).

$$\frac{dB}{dt} = \mu \frac{N}{N + K_N} \frac{C}{C + K_C} \frac{P}{P + K_P} B - \mu_d B^2$$

$$\frac{dC}{dt} = -Y_{C1}\left(\mu \frac{N}{N + K_N} \frac{C}{C + K_C} \frac{P}{P + K_P} B - \mu_d B^2\right) - Y_{C2} B$$

$$\frac{dN}{dt} = -Y_{N1}\left(\mu \frac{N}{N + K_N} \frac{C}{C + K_C} \frac{P}{P + K_P} B - \mu_d B^2\right) - Y_{N2} B$$

$$\frac{dP}{dt} = -Y_{P1}\left(\mu \frac{N}{N + K_N} \frac{C}{C + K_C} \frac{P}{P + K_P} B - \mu_d B^2\right) - Y_{P2} B \tag{17}$$

In this system, the variables B, C, N, and P represent the biomass, carbon, nitrogen, and phosphate concentrations, respectively (all in mg/L). These species are summarized in the vector $X = \{B, C, N, P\}$. The parameters K_N, K_C, and K_P represent the half-velocity coefficient of the corresponding substrates, where the parameters Y_{i1} and Y_{i2} are growth-dependent and growth-independent yield coefficients of the species $i = \{C, N, P\}$. The biomass growth and death are given by μ and μ_d. The concentration of the biomass is divided by 1000 so that the originally reported parameter values can be used (Zhang et al., 2015). The time window investigated corresponds to $t = [0, 180]$ hours. The values of the parameters are given in Table S4 of the supporting information. As in CSI, the first two and last five points (polynomial approach) or the first two and last two points (BMS approach) were excluded for model training.

5. Results and discussion

Below, the results of the BMS are compared to the ones obtained with the ANN. A summary of the obtained coefficients of determination (R^2) for the model training and testing is given in Table 2. The performance metrics are displayed for the different scenarios (as visualized in Fig. 4), while results are also depicted in Fig. 6 for CSI and Fig. 7 for CSII. These plots show the calculated derivative values against the model predictions for both modelling approaches (BMS and ANN). Additional results are given in Section S5 of the supporting information.

In general, both models achieve similar performance in both the derivative and the state (concentration) space, with our approach often outperforming the ANN, but not by much, as discussed next. Recall that the models should not only train well in the derivative space but also after integration since we are interested in predicting concentration profiles. Therefore, we focus first on the model with best performance in the state space of the unseen test batches (in Table 2, the highest R^2 value of the test set is highlighted in bold). The best-performing models are identified by the BMS in most scenarios, although the differences with the ANN are small. The only exception where the ANN outperforms the BMS is in CSI-SR-40, although also there, differences are marginal, as also seen in Fig. 5 displaying the state space predictions for this scenario. In addition to the data given in Table 2, Fig. 5 shows one of the test batches results for the models identified in scenarios CSI-Poly-40 (top row) and CSI-Poly-20 (bottom row). As shown in this figure, the models are well able to predict the evolution of the concentration, even if a

Table 2
The coefficients of determination (R^2, unitless) are shown for the training and testing runs (notation: train/test) for the two case studies and their respective scenarios. For each case scenario, the best-performing approach in terms of state-space performance is indicated in bold text. The CPU times for the model training are indicated as mean values of the times for training models of the different species. In the ANN case, the time for the grid search is included. The raw values of the CPU times are indicated in Section S5 of the supporting information.

CS	Derivative Method	CPU model training [s]		BMS State R^2	BMS Derivative R^2	ANN State R^2	ANN Derivative R^2
		BMS	ANN				
I	Poly-40	7271	88	0.961 / **0.999**	0.995 / 0.995	0.837 / 0.986	0.992 / 0.994
	SR-40	4132	88	0.977 / 0.900	0.979 / 0.990	0.959 / **0.990**	0.982 / 0.988
	Poly-20	5803	59	0.998 / **0.995**	0.997 / 0.996	0.994 / 0.994	0.996 / 0.996
	SR-20	8167	57	0.993 / **0.989**	0.981 / 0.986	0.988 / 0.984	0.983 / 0.988
II	Poly-40	9384	130	1.000 / **1.000**	0.995 / 0.995	0.996 / 0.998	0.958 / 0.989
	SR-40	9275	162	1.000 / **1.000**	0.995 / 0.997	0.997 / 0.999	0.973 / 0.993
	Poly-20	2120	155	1.000 / **1.000**	0.986 / 0.986	0.996 / 0.996	0.885 / 0.901
	SR-20	4492	151	1.000 / **1.000**	0.997 / 0.994	0.999 / 0.999	0.971 / 0.978

