









EnzymeML—a data exchange format for biocatalysis and enzymology

Jan Range¹ , Colin Halupczok¹, Jens Lohmann¹ , Neil Swainston² , Carsten Kettner³ , Frank T. Bergmann⁴ , Andreas Weidemann⁵, Ulrike Wittig⁵ , Santiago Schnell^{6,7}  and Jürgen Pleiss¹ 

1 Institute of Biochemistry and Technical Biochemistry, University of Stuttgart, Germany

2 Institute of Systems, Molecular and Integrative Biology, University of Liverpool, UK

3 Beilstein-Institut, Frankfurt am Main, Germany

4 BioQUANT/COS, Heidelberg University, Germany

5 Heidelberg Institute for Theoretical Studies, Germany

6 Department of Molecular & Integrative Physiology, University of Michigan Medical School, Ann Arbor, MI, USA

7 Department of Computational Medicine & Bioinformatics, University of Michigan Medical School, Ann Arbor, MI, USA

Keywords

biocatalysis; bioinformatics; data exchange; enzymology; FAIR data principles; Python; research data management; Systems Biology Markup Language; XML

Correspondence

J. Pleiss, Institute of Biochemistry and Technical Biochemistry, University of Stuttgart, Allmandring 31, Stuttgart 70569, Germany

Tel: +49-711-68563191

E-mail: juergen.pleiss@itb.uni-stuttgart.de

(Received 22 September 2021, revised 15 November 2021, accepted 9 December 2021)

doi:10.1111/febs.16318

EnzymeML is an XML-based data exchange format that supports the comprehensive documentation of enzymatic data by describing reaction conditions, time courses of substrate and product concentrations, the kinetic model, and the estimated kinetic constants. EnzymeML is based on the Systems Biology Markup Language, which was extended by implementing the STRENDA Guidelines. An EnzymeML document serves as a container to transfer data between experimental platforms, modeling tools, and databases. EnzymeML supports the scientific community by introducing a standardized data exchange format to make enzymatic data findable, accessible, interoperable, and reusable according to the FAIR data principles. An application programming interface in Python supports the integration of software tools for data acquisition, data analysis, and publication. The feasibility of a seamless data flow using EnzymeML is demonstrated by creating an EnzymeML document from a structured spreadsheet or from a STRENDA DB database entry, by kinetic modeling using the modeling platform COPASI, and by uploading to the enzymatic reaction kinetics database SABIO-RK.

Introduction

Enzyme catalysis and enzymology provide a powerful toolbox for sustainable synthesis routes and innovative solutions for bio-based chemistry. A better understanding of cellular biochemistry and the comprehensive biochemical characterization of the desired enzyme-catalyzed reaction enable novel approaches in enzyme engineering and process development [1].

Standardization of reporting of enzymatic data and metadata is considered to be pivotal to accelerating bioprocess development and reducing costs [2], facilitating sharing, analysis, and reuse of data and thus enabling quality control and reproducibility of experiments [3]. Therefore, a major challenge for enzymology and biocatalysis lies in the current practices of

Abbreviations

API, application programming interface; CSV, comma-separated values; MIRIAM, Minimal Information Required in the Annotation of Models; SBML, Systems Biology Markup Language; SBO, Systems Biology Ontology; SiLA, Standards in Laboratory Automation; STRENDA, Standards for Reporting Enzymology Data; XML, eXtensible Markup Language.

dealing with experimental data in academic laboratories [4]. In most academic research groups, data acquisition, curation, and documentation are performed manually without a universally accepted standard across laboratories. Data and metadata are typically stored in *ad hoc* repositories, such as paper lab notebooks, spreadsheets in different formats, and semistructured text files containing custom annotations. Experimental or computational data are often poorly annotated, lacking a complete description of the acquisition and analysis procedures, or associated metadata. Despite previous efforts to address these issues [5], raw data are rarely available in machine-readable, even less in machine-actable format, preventing their further analysis and third-party validation. As it stands, the process of data acquisition, data analysis, and documentation is time-consuming and error-prone, as is the recovery and interpretation of legacy data in most academic laboratories. Consequently, both the quality and the completeness of data and metadata solely rely on the experimenter's expertise and care.

Recent meta-research results indicate that the reproducibility crisis in the biomedical sciences is caused by the lack of standards in reporting and sharing experimental protocols, results, and data [6,7]. This is also true for enzymology and biocatalysis. An empirical analysis of published papers investigating enzyme function illustrates how critical information for the reproducibility of experimental finding is missing in the literature [8]; the missing information includes the concentration of enzyme and/or substrates, the composition of the entire buffer systems including the identity of counter-ions, pH values, and assay temperatures.

The incompleteness of metadata prevents the interpretation of inconsistent data arising from different studies. An example of such variability is demonstrated in a large global benchmark study [9], in which the variability of a dissociation constant for a protein–protein interaction determined by 150 participants using a general protocol exceeded its average value. When investigators were given detailed fixed protocols, the dissociation constants still varied up to 20% [10,11]. This kind of irreproducibility is commonplace in enzymology and has an essential impact on subsequent research.