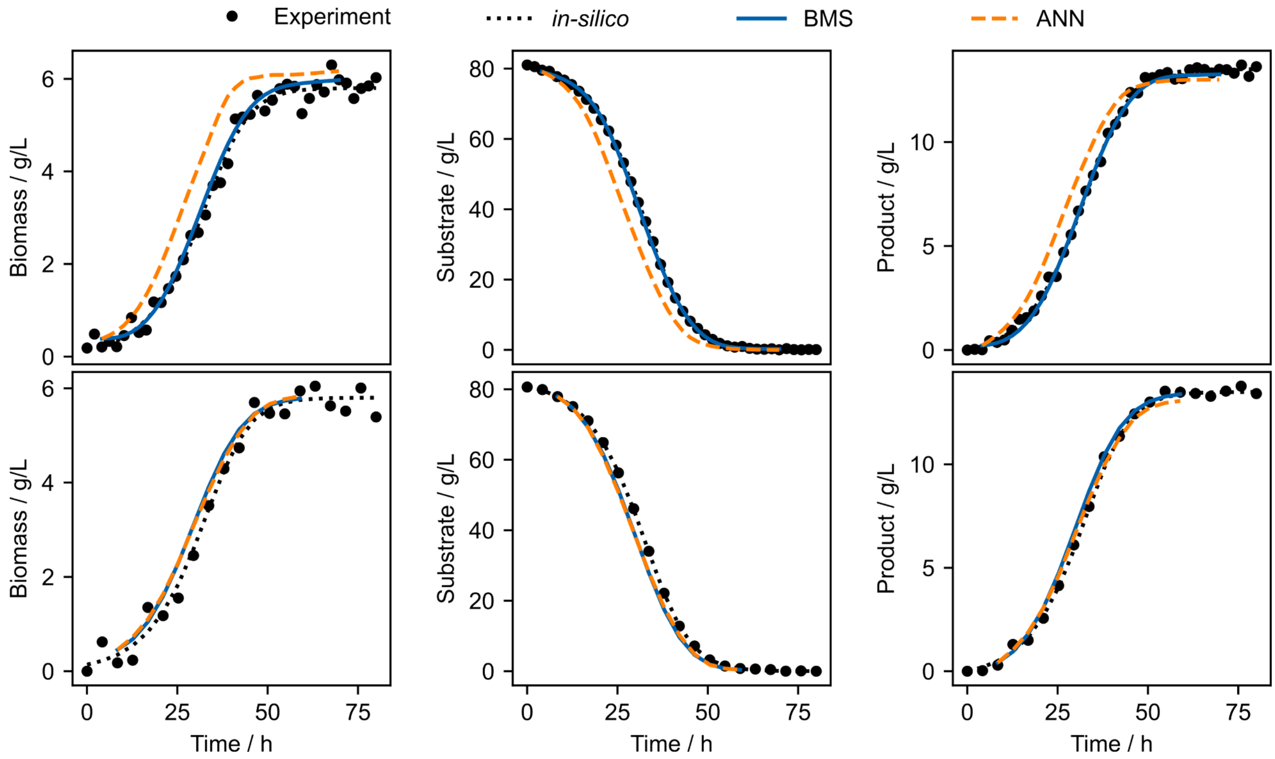

Fig. 5. The concentration profiles of the three species in CSI are shown together with the model predictions. The black circles represent the observed noisy data. The dashed orange line represents the ANN predictions, whereas the blue solid line represents the BMS predictions. It is worth mentioning the model predictions are only shown for the experimental time points that were used for model training, since some initial and last samples were removed from the training, as discussed in Section 3.1. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

lower sampling frequency was used (20 vs. 40 samples). From the results shown in Table 2, having fewer data points per batch does not significantly impact the performance of the models. Also, there was no clear difference in performance when comparing the two differentiation approaches.

Considering the reaction rates space, both models lead to very similar performance in all scenarios. Interestingly, although trained only in the derivative space, both models can predict well after integration. This would support the assumption that the rate expressions can be well approximated by both models. At this point, it is worth mentioning that – although it was never observed during the calculations of the present work – the challenge of stiff ODEs might be encountered. Since the model is identified in the derivative space and then incorporated into an ODE that is subsequently integrated, the occurrence of such stiff ODE systems might not be fully circumventable by the presented approach. However, the analysis of stiffness and stability was out of scope of this present work and might be the focus of a future study.

Although both models seem to perform similarly throughout the case studies, there is one significant advantage of using BMS. After identification of the rate expression, the model is provided in analytical form and can be, arguably, interpreted more easily than purely data-driven models. For CSI (CSI-Poly-40), the most plausible expressions obtained by the BMS for the ODE system are given in (18) as an example. Additionally, the corresponding estimated values of the parameters in (18) are given in Table 3. The identified BMS models with the corresponding estimated model parameters for the other scenarios are summarized in Section S6 of the supporting information.

$$\frac{dB}{dt} = a_0\left((S\cdot B)^{a_1+\left(\frac{a_2}{a_2+P}\right)}\right) \tag{18}$$

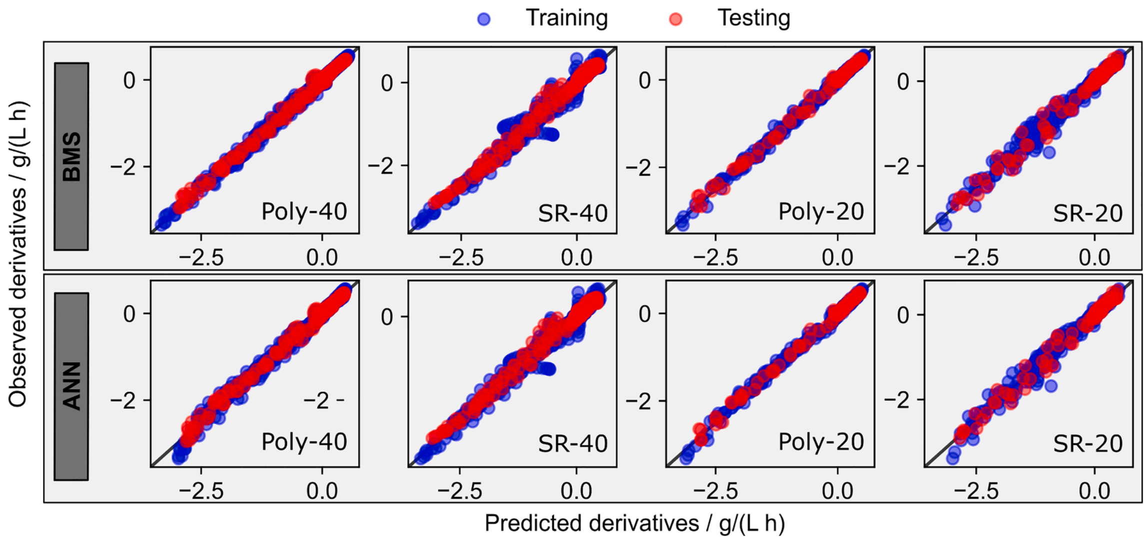

Fig. 6. The observed values are plotted against the model prediction values for CSI. The columns represent the different scenarios of the case study. The top row shows the results obtained from the BMS predictions, whereas the bottom row shows the results from the neural network. Blue points represent the training data, whereas red points correspond to the test data. The black line represents the values where the observed value corresponds to the model predictions. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

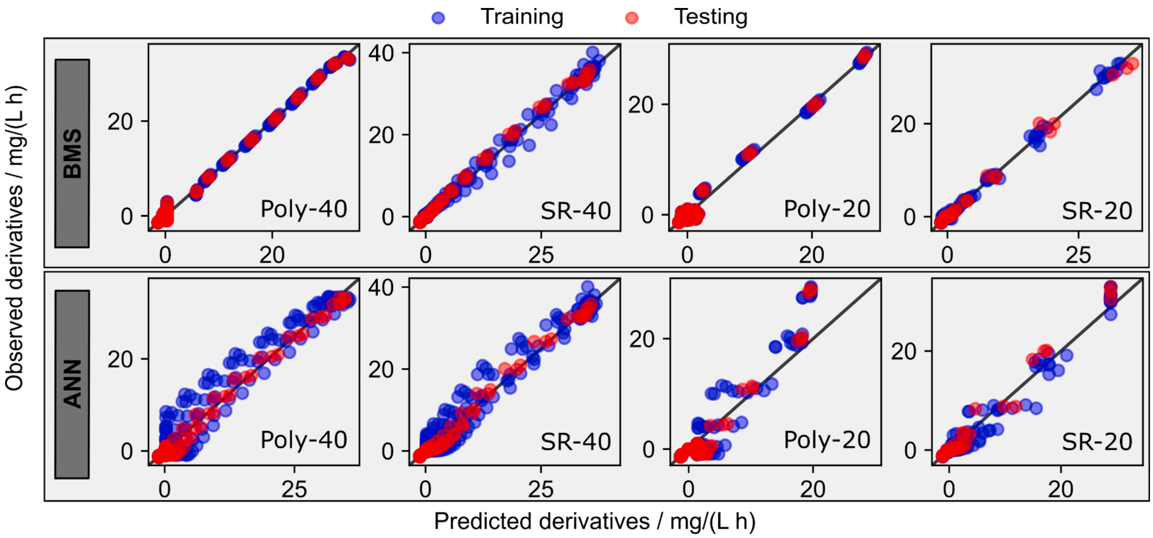

Fig. 7. The observed values are plotted against the model prediction values for CSII. The columns represent the different scenarios of the case. The top row shows the results obtained from the BMS predictions, whereas the bottom row shows the results from the neural network. Blue points represent the training data, whereas red points correspond to the test data. The black line represents the values where the observed value corresponds to the model predictions. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 3
Parameter values of the most plausible algebraic models identified by the Bayesian machine scientist (BMS) for CSI-Poly-40.

Parameter	Rate equation for CSI		
	dB/dt	dS/dt	dP/dt
a_0	$4.164{\cdot}10^{-3}$	$5.343{\cdot}10^{-2}$	$-1.797{\cdot}10^{0}$
a_1	$8.285{\cdot}10^{-1}$	$2.072{\cdot}10^{-2}$	$4.059{\cdot}10^{-1}$
a_2	$-1.086{\cdot}10^{-1}$	$2.302{\cdot}10^{-2}$	$4.175{\cdot}10^{-3}$

$$\frac{dS}{dt} = a_2 + a_0 - a_1 \left(P + B^{S\cdot a_1}\right)\cdot S^{\left(\frac{1}{S\cdot P}\right)^{a_0}} \tag{19}$$

$$\frac{dP}{dt} = -\left(\frac{a_2 (B\cdot P)^{a_1}}{S^{a_0}}\right)\left(\left(\frac{a_2}{P}\right) + \frac{a_0}{S + (a_1^S)}\right) \tag{20}$$

After the model training and the deployment for predicting the time dependency of the concentration profiles, one can analyse the obtained ODEs to gather some qualitative knowledge from those closed-form expressions. This will be shown with the example of the biomass growth and the substrate consumption. It is worth to be mentioned that this analysis is done on a conceptual and qualitative level to extract some knowledge and trends about the underlying system.

Considering the growth of the biomass B in Eq. (18), all three species – B, S, and P – seem to influence the change in biomass concentration.

These findings can be interpreted using the underlying ground truth model in (16), which was used to generate the noisy data. In this underlying ground truth model the product concentration P is not involved in the rate equation of the biomass. Nevertheless, the BMS equation takes also P into account in (18). However, taking a closer look at the exponent in this equation, namely $a_1 + (a_2/(a_2+P))$, one can observe Monod-type similarities with an asymptotic behaviour. The value of this entire exponent converges towards a given value a_2, which is displayed in Fig. 8 (a).

Although the BMS considers the product in the identified model expression for dB/dt, the effect of a change in P is more significant in the beginning and becomes less important throughout the reaction (when the product is formed, and its concentration increases). In other words, the main influences on dB/dt result from the part $a_0(S{\cdot}B)$, for most of the reaction time, since the exponent has more or less a similar value around ≈ 0.8 (Fig. 8 (a)) during most of the time, which is in-line with the underlying ground truth equation in (16) (no impact of P). Fig. 8 (b) displays the true change of biomass ($\phi{\cdot}B$) as a function of the substrate and biomass concentrations. Considering two specific values of the biomass (i.e., 0.02 or 0.30 g/L), the growth can be shown as a function of the substrate. In case of low biomass availability (blue dashed line), the growth seems to be less dependent on the substrate, whereas in case more biomass is available (black dotted-dashed line), the substrate concentration shows a greater impact on the growth. In such a case, as expected, as soon as the substrate level drops, a significant decrease in growth rate can be observed (right part of Fig. 8 (b) for the black dotted-dashed line). The predicted time series profiles by the BMS given in Fig. 5 show a good accuracy also in the beginning and at the end of the process operation, for which the mentioned significant drop in the growth needs to be captured. The BMS was able to describe such trends without the need of chemical or biological background knowledge. If the growth predicted by the BMS – the right-hand side of equation (18) – is visualized (Fig. 9), a similar trend can be observed, although slight numerical discrepancies are observable compared to the underlying system in Fig. 8 (b).