In response to the reproducibility crisis, the scientific community is developing and adopting new guidelines for reporting experimental protocols and statistical analysis. Scientific journals are responding accordingly [12], and there has been a recommendation to modify the academic reward system by recognizing scientists who aligned with best practices for reproducible

research [13]. Initiatives such as the German National Research Data Infrastructure develop an infrastructure for standardized research data exchange [14], the Standards in Laboratory Automation consortium (SiLA) provide a framework for the exchange, integration, sharing, and retrieval of electronic laboratory information (https://sila2.gitlab.io/sila_base/), and data repositories such as Zenodo and Dataverse enable data sharing [15]. Efforts in standardization and data reproducibility have been long established in other 'omics fields, with standard exchange formats for transcriptomics [16], proteomics [17], and metabolomics [18] data becoming increasingly developed and adopted over the last twenty years. However, in biocatalysis and enzymology exchange standards or software support to aid data analysis, management, and sharing is still absent, and raw experimental data such as the time dependency of substrate or product concentration, derived data such as kinetic parameters, and metadata such as reaction conditions or the kinetic model are typically reported in plain text, figures, or tables [19]. Currently, kinetic parameters and corresponding information about the reactions, enzymes, and experimental conditions are extracted and annotated manually from scientific publications and inserted into databases such as SABIO-RK [20] or BRENDA [21] to structure and standardize the data. Missing information such as unambiguous external identifiers is added manually by database curators. As a first step for the standardized reporting of enzyme function data, the enzymology and biocatalysis community has established the Standards for Reporting Enzymology Data (STREND A) Guidelines, which provide the minimum information necessary to describe assay conditions and enzyme activity data [22,23]. Currently, more than 55 international biochemistry journals have included adherence to the STREND A Guidelines in their instructions for authors reporting enzymology data. As the guidelines are rarely enforced by the journals and as essential information is still omitted in the literature, the web-based software tool STREND A DB was developed that incorporates the guidelines [24]. STREND A DB has been established as a public database to support authors checking the completeness of their data upon submission of their manuscript and to provide public access to data on reaction conditions and kinetic parameters of an experiment. However, the upload of data is performed manually via a graphical user interface, and the process from data acquisition to kinetic modeling and publication is still time-consuming and error-prone. Most importantly, original data such as the measured time course of substrate and product concentrations is not reported or has to be extracted from figures, thus preventing the reuse of original data

for kinetic modeling. Not only is published data incomplete and inaccessible, but also unpublished research data and metadata are stored by research group members with insufficient documentation and annotation. In addition, the current data management prevents researchers from upscaling their experimental designs to high-throughput biocatalytic approaches by using pipetting robots [25] or flow reactors [26], and hinders the comprehensive study of the multidimensional parameter space of biocatalytic reactions.

Here, we introduce EnzymeML, a data exchange format for biocatalysis and enzymology, which makes enzyme data findable, accessible, interoperable, and reusable in accordance to the FAIR data principles [27]. An application programming interface (API) provides a Python library to integrate applications and databases and to enable a seamless data flow from the bench to kinetic modeling tools and publication platforms. The machine-actable EnzymeML document on data and metadata of an enzymatic reaction could serve as a micropublication, supplementing the respective scientific paper.

Principles of EnzymeML

EnzymeML has been designed to support data acquisition, data analysis, and sharing of data by providing a

standardized exchange format for enzymatic data (Fig. 1). EnzymeML is written in eXtensible Markup Language (XML) and comprises the most relevant data and metadata from measurement and modeling. Given the ubiquity of XML, vast amounts of software are available that read, write, manipulate, and process XML documents. More importantly, XML allows for the specification of a machine-actable schema, which ensures interoperability. The central core of EnzymeML is the Systems Biology Markup Language (SBML), an established data format in systems biology for sharing, evaluating, and developing models of biochemical reaction networks [28]. Furthermore, EnzymeML extends SBML toward the inclusion of measurement data as comma-separated files (CSV), which together with the XML part are an integral part of the resulting EnzymeML archive. Interoperability with existing software tools and databases is achieved by applying a common terminology and vocabulary that allow the integration of data from various sources for subsequent processing, because many of the concepts supported by SBML—educts, products, reactions, modifiers, reaction rates—are common to enzymology and biocatalysis. However, EnzymeML goes beyond SBML because it serves to describe the effect of enzyme sequence and reaction medium to an enzymatic reaction.