A similar analysis can be performed for example for the identified equation of the substrate consumption rate, given in (19). The BMS identified an expression where all state variables show an inhibiting influence on the rate of S. In other words, the consumption of the substrate is enhanced by increasing the concentration of the other species in the system. Due to the closed-form availability of the model, a deeper analysis of the rate equation is possible, which is showcased by a further decomposition of the identified expression into individual terms, namely h_1, h_2, and h_3 given below. Compared to the pure ANN, this poses an advantage since knowledge about a system can be extracted.

$$h_1 = a_2 + a_0 \tag{21}$$

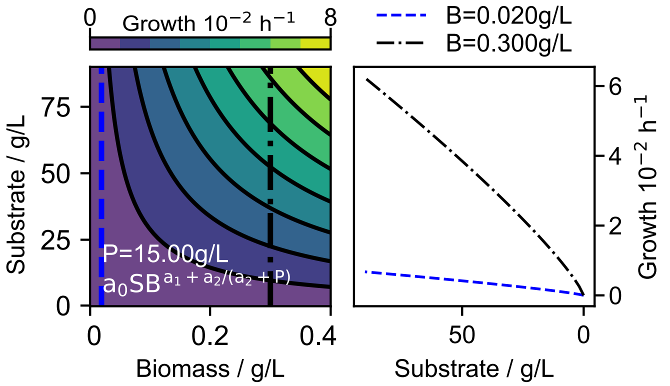

Fig. 9. The biomass growth identified by the BMS equation is visualized as a function of the substrate concentration S and the biomass concentration B (for the indicated concentration of the product). Similarly to Fig. 8, two scenarios are highlighted by the blue dashed (constant biomass of 0.02 g/L) and black dotted-dashed lines (constant biomass of 0.30 g/L), for which the growth is shown as a univariate function of the substrate concentration. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$h_2 = a_1{\cdot}P{\cdot}S^{\left(\frac{1}{S\cdot P}\right)^{a_0}} \tag{22}$$

$$h_3 = a_1{\cdot}B^{S\cdot a_1}{\cdot}S^{\left(\frac{1}{S\cdot P}\right)^{a_0}} \tag{23}$$

Since h_1 only consists of constants, this term is disregarded for the time being, as no metabolite influences it. Considering the terms h_2 and h_3, it is observable that the constant a_1 and the part $S^{\left({}^{1}/_{S\cdot P}\right)^{a_0}}$ is the same for both terms. In case one is interested in the significance of the individual parts, the numerical ratio of the two terms will matter, since both terms, h_2 and h_3 have the same sign and therefore the same impact on the consumption of the substrate. Creating such a ratio $\psi = h_2/h_3 = P/B^{Sa_1}$ will result in the following consumption rate of the substrate (still disregarding h_1):

$$\frac{dS}{dt} \approx -a_1{\cdot}P{\cdot}S^{\left(\frac{1}{S\cdot P}\right)^{a_0}} - a_1{\cdot}B^{S\cdot a_1}{\cdot}S^{\left(\frac{1}{S\cdot P}\right)^{a_0}} \approx \underbrace{-a_1{\cdot}P{\cdot}S^{\left(\frac{1}{S\cdot P}\right)^{a_0}}}_{h_2}\underbrace{-a_1{\cdot}\frac{P}{\psi}{\cdot}S^{\left(\frac{1}{S\cdot P}\right)^{a_0}}}_{h_3} \tag{24}$$

With this, one can observe that if $\psi > 1$, it results in a case where

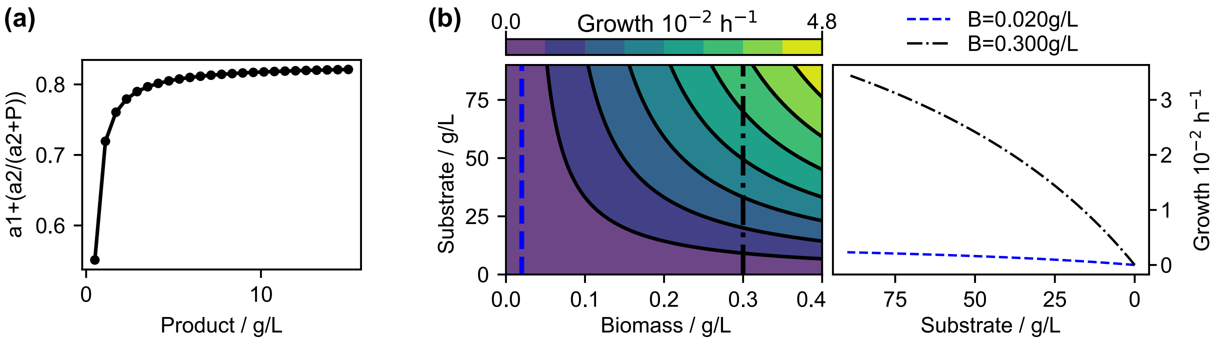

Fig. 8. (a) The exponent in equation (18) is shown as a function of the product concentration P. In (b), the biomass growth $\phi{\cdot}B$ given by the underlying system in equation (16) is visualized as a function of the substrate concentration S and the biomass concentration B. Additionally, in (b), two scenarios are highlighted by the blue dashed (constant biomass of 0.02 g/L) and black dotted-dashed lines (constant biomass of 0.30 g/L), for which the growth is shown as a univariate function of the substrate concentration. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

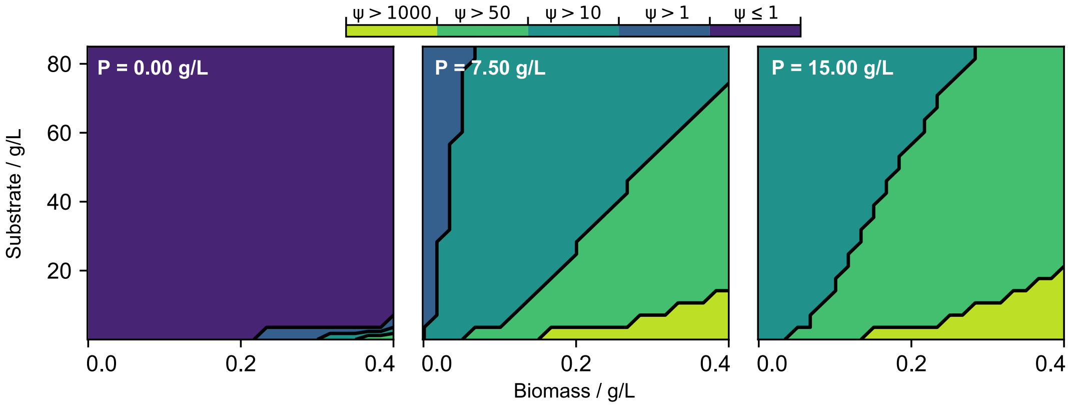

Fig. 10. The contour plots of the numerical ratio ψ are shown for three different levels of the product concentration P, which are [0, 7.5, 15] g/L. The colours of the contour represent the value of ψ.

$h_3 < h_2$. On the other hand, if $\psi < 1$, the case $h_3 > h_2$ is obtained. Visualizing the value of ψ for different ranges of the biomass and substrate concentration and a given value of the product concentration (P) in Fig. 10, one can observe how the terms change their numerical relevance compared to each other (which term has more impact on the substrate concentration). With growing product concentration P, the value of ψ starts to grow as well ($\psi \gg 1$), leading to higher contributes by the term h_2.

The obtained closed-form expressions models bring not only the advantage of being able to extract some knowledge on the system's behaviour. Due to the algebraic form of the models, another useful benefit is the possibility to calculate the gradients analytically. This opens the opportunity to include these models for example in deterministic optimization algorithms, where the objective functions and constraints need to be available in closed-form manner (Bongartz and Mitsos, 2019; Misener and Floudas, 2014; Smith and Pantelides, 1999; Tawarmalani and Sahinidis, 2002).

Despite the above-discussed advantages the closed-form analytical equations provide, there are also disadvantages, where the high CPU times for the BMS model training is one of the main drawbacks. Considering the averaged CPU times for the BMS training in Table 2, the models required at least 35 min (CSII-Poly-20) and at most 156 min (CSII-Poly-40). The exact CPU times are documented in Tables S7–S9 in Section S5 of the supporting information. As discussed in earlier works (Forster et al., 2023b; Negri et al., 2022; Vázquez et al., 2022), the BMS in general requires significantly more training time than the benchmark surrogates (i.e., ANN and GP). This is because the latter are based on a fixed canonical formalism and highly efficient algorithms, such as those implemented in the used Python packages Scikit-learn (Pedregosa et al., 2011). Also, the BMS algorithm was originally designed by the authors to only allow the number of MCMC steps as a stopping criterion (Guimerà et al., 2020), which limits the ability of the algorithm to find models with better description length. Regarding this, the evolution of the description length, given in expression (13), is shown in Fig. 11 as a function of the number of executed MCMC steps.

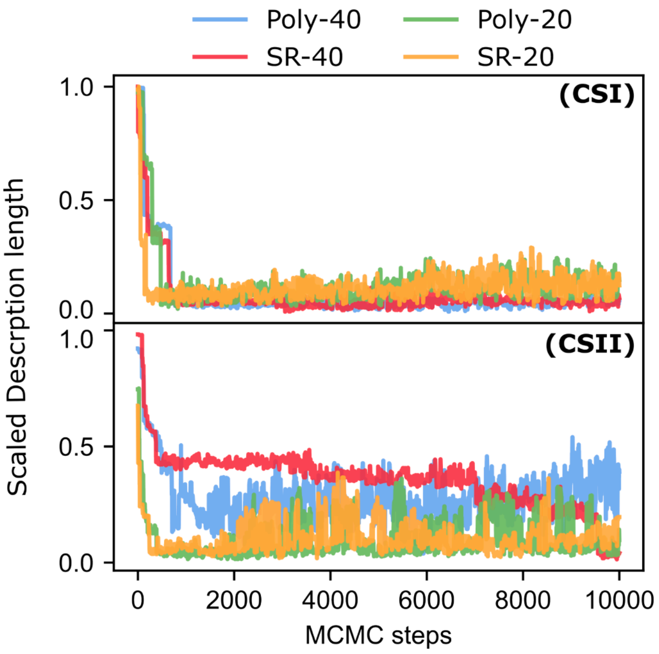

Fig. 11. The scaled mean description lengths are visualized for each MCMC step for CSI (top) and CSII (bottom). The mean results from averaging the description lengths of the different BMS models obtained for each species of each scenario (Poly-40, SR-40, Poly-20, SR-20).

To compare the case studies, the description lengths were scaled to a range between zero and one. For CSI, it can be observed in Fig. 11 (top) that after around 800 MCMC steps, the description length does not significantly change. A similar picture is observed in Fig. 11 (bottom) for CSII, where the most significant decline in the description length was achieved in the first 2000 MCMC steps. The only exception can be observed in the scenario CSII-SR-40, where the description length declines gradually. These observations imply that the models identified after those steps perform similarly in terms of training predictions.

6. Conclusion

In this work, we investigated the use of machine learning to identify kinetic models of bioprocesses without assuming a pre-defined model structure. A symbolic regression algorithm, the Bayesian machine scientist, was employed to generate suitable models considering their error and level of similarity with a predefined corpus of equations. The model training was performed following a two-step approach, thus avoiding the iterative integration of differential equations, by using two methods to calculate derivatives, i.e., polynomial fitting and univariate symbolic regression. Also, the influence of the sample size was studied.

Our approach was applied to two different case studies to showcase its capabilities. Our method performed slightly better than ANNs, while leading to analytical expressions that can be more easily analysed. However, the BMS leads to higher computational times, which might be reduced in the future as symbolic regression algorithms reach higher maturity levels. Future work should focus on guiding the SR algorithm more efficiently towards equations that are more likely to explain the data precisely, for example by using tailored standard kinetic equations during the training of the SR algorithm.

CRediT authorship contribution statement

Tim Forster: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Daniel Vázquez:** Writing – review & editing, Methodology, Conceptualization. **Claudio Müller:** Methodology, Conceptualization. **Gonzalo Guillén-Gosálbez:** Writing – review & editing, Supervision, Project administration, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Codes for the data generation and preparation will be published.