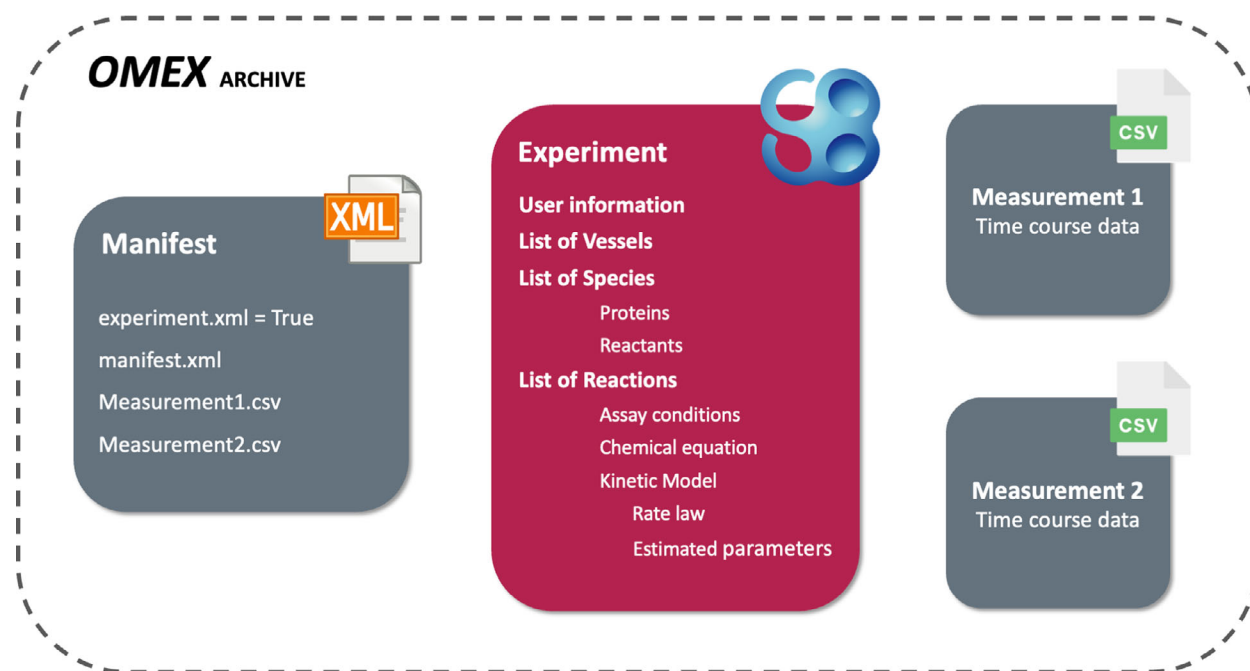


Fig. 1. Structure of an EnzymeML document. An EnzymeML document is a ZIP container in OMEX format and contains the experiment file (SBML) with the metadata of the experiment, the kinetic model, and the estimated kinetic parameters, and the measurement files (CSV) with the time courses of substrate and product concentrations. The manifest file (XML) lists the content of the ZIP container.

EnzymeML implements the STREND A guidelines

For the complete machine-actable description of an enzymatic experiment, the STREND A Guidelines were incorporated. In addition, metadata on the experiments and the kinetic model were included, resulting in a comprehensive data exchange format that comprises 71 attributes (Table 1). The current version of EnzymeML includes all STREND A fields with a controlled vocabulary or values and excludes fields with plain text such as experiment methodology, in order to make EnzymeML structured and machine actable.

EnzymeML was built within the framework of several internationally recognized standards

SBML is a widely used XML-based markup language and describes almost 50% of the attributes (Table 1). MathML was applied to describe the equation of the kinetic model [28], and the guidelines on Minimal Information Required in the Annotation of Models (MIRIAM) [29] were applied for the consistent annotation of components such as reactants, products, and enzymes, using terms from external data repositories such as ChEBI [30] and UniProt [31]. A controlled, relational vocabulary of terms, the Systems Biology Ontology (SBO) [32], was used to define reactants, inhibitors, activators, parameters, and the kinetic model. All files are combined into a single document using the OMEX format [33]. Furthermore, EnzymeML uses the Distributions package for SBML Level 3 (http://sbml.org/Documents/Specifications/SBML_Level_3/Packages/distrib) to support the specification of ranges of initial concentrations.

EnzymeML is extensible

EnzymeML-specific attributes are added to SBML using the “annotation” element, which supports metadata specific to enzymology to be added to the XML document whilst maintaining compatibility with SBML. EnzymeML documents are valid SBML files and can therefore be used and manipulated by many software tools that support the SBML format.

EnzymeML is platform independent

XML has been designed to store and transfer data, and is fully agnostic to the operating system and supported by different programming languages. Comma-Separated Values (CSV) is a platform-independent text file format, which was designed for storing and transporting data structured in tables. CSV-formatted files

can be read by the modeling platform COPASI [34] and by spreadsheet editors such as Excel. All components of EnzymeML are self-descriptive (SBML, MathML, OMEX), which makes EnzymeML human readable and machine actable.

EnzymeML is modular

EnzymeML was developed as a container for experimental and modeling data, supporting a seamless data flow between different applications (Fig. 2). Data obtained from an experiment and metadata on experimental conditions can be stored by the experimentalist in a spreadsheet, which is convertible into EnzymeML using the API. Longer term, it is hoped that electronic lab notebooks, laboratory information management systems, and enzymology software will support the format. The EnzymeML document contains sufficient experimental data to allow for the estimation of the kinetic parameters by modeling platforms such as COPASI [34], BioCatNet [35], or Matlab™. Kinetic parameters can then be included in the EnzymeML document. As a consequence, enzyme assay data may be easily reanalyzed and checked with a range of data fitting algorithms, increasing reusability and confidence in both the experimental data and reported kinetic parameters.

EnzymeML enables data publication in compliance with FAIR principles

An EnzymeML document stores comprehensive information about data and metadata of an enzymatic experiment: the experimental conditions, the time course of substrate and product concentration, the kinetic model, and the estimated kinetic parameters, thus making the experiment and its analysis reproducible. Upon publication, it is recommended to use EnzymeML documents as supplementary material. By depositing EnzymeML documents on platforms such as FAIRDOMHub [36] or Dataverse [37] using a digital object identifier, EnzymeML documents are findable and accessible. EnzymeML documents also include references to the scientific publications from which they arose, providing contextual information.

Structure of EnzymeML documents

An EnzymeML document is a ZIP container in the widely used OMEX format [33]. It consists of three file types: a file using SBML to describe the experimental reaction conditions, the kinetic model, and the kinetic parameters, CSV (comma-separated values)-formatted files to store the time courses of substrate and product

Table 1. List of attributes derived from the STRENDA recommendations. Mandatory fields are marked with an asterisk (*).

STRENDA guidelines	EnzymeML
List level 1A	
<i>Identity of the enzyme</i>	
Name of reaction catalyst*	SBML Species:Name
EC number	EnzymeML Protein:ECNumber
Sequence accession number	EnzymeML Protein:seqAcc
Organism/species and strain	EnzymeML Protein:organism
<i>Additional information on the enzyme</i>	
Isoenzyme (variant)	Not included
Tissue	Not included
Organelle	Not included
Localization	Not included
Post-translational modification	Not included
<i>Preparation</i>	
Description	Not included
Artificial modification	Not included
enzyme or protein purity	Not included
Metalloenzyme	Not included
<i>Storage conditions</i>	
Storage temperature	Not included
Atmosphere if not air	Not included
pH	Not included
At which temperature was the pH measured?	Not included
Buffer and concentrations (including counter-ion)	Not included
Metal salt(s) and concentrations	Not included
Other components	Not included
Enzyme/protein concentration	Not included
<i>Assay conditions</i>	
Substrate purity	Not included
Measured reaction*	SBML Reaction:name
Assay temperature	EnzymeML Conditions:temperature
Assay pressure	Not included
Atmosphere if not air	Not included
Assay pH	EnzymeML Conditions:pH
Buffer and concentrations*	SBML Species:name/ initialConcentration
Metal salt(s) and concentrations*	SBML Species:name/ initialConcentration
Other assay components*	SBML Species:name/ initialConcentration
Coupled assay components	Not included
Substrate and concentration ranges*	SBML Species:name/ initialConcentration EnzymeML InitConcs:initConc
Enzyme/protein concentration*	SBML Species:name/ initialConcentration
Varied components	EnzymeML InitConcs
Total assay mixture ionic strength	Not included

Table 1. (Continued).

STRENDA guidelines	EnzymeML
<i>Activity</i>	
Initial rates of the reaction measured*	SBML KineticLaw:localParameter
Enzyme activity*	SBML KineticLaw:localParameter
<i>Methodology</i>	
Assay method	Not included
Type of assay	Not included
Reaction stopping	Not included
Direction of the assay	Not included
Reactant determined	Not included
<i>Additional material desirable</i>	
Free metal cation	SBML:Species Modifier
Reaction equilibrium constant*	SBML KineticLaw:localParameter
List level 1B	
<i>Required data for all enzyme functional data</i>	
Number of independent experiments	EnzymeML:listOfMeasurements
Precision of measurement	Not included
Referring to subunit or oligomeric form	Not included
<i>Data necessary for reporting kinetic parameters</i>	
k_{cat}	SBML KineticLaw:localParameter
V_{max}	SBML KineticLaw:localParameter
k_{cat}/K_m	SBML KineticLaw:localParameter
K_m	SBML KineticLaw:localParameter
S0.5	SBML KineticLaw:localParameter
Coefficients of cooperativity	SBML KineticLaw:localParameter
How was the given parameter obtained	SBML KineticLaw:localParameter
Model used to determine the parameters	SBML KineticLaw:localParameter
Substrate inhibition (K_i value)	SBML KineticLaw:localParameter
<i>Data required for reporting inhibition and activation data</i>	
Time-dependence and reversibility	SBML KineticLaw:localParameter
Inhibition types (reversible, irreversible)	SBML KineticLaw:localParameter
<i>Additional data in EnzymeML beyond STRENDA guidelines</i>	
Product(s)*	SBML Species:Name
Time course data of substrate and product	CSV
CSV column definition*	EnzymeML:format
Replicate definition*	EnzymeML:replica
Amino acid sequence*	EnzymeML Protein:sequence
General kinetic model	SBML KineticLaw:localParameter
InChI identifier for substrates and products	EnzymeML:inchi
SMILES identifier for substrates and products	EnzymeML:smiles
Literature reference: PubMed ID	EnzymeML:pmid
Literature reference: DOI	EnzymeML:doi
Literature reference: URL	EnzymeML:url

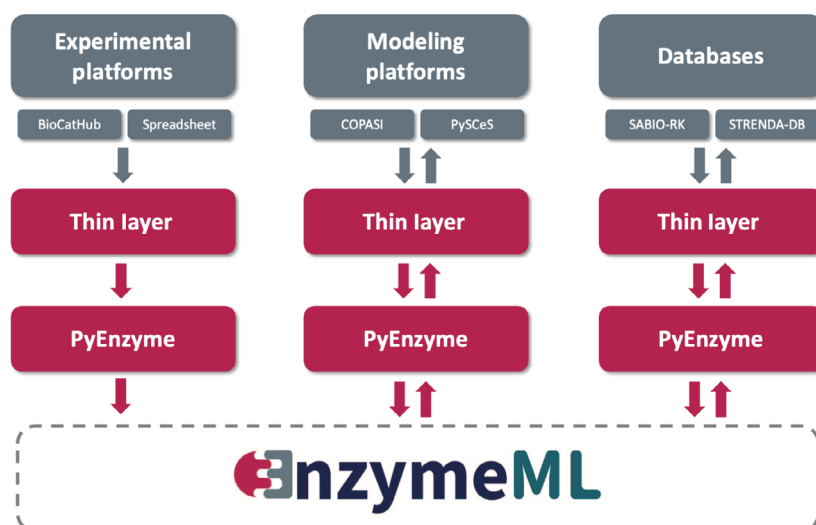


Fig. 2. Integration of software tools. The EnzymeML document serves as a container to transfer data between tools such as experimental platforms, modelling tools, and databases for the publication of enzymatic experiments. The EnzymeML API consists of a Python library PyEnzyme and provides read and write functionalities to the applications. The API is adapted to each application by an application-specific thin API layer.

concentrations, and a manifest file lists the content of the ZIP container (Fig. 1).

The experimental conditions are reported according to the STRENDA recommendations, the kinetic model is described by using MathML and SBML in the experiment file (Fig. S1). This file also describes the format of the CSV-formatted file, which contains the raw time course data. Instead of using headers to describe columns, the complete CSV-formatted file description is done within the SBML file. This approach has the advantage of enabling a comprehensive description of each column, such as measured species, units, and data types, instead of a single header. The SBML file uses two elements, notes and annotation. A notes tag contains human-readable information as plain text, whereas an annotation tag contains structured, machine-actable information. Notes and annotation tags are used to add information, which is required by the STRENDA Guidelines, but not included in SBML, such as protein sequence, pH, or temperature. Thus, this file is a valid SBML document, which contains additional information on enzyme-catalyzed reactions. Furthermore, any information that is not machine-readable but required by the STRENDA Guidelines or not part of the data model can be included to the OMEX archive as an arbitrary file (e.g., methodology or figures). An extensive description of the EnzymeML document structure is available in the [Supporting Information](#).