Acknowledgements

The authors would like to acknowledge the financial support from the Swiss National Science Foundation (Project LEARN-D, MINT 200021_214877).

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.ces.2024.120606.

References

Bishop, C.M., 2006. Pattern Recognition and Machine Learning, EAI/Springer Innovations in Communication and Computing.

Bongartz, D., Mitsos, A., 2019. Deterministic global flowsheet optimization: between equation-oriented and sequential-modular methods. AIChE J. 65, 1022–1034. https://doi.org/10.1002/aic.16507.

Brendel, M., Bonvin, D., Marquardt, W., 2006. Incremental identification of kinetic models for homogeneous reaction systems. Chem. Eng. Sci. 61, 5404–5420. https://doi.org/10.1016/j.ces.2006.04.028.

Brunton, S.L., Proctor, J.L., Kutz, J.N., Bialek, W., 2016. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc. Natl. Acad. Sci. U.S.A., 113, 3932–3937. https://doi.org/10.1073/pnas.1517384113.

Costa, L., Oliveira, P., 2001. Evolutionary algorithms approach to the solution of mixed integer non-linear programming problems. Comput. Chem. Eng. 25, 257–266. https://doi.org/10.1016/S0098-1354(00)00653-0.

Cozad, A., Sahinidis, N.V., 2018. A global MINLP approach to symbolic regression. Math. Program. 170, 97–119. https://doi.org/10.1007/s10107-018-1289-x.

Cranmer, M., Sanchez-Gonzalez, A., Battaglia, P., Xu, R., Cranmer, K., Spergel, D., Ho, S., 2020. Discovering Symbolic Models from Deep Learning with Inductive Biases. Advances in Neural Information Processing Systems 2020-Decem, 1–14.

Cranmer, M., 2020. PySR: Fast And Parallelized Symbolic Regression in Python/Julia. https://doi.org/10.5281/zenodo.4041459.

DataModeler, 2023. DataModeler.

Servia, M.Á. de C., Sandoval, I.O., Hellgardt, K., Kuok, K., Hii, Zhang, D., Chanona, E.A. del R., 2023. The Automated Discovery of Kinetic Rate Models – Methodological Frameworks.

Del Rio-Chanona, E.A., Cong, X., Bradford, E., Zhang, D., Jing, K., 2019. Review of advanced physical and data-driven models for dynamic bioprocess simulation: case study of algae–bacteria consortium wastewater treatment. Biotechnol. Bioeng. 116, 342–353. https://doi.org/10.1002/bit.26881.

Diveev, A., Shmalko, E., 2021. Machine Learning Control by Symbolic Regression. Machine Learning Control by Symbolic Regression. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-030-83213-1.

Ferreira, J., Pedemonte, M., Torres, A.I., 2019a. A Genetic Programming Approach for Construction of Surrogate Models, in: Computer Aided Chemical Engineering. Elsevier, pp. 451–456. https://doi.org/10.1016/B978-0-12-818597-1.50072-2.

Ferreira, J., Torres, A.I., Pedemonte, M., 2019b. A Comparative Study on the Numerical Performance of Kaizen Programming and Genetic Programming for Symbolic Regression Problems, in: 2019 IEEE Latin American Conference on Computational Intelligence (LA-CCI). pp. 1–6. https://doi.org/10.1109/LA-CCI47412.2019.9036755.

Forster, T., Vázquez, D., Guillén-Gosálbez, G., 2023c. Global optimization of symbolic surrogate process models based on Bayesian learning, in: Kokossis, A.C., Georgiadis, M.C., Pistikopoulos, E. (Eds.), Computer Aided Chemical Engineering, 33 European Symposium on Computer Aided Process Engineering. Elsevier, pp. 1241–1246. https://doi.org/10.1016/B978-0-443-15274-0.50198-0.

Forster, T., Vázquez, D., Cruz-Bournazou, M.N., Butté, A., Guillén-Gosálbez, G., 2023a. Modeling of bioprocesses via MINLP-based symbolic regression of S-system formalisms. Comput. Chem. Eng. 170, 108108 https://doi.org/10.1016/j.compchemeng.2022.108108.

Forster, T., Vázquez, D., Guillén-Gosálbez, G., 2023b. Algebraic surrogate-based process optimization using Bayesian symbolic learning. AIChE Journal, e18110. https://doi.org/10.1002/aic.18110.

Gnoth, S., Jenzsch, M., Simutis, R., Lübbert, A., 2007. Product formation kinetics in a recombinant protein production process. IFAC Proc. 40, 201–206. https://doi.org/10.3182/20070604-3-MX-2914.00035.

Gnoth, S., Jenzsch, M., Simutis, R., Lübbert, A., 2008. Control of cultivation processes for recombinant protein production: a review. Bioprocess Biosyst. Eng. 31, 21–39. https://doi.org/10.1007/s00449-007-0163-7.

Gnoth, S., Simutis, R., Lübbert, A., 2010. Selective expression of the soluble product fraction in Escherichia coli cultures employed in recombinant protein production processes. Appl. Microbiol. Biotechnol. 87, 2047–2058. https://doi.org/10.1007/s00253-010-2608-1.

Grünwald, P.D., 2007. The Minimum Description Length Principle. The MIT Press.

Guillén-Gosálbez, G., Miró, A., Alves, R., Sorribas, A., Jiménez, L., 2013. Identification of regulatory structure and kinetic parameters of biochemical networks via mixed-integer dynamic optimization. BMC Syst. Biol. 7, 113. https://doi.org/10.1186/1752-0509-7-113.

Guimerà, R., Reichardt, I., Aguilar-Mogas, A., Massucci, F.A., Miranda, M., Pallarès, J., Sales-Pardo, M., 2020. A Bayesian machine scientist to aid in the solution of challenging scientific problems. Sci. Adv. 6 https://doi.org/10.1126/sciadv.aav6971.

Hansen, M.H., Yu, B., 2001. Model selection and the principle of minimum description length. J. Am. Stat. Assoc. 96, 746–774. https://doi.org/10.1198/016214501753168398.