EnzymeML application programming interface

Although EnzymeML is semi-human-readable, the user is not expected to read or write EnzymeML

documents directly, but to use software to generate EnzymeML documents, which can then be used as a standardized exchange format to transfer data between applications (Fig. 2). APIs to read, write, edit, and visualize EnzymeML have therefore been developed, using the popular programming language Python, to support the development of such software tools. The library PyEnzyme was built based on its respective SBML counterpart libSBML. To simplify the implementation of the libraries for enzyme-catalyzed reactions, the terminology of enzymology and biocatalysis is used, hiding the more systems biology focused SBML terms, while maintaining full compatibility with the SBML format.

The adaption of the API to an application is enabled by an additional thin layer, which maps the objects of the API to the equivalent objects defined within the respective application. Thus, by editing a template, the functionality of reading and writing of EnzymeML can be easily incorporated into an application without the need to modify the API. For five applications (COPASI import/export, STRENDA DB export, BioCatNet export, SABIO-RK import, simulation of time course data), application-specific thin API layers are provided (TL_COPASI, TL_STRENDAML and TL_BioCatNet, respectively). Because the API enables batch processing, management of enzymatic data is scalable, and high throughput strategies of experimentation and data analysis become feasible. By data export in formats such as Pandas DataFrame, large datasets could be analyzed by novel analysis methods based on machine learning.

Upon reading, writing, and visualization of EnzymeML documents, the API controls data

completeness and consistency, such as checking the definition of reactants and proteins upon reading or writing of a reaction, or by checking that scalar properties such as pH are within the necessary range. A specific validation tool guarantees compatibility with SBML. Further application-specific validation tools have been added, such as a STRENDA DB validator to check for compatibility with the STRENDA Guidelines. For more details, readers can find a description of the API below and the [Supporting Information](#).

PyEnzyme provides a functionality to validate an EnzymeML document by using an EnzymeML validation object. The validation object specifies the minimal requirements of a database for all fields present in an EnzymeML document and is hosted at the database. Prior to upload of an EnzymeML document to a database, the validation object is read by PyEnzyme and the compliance of the EnzymeML document with the data model of the database is checked by validating that all mandatory fields are provided by the EnzymeML document. Furthermore, field-specific contents such as value ranges, controlled vocabularies, and ontologies are checked. Database providers can specify mandatory fields and field-specific contents by using the EnzymeML validation spreadsheet (<https://github.com/EnzymeML/PyEnzyme/tree/main/templates>), which is converted to an EnzymeML validation object.

Application of EnzymeML

The power of EnzymeML is demonstrated by selected applications for experimental enzymologists, modelers, and software developers (Fig. 3).

Creating EnzymeML documents from structured spreadsheets

In the absence of a standard format, experimentalists typically store their experimental time course data in a spreadsheet following an *ad hoc* structure. Recently, a CSV-formatted spreadsheet, the BioCatNet template [35], was proposed to store and report experimental data on enzyme-catalyzed reactions according to the STRENDA Guidelines. The API was used to convert the BioCatNet spreadsheet, containing time course data on substrate and product concentration and comprehensive information as the reaction conditions, to EnzymeML. Initially, each field of the respective spreadsheet template was extracted via a thin API layer (TL_BioCatNet) and further processed by the API to an object layer.

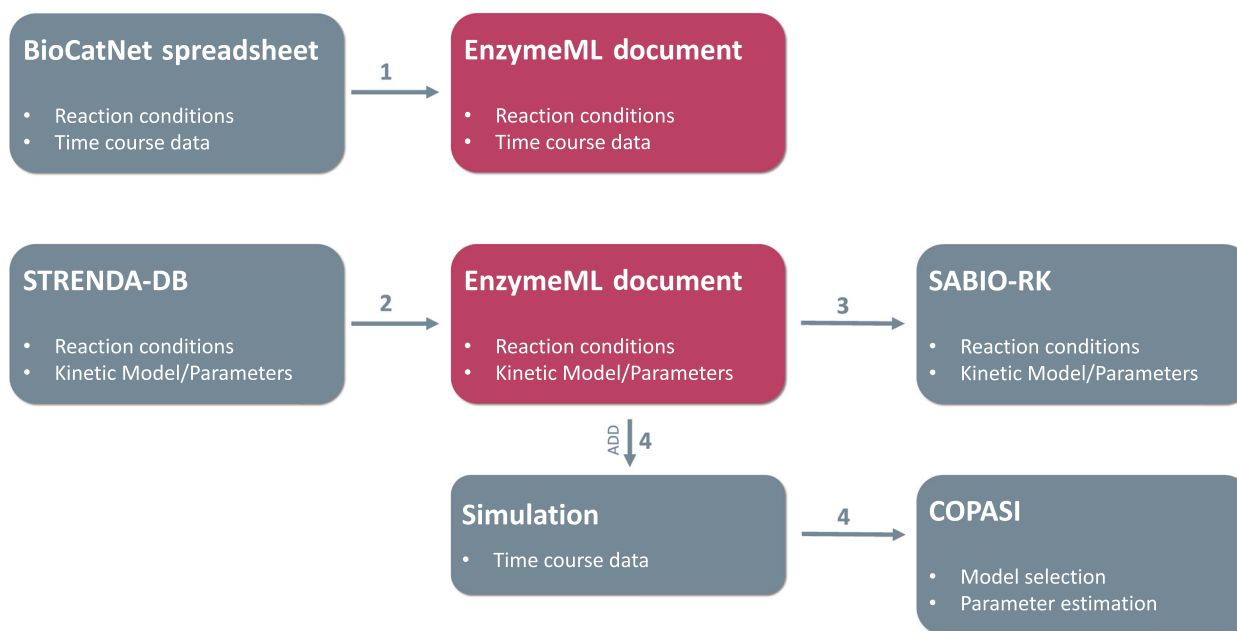


Fig. 3. Applications of EnzymeML. The PyEnzyme API was used for the conversion of a BioCatNet spreadsheet to an EnzymeML document (1) using the Thin Layer TL_BioCatNet. Furthermore, a STRENDA-DB entry was converted to an EnzymeML document by using the Thin Layer TL_STRENDA (2) and uploaded to the database SABIO-RK (3). Finally, this EnzymeML document was used to simulate and add time course data (4) for a parameter estimation using COPASI (4).

Finally, the objects were written to an EnzymeML document (see SI Chapter 3.1).

Creating EnzymeML documents from STRENDA DB entries

STRENDA DB is a database on enzyme-catalyzed reactions, which covers the most important information on reaction conditions and kinetic parameters [24]. The API was used to create an EnzymeML document from a STRENDA DB entry via a STRENDA DB-specific thin API layer (TL_STRENDA) to the object layer using the PyEnzyme library. The resulting EnzymeML document was then created by the API (see SI Chapter 3.2).

Upload of EnzymeML documents to SABIO-RK

SABIO-RK is a curated database that contains information about biochemical reactions, their kinetic rate equations with parameters, and experimental conditions [20]. An already existing SBML parser for the upload of SBML models in SABIO-RK was extended to read the additional annotations in EnzymeML to allow the import of EnzymeML documents and to create a new SABIO-RK entry in the internal curation interface (see SI Chapter 3.3). SABIO-RK curators check the new SABIO-RK entries for consistency and completeness according to the SABIO-RK requirements before they are finally submitted to the public SABIO-RK database.

Editing of EnzymeML: simulation of time course data from kinetic parameters

STRENDA-DB entries provide for an enzyme-catalyzed reaction the kinetic parameters K_M and k_{cat} assuming a Michaelis–Menten model and the concentration range of the substrate. However, they are lacking information on the product and on the time course of substrate or product concentrations. PyEnzyme was used to add the product and time course data to the EnzymeML document (see SI Chapter 3.4). By a single function in the API, the time course of substrate concentrations was simulated from the kinetic parameters for initial concentrations from 0 to 0.5 mM for a time interval of 200 s to visualize kinetic behavior and study the effect of kinetic parameters.

Kinetic modeling of EnzymeML data by COPASI

COPASI is a modeling and simulation environment, which supports the OMEX format [34]. Using the PyEnzyme library and a COPASI-specific thin API layer (TL_COPASI), the time course data (measured concentrations of substrate or product) are loaded into COPASI. Within COPASI, different kinetic laws are applied, kinetic parameters are estimated, and plots are generated to assess the result. The selected kinetic model and the estimated kinetic parameters are then added to the EnzymeML document (see SI Chapter 3.5).

Outlook

For many years, researchers worldwide from various disciplines have recognized that data published in the literature are not reliable unless the full set of information required is provided [23]. Therefore, the FAIR principles were introduced to encourage the comprehensive documentation of structured metadata in all stages of their life cycle in order to guarantee reproducibility of experiments and to enable reuse of results. A discipline-specific standard data exchange format such as EnzymeML therefore provides three functionalities to optimize research in biocatalysis and enzymology: it allows the experimentalist to collect data and metadata in a structured format for data analysis; it allows project partners to transfer data and metadata between different sites and different applications; and it enables findable and reusable publication and archiving of data and metadata [38].

Currently, data flow from laboratory to publication is a challenging and complex process involving diverse processing stages, and numerous steps of data reformatting and manual input. Such manual approaches are becoming increasingly unsustainable, especially in the light of recent advances in miniaturization and robotics, which have enabled the intensive, high-throughput screening of enzymes and process conditions [39]. Such technological advances foster the discovery of novel enzymatic systems and the (retro-) synthetic design of enzyme-catalyzed reaction cascades through integration of systematic data acquisition, data analysis, and simulation [40].

In a fully digitalized biocatalytic laboratory, an electronic lab notebook supports researchers at the bench to plan experiments and to collect experimental data and metadata [41,42], all laboratory devices are connected by a common standard [43], various modeling

and data analysis tools are combined to analyze the data [34,35,44], and the results are uploaded to searchable repositories without manual intervention [20,24].

With the integration of EnzymeML, the interoperability and compatibility of the tools and databases will be improved, and possible current limitations and inconsistencies in the data models of the repositories will be resolved. In the future, EnzymeML will be combined with other standards to enrich the data model and to connect disciplines that are relevant to enzymology. Incorporating AniML [43] or SiLA enables access to laboratory devices, and ThermoML [42] offers a comprehensive description of the reaction medium.

The introduction of EnzymeML as a uniform transport container for experimental data and metadata will encourage the development of software infrastructure built on this standardized format to greatly simplify the process of analyzing and publishing enzymology data, supporting the increasing experimental throughput, and ultimately promoting the digitalization of the fields of enzymology and biocatalysis [14].