Hastings, W.K., 1970. Monte Carlo sampling methods using Markov chains and their applications. Biometrika 57, 97. https://doi.org/10.2307/2334940.

Kahrs, O., Marquardt, W., 2007. The validity domain of hybrid models and its application in process optimization. Chem. Eng. Process. 46, 1054–1066. https://doi.org/10.1016/j.cep.2007.02.031.

Keane, M.A., Koza, J.R., Rice, J.P., 1993. Finding an Impulse Response Function Using Genetic Programming, in: 1993 American Control Conference. IEEE, pp. 2345–2350. https://doi.org/10.23919/ACC.1993.4793307.

Koza, J.R., 1994. Genetic programming as a means for programming computers by natural selection. Stat. Comput. 4, 87–112. https://doi.org/10.1007/BF00175355.

Kroll, P., Hofer, A., Stelzer, I.V., Herwig, C., 2017. Workflow to set up substantial target-oriented mechanistic process models in bioprocess engineering. Process Biochem. 62, 24–36. https://doi.org/10.1016/j.procbio.2017.07.017.

McKay, B., Willis, M., Searson, D., Montague, G., 1999. Non-Linear Continuum Regression Using Genetic Programming. Proc. of the Genetic and Evolutionary Computation Conf. (GECCO)-99 2, 1106–1111.

McKay, B., Willis, M., Barton, G., 1997. Steady-state modelling of chemical process systems using genetic programming. Comput. Chem. Eng. 21, 981–996. https://doi.org/10.1016/S0098-1354(96)00329-8.

Mercier, S.M., Diepenbroek, B., Wijffels, R.H., Streefland, M., 2014. Multivariate PAT solutions for biopharmaceutical cultivation: current progress and limitations. Trends Biotechnol. 32, 329–336. https://doi.org/10.1016/j.tibtech.2014.03.008.

Michalik, C., Chachuat, B., Marquardt, W., 2009. Incremental global parameter estimation in dynamical systems. Ind. Eng. Chem. Res. 48, 5489–5497. https://doi.org/10.1021/ie8015472.

Miró, A., 2014. Dynamic mathematical tools for the identification of regulatory structures and kinetic parameters in systems biology (Doctoral dissertation). Rovira I Virgili University, Tarragona, Spain https://www.tdx.cat/bitstream/handle/10803/284043/Tesi%20Antoni%20miro%20roig.pdf?sequence=1.

Misener, R., Floudas, C.A., 2014. ANTIGONE: algorithms for coNTinuous / integer global optimization of nonlinear equations. J. Glob. Optim. 59, 503–526. https://doi.org/10.1007/s10898-014-0166-2.

Mowbray, M.R., Wu, C., Rogers, A.W., Rio-Chanona, E.A.D., Zhang, D., 2023. A reinforcement learning-based hybrid modeling framework for bioprocess kinetics identification. Biotechnol. Bioeng. 120, 154–168. https://doi.org/10.1002/bit.28262.

Murphy, K.P., 2013. Machine Learning: A Probabilistic Perspective. The MIT Press.

Narayanan, H., Luna, M.F., von Stosch, M., Cruz Bournazou, M.N., Polotti, G., Morbidelli, M., Butté, A., Sokolov, M., 2020. Bioprocessing in the digital age: the role of process models. Biotechnol. J. 15, 1–10. https://doi.org/10.1002/biot.201900172.

Narayanan, H., Seidler, T., Luna, M.F., Sokolov, M., Morbidelli, M., Butté, A., 2021. Hybrid Models for the simulation and prediction of chromatographic processes for protein capture. J. Chromatogr. A 1650, 462248. https://doi.org/10.1016/j.chroma.2021.462248.

Narayanan, H., von Stosch, M., Feidl, F., Sokolov, M., Morbidelli, M., Butté, A., 2023. Hybrid modeling for biopharmaceutical processes: advantages, opportunities, and implementation. Front. Chem. Eng. 5.

Negri, V., Vázquez, D., Sales-Pardo, M., Guimerà, R., Guillén-Gosálbez, G., 2022. Bayesian symbolic learning to build analytical correlations from rigorous process simulations: application to CO_2 capture technologies. ACS Omega 7, 41147–41164. https://doi.org/10.1021/acsomega.2c04736.

Neumann, P., Cao, L., Russo, D., Vassiliadis, V.S., Lapkin, A.A., 2020. A new formulation for symbolic regression to identify physico-chemical laws from experimental data. Chem. Eng. J. 387, 123412 https://doi.org/10.1016/j.cej.2019.123412.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Édouard, D., 2011. Scikit-learn: machine learning in python. J. Mach. Learn. Res. 12, 2825–2830.

Petsagkourakis, P., Sandoval, I.O., Bradford, E., Zhang, D., del Rio-Chanona, E.A., 2020. Reinforcement learning for batch bioprocess optimization. Comput. Chem. Eng. 133, 106649 https://doi.org/10.1016/j.compchemeng.2019.106649.

Psichogios, D.C., Ungar, L.H., 1992. A hybrid neural network-first principles approach to process modeling. AIChE J. 38, 1499–1511. https://doi.org/10.1002/aic.690381003.

Rivera, E.C., Costa, A.C., Andrade, R.R., Atala, D.I.P., Maugeri, F., Maciel Filho, R., 2007. Development of adaptive modeling techniques to describe the temperature-dependent kinetics of biotechnological processes. Biochem. Eng. J. 36, 157–166. https://doi.org/10.1016/j.bej.2007.02.011.

Sadino-Riquelme, M.C., Rivas, J., Jeison, D., Hayes, R.E., Donoso-Bravo, A., 2020. Making sense of parameter estimation and model simulation in bioprocesses. Biotechnol. Bioeng. 117, 1357–1366. https://doi.org/10.1002/bit.27294.

Savageau, M.A., 1969a. Biochemical systems analysis. J. Theor. Biol. 25, 365–369. https://doi.org/10.1016/S0022-5193(69)80026-3.