Acknowledgements

The authors acknowledge Michael Hucka (California Institute of Technology) for inspiring discussions and constructive comments during the meetings of the EnzymeML Development Team and Patrick Buchholz (University of Stuttgart) for his support with BioCat-Net. JP acknowledges funding from the Deutsche Forschungsgemeinschaft (DFG, grants EXC310 and EXC2075). NS acknowledges funding from the Biotechnology and Biological Sciences Research Council (BBSRC) under grant “GeneORator: a novel and high-throughput method for the synthetic biology-based improvement of any enzyme” (BB/S004955/1) and from the University of Liverpool. AW and UW acknowledge funding from the Klaus Tschira Foundation and the German Federal Ministry of Education and Research within de.NBI (031A540). FTB acknowledges funding from the German Federal Ministry of Education and Research within de.NBI (031L0104A). We are grateful for the support of Beilstein-Institut zur Förderung der Chemischen Wissenschaften by supporting discussions through its Beilstein Enzymology Symposia and STRENDA Commission Meetings. Open access funding enabled and organized by ProjektDEAL.

Conflict of interest

The authors declare no conflict of interest.

Author contributions

JR conceptualized the project, developed the data format and new software, and wrote the manuscript; CH and JL developed the data format; NS, CK, FTB, UW, and SS conceived the project and wrote the manuscript; AW developed new software; JP conceptualized and supervised the project and wrote the manuscript.

Data availability statement

The XML Schema, the API, templates of the thin API layer, and all files mentioned in the Application section are available at <https://github.com/EnzymeML> and <https://zenodo.org/record/5021263#.YNQPtS223BI>.

References

- Pellis A, Cantone S, Ebert C, Gardossi L. Evolving biocatalysis to meet bioeconomy challenges and opportunities. *N Biotechnol.* 2018;**40**:154–69.
- Decoene T, De Paepe B, Maertens J, Coussemont P, Peters G, De Maeseneire SL, et al. Standardization in synthetic biology: an engineering discipline coming of age. *Crit Rev Biotechnol.* 2018;**38**:647–56.
- Lapatas V, Stefanidakis M, Jimenez RC, Via A, Schneider MV. Data integration in biological research: an overview. *J Biol Res.* 2015;**22**:1–16.
- Kettner C, Cornish-Bowden A. Quo Vadis, enzymology data? Introductory remarks. *Perspect Sci.* 2014;**1**:1–6.
- Swainston N, Golebiewski M, Messiha HL, Malys N, Kania R, Kengne S, et al. Enzyme kinetics informatics: from instrument to browser. *FEBS J.* 2010;**277**:3769–79.
- Stark PB. No reproducibility without preproducibility. *Nature.* 2018;**557**:613.
- Baker M, Penny D. Is there a reproducibility crisis? *Nature.* 2016;**533**:452–4.
- Halling P, Fitzpatrick PF, Raushel FM, Rohwer J, Schnell S, Wittig U, et al. An empirical analysis of enzyme function reporting for experimental reproducibility: missing/incomplete information in published papers. *Biophys Chem.* 2018;**242**:22–7.
- Rich RL, Papalia GA, Flynn PJ, Furneisen J, Quinn J, Klein JS, et al. A global benchmark study using affinity-based biosensors. *Anal Biochem.* 2009;**386**:194–216.
- Cannon MJ, Papalia GA, Navratilova I, Fisher RJ, Roberts LR, Worthy KM, et al. Comparative analyses of a small molecule/enzyme interaction by multiple users of Biacore technology. *Anal Biochem.* 2004;**330**:98–113.
- Myszka DG, Abdiche YN, Arisaka F, Byron O, Eisenstein E, Hensley P, et al. The ABRF-MIRG'02