Savageau, M.A., 1969b. Biochemical systems analysis. J. Theor. Biol. 25, 370–379. https://doi.org/10.1016/S0022-5193(69)80027-5.

Savageau, M.A., 1970. Biochemical systems analysis. J. Theor. Biol. 26, 215–226. https://doi.org/10.1016/S0022-5193(70)80013-3.

Savitzky, A., Golay, M.J.E., 1964. Smoothing and differentiation. Anal. Chem 36, 1627–1639.

Schmidt, M., Lipson, H., 2009. Distilling free-form natural laws from experimental data. Science 324, 81–85. https://doi.org/10.1126/science.1165893.

Sha, S., Huang, Z., Wang, Z., Yoon, S., 2018. Mechanistic modeling and applications for CHO cell culture development and production. Curr. Opin. Chem. Eng. 22, 54–61. https://doi.org/10.1016/j.coche.2018.08.010.

Smith, E.M.B., Pantelides, C.C., 1999. A symbolic reformulation/spatial branch-and-bound algorithm for the global optimisation of nonconvex MINLPs. Comput. Chem. Eng. 23, 457–478. https://doi.org/10.1016/S0098-1354(98)00286-5.

Sun, W., Braatz, R.D., 2020. ALVEN: algebraic learning via elastic net for static and dynamic nonlinear model identification. Comput. Chem. Eng. 143, 107103 https://doi.org/10.1016/j.compchemeng.2020.107103.

Tawarmalani, M., Sahinidis, N.V., 2002. Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming, Nonconvex Optimization and Its Applications. Springer US, Boston, MA. https://doi.org/10.1007/978-1-4757-3532-1.

Taylor, C.J., Booth, M., Manson, J.A., Willis, M.J., Clemens, G., Taylor, B.A., Chamberlain, T.W., Bourne, R.A., 2021. Rapid, automated determination of reaction models and kinetic parameters. Chem. Eng. J. 413, 127017 https://doi.org/10.1016/j.cej.2020.127017.

Tjoa, I.B., Biegler, L.T., 1991. Simultaneous solution and optimization strategies for parameter estimation of differential-algebraic equation systems. Ind. Eng. Chem. Res. 30, 376–385. https://doi.org/10.1021/ie00050a015.

Tonner, P.D., Darnell, C.L., Engelhardt, B.E., Schmid, A.K., 2017. Detecting differential growth of microbial populations with Gaussian process regression. Genome Res. 27, 320–333. https://doi.org/10.1101/gr.210286.116.

TuringBot, 2023. Symbolic Regression Software.

Turton, R., Shaeiwitz, J.A., Bhattacharyya, D., Whiting, W.B., 2018. Analysis. Synthesis and Design of Chemical Processes, Pearson Prentice Hall.

Udrescu, S.M., Tegmark, M., 2019. AI feynman: A Physics-inspired method for symbolic regression. arXiv.

Vázquez, D., Guimerà, R., Sales-Pardo, M., Guillén-Gosálbez, G., 2022. Automatic modeling of socioeconomic drivers of energy consumption and pollution using Bayesian symbolic regression. Sustain. Prod. Consump. 30, 596–607. https://doi.org/10.1016/j.spc.2021.12.025.

Vladislavleva, E., Friedrich, T., Neumann, F., Wagner, M., 2013. Predicting the energy output of wind farms based on weather data: Important variables and their correlation. Renew. Energy 50, 236–243. https://doi.org/10.1016/j.renene.2012.06.036.

Voit, E.O., Almeida, J., 2004. Decoupling dynamical systems for pathway identification from metabolic profiles. Bioinformatics 20, 1670–1681. https://doi.org/10.1093/bioinformatics/bth140.

von Stosch, M., Oliveira, R., Peres, J., Feyo de Azevedo, S., 2014. Hybrid semi-parametric modeling in process systems engineering: Past, present and future. Comput. Chem. Eng. 60, 86–101. https://doi.org/10.1016/j.compchemeng.2013.08.008.

Weng, B., Song, Z., Zhu, R., Yan, Q., Sun, Q., Grice, C.G., Yan, Y., Yin, W.-J., 2020. Simple descriptor derived from symbolic regression accelerating the discovery of new perovskite catalysts. Nat. Commun. 11, 3513. https://doi.org/10.1038/s41467-020-17263-9.

Willis, M.J., Montague, G.A., Peel, C., 1995. On the application of artificial neural networks to process control. Applic. Neural Netw. 191–219.

Willis, M.J., von Stosch, M., 2017. Simultaneous parameter identification and discrimination of the nonparametric structure of hybrid semi-parametric models. Comput. Chem. Eng. 104, 366–376. https://doi.org/10.1016/j.compchemeng.2017.05.005.

Wilson, Z.T., Sahinidis, N.V., 2017. The ALAMO approach to machine learning. Comput. Chem. Eng. 106, 785–795. https://doi.org/10.1016/j.compchemeng.2017.02.010.

Zhang, S., Androulakis, I.P., Ierapetritou, M.G., 2013. A hybrid kinetic mechanism reduction scheme based on the on-the-fly reduction and quasi-steady-state approximation. Chem. Eng. Sci. 93, 150–162. https://doi.org/10.1016/j.ces.2013.01.066.

Zhang, D., Dechatiwongse, P., Del-Rio-Chanona, E.A., Hellgardt, K., Maitland, G.C., Vassiliadis, V.S., 2015. Analysis of the cyanobacterial hydrogen photoproduction process via model identification and process simulation. Chem. Eng. Sci. 128, 130–146. https://doi.org/10.1016/j.ces.2015.01.059.

Zhang, D., Del Rio-Chanona, E.A., Petsagkourakis, P., Wagner, J., 2019. Hybrid physics-based and data-driven modeling for bioprocess online simulation and optimization. Biotechnol. Bioeng. 116, 2919–2930. https://doi.org/10.1002/bit.27120.

Zhang, D., Savage, T.R., Cho, B.A., 2020. Combining model structure identification and hybrid modelling for photo-production process predictive simulation and optimisation. Biotechnol. Bioeng. 117, 3356–3367. https://doi.org/10.1002/bit.27512.