- study: assembly state, thermodynamic, and kinetic analysis of an enzyme/inhibitor interaction. *J Biomol Tech.* 2003;**14**:247–69.
- 12 McNutt M. Journals unite for reproducibility. *Science.* 2014;**346**:679.
- 13 Ioannidis JPA. How to make more published research true. *PLoS Med.* 2014;**11**:e1001747.
- 14 Wulf C, Beller M, Boenisch T, Deutschmann O, Hanf S, Kockmann N, et al. A unified research data infrastructure for catalysis research – challenges and concepts. *ChemCatChem.* 2021;**13**:3223–36.
- 15 Wilkinson MD, Verborgh R, da Silva Santos LOB, Clark T, Swertz MA, Kelpin FDL, et al. Interoperability and FAIRness through a novel combination of Web technologies. *PeerJ Comput Sci.* 2017;**2017**:e110.
- 16 Spellman PT, Miller M, Stewart J, Troup C, Sarkans U, Chervitz S, et al. Design and implementation of microarray gene expression markup language (MAGE-ML). *Genome Biol.* 2002;**3**:RESEARCH0046.
- 17 Pedrioli PGA, Eng JK, Hubley R, Vogelzang M, Deutsch EW, Raught B, et al. A common open representation of mass spectrometry data and its application to proteomics research. *Nat Biotechnol.* 2004;**22**:1459–66.
- 18 Larralde M, Lawson TN, Weber RJM, Moreno P, Haug K, Rocca-Serra P, et al. mzML2ISA & nmrML2ISA: generating enriched ISA-Tab metadata files from metabolomics XML data. *Bioinformatics.* 2017;**33**:2598–600.
- 19 Wittig U, Kania R, Bittkowski M, Wetsch E, Shi L, Jong L, et al. Data extraction for the reaction kinetics database SABIO-RK. *Perspect Sci.* 2014;**1**:33–40.
- 20 Wittig U, Kania R, Golebiewski M, Rey M, Shi L, Jong L, et al. SABIO-RK – database for biochemical reaction kinetics. *Nucleic Acids Res.* 2011;**40**:D790–6.
- 21 Schomburg I, Chang A, Schomburg D. BRENDA, enzyme data and metabolic information. *Nucleic Acids Res.* 2002;**30**:47–9.
- 22 Apweiler R, Armstrong R, Bairoch A, Cornish-Bowden A, Halling PJ, Hofmeyr J-HS, et al. A large-scale protein-function database. *Nat Chem Biol.* 2010;**6**:785.
- 23 Tipton KF, Armstrong RN, Bakker BM, Bairoch A, Cornish-Bowden A, Halling PJ, et al. Standards for reporting enzyme data: the STRENDA Consortium: what it aims to do and why it should be helpful. *Perspect Sci.* 2014;**1**:131–7.
- 24 Swainston N, Baici A, Bakker BM, Cornish-Bowden A, Fitzpatrick PF, Halling P, et al. STRENDA DB: enabling the validation and sharing of enzyme kinetics data. *FEBS J.* 2018;**285**:2193–204.
- 25 Dörr M, Fibinger MPC, Last D, Schmidt S, Santos-Aberturas J, Böttcher D, et al. Fully automatized high-throughput enzyme library screening using a robotic platform. *Biotechnol Bioeng.* 2016;**113**:1421–32.
- 26 Ringborg RH, Toftgaard Pedersen A, Woodley JM. Automated determination of oxygen-dependent enzyme kinetics in a tube-in-tube flow reactor. *ChemCatChem.* 2017;**9**:3285–8.
- 27 Wilkinson MD, Dumontier M, Aalbersberg JJ, Appleton G, Axton M, Baak A, et al. The FAIR guiding principles for scientific data management and stewardship. *Sci Data.* 2016;**3**:160018.
- 28 Hucka M, Bergmann FT, Dräger A, Hoops S, Keating SM, Le Novère N, et al. The Systems Biology Markup Language (SBML): language specification for level 3 version 2 core. *J Integr Bioinform.* 2018;**15**:20170081. <https://doi.org/10.1515/jib-2017-0081>
- 29 Le NN, Finney A, Hucka M, Bhalla US, Campagne F, Collado-Vides J, et al. Minimum information requested in the annotation of biochemical models (MIRIAM). *Nat Biotechnol.* 2005;**23**:1509–15.
- 30 Hastings J, Owen G, Dekker A, Ennis M, Kale N, Muthukrishnan V, et al. ChEBI in 2016: improved services and an expanding collection of metabolites. *Nucleic Acids Res.* 2016;**44**:D1214–9.
- 31 The UniProt Consortium. UniProt: the universal protein knowledgebase. *Nucleic Acids Res.* 2018;**46**:2699.
- 32 Courtot M, Juty N, Knüpfner C, Waltemath D, Zhukova A, Dräger A, et al. Controlled vocabularies and semantics in systems biology. *Mol Syst Biol.* 2011;**7**:543.
- 33 Bergmann FT, Adams R, Moodie S, Cooper J, Glont M, Golebiewski M, et al. COMBINE archive and OMEX format: one file to share all information to reproduce a modeling project. *BMC Bioinformatics.* 2014;**15**:369.
- 34 Hoops S, Sahle S, Gauges R, Lee C, Pahle J, Simus N, et al. COPASI – a complex pathway simulator. *Bioinformatics.* 2006;**22**:3067–74.
- 35 Buchholz PCF, Ohs R, Spiess AC, Pleiss J. Progress curve analysis within BioCatNet: comparing kinetic models for enzyme-catalyzed self-ligation. *Biotechnol J.* 2019;**14**:1–8.
- 36 Wolstencroft K, Krebs O, Snoep JL, Stanford NJ, Bacall F, Golebiewski M, et al. FAIRDOMHub: a repository and collaboration environment for sharing systems biology research. *Nucleic Acids Res.* 2017;**45**:D404–7.
- 37 Crosas M. The dataverse network®: an open-source application for sharing, discovering and preserving data. *D-Lib Mag.* 2011;**17**. <https://doi.org/10.1045/january2011-crosas>
- 38 Pleiss J. Standardized data, scalable documentation, sustainable storage – EnzymeML as a basis for FAIR data management in biocatalysis. *ChemCatChem.* 2021;**13**:3909–13.
- 39 Fernandes P. Miniaturization in biocatalysis. *Int J Mol Sci.* 2010;**11**:858–79.

- 40 Rabe KS, Müller J, Skoupi M, Niemeyer CM. Cascades in compartments: en route to machine-assisted biotechnology. *Angew Chem Int Ed Engl.* 2017;**56**:13574–89.
- 41 Barillari C, Ottoz DSM, Fuentes-Serna JM, Ramakrishnan C, Rinn B, Rudolf F. openBIS ELN-LIMS: an open-source database for academic laboratories. *Bioinformatics.* 2016;**32**:638–40.
- 42 Tremouilhac P, Nguyen A, Huang Y-C, Kotov S, Lütjohann DS, Hübsch F, et al. Chemotion ELN: an Open Source electronic lab notebook for chemists in academia. *J Cheminform.* 2017;**9**:54.
- 43 Bär H, Hochstrasser R, Papenfuß B. SiLA: basic standards for rapid integration in laboratory automation. *J Lab Autom.* 2012;**17**:86–95.
- 44 Christensen CD, Hofmeyr JHS, Rohwer JM. PySCeSToolbox: a collection of metabolic pathway analysis tools. *Bioinformatics.* 2018;**34**:124–5.

Supporting information

Additional supporting information may be found online in the Supporting Information section at the end of the article.

Table S1. SBO-terms in EnzymeML.

Fig. S1. Structure of an EnzymeML document.

Fig. S2. Example EnzymeML conditions annotation.

Fig. S3. Example EnzymeML annotation as described by the List Of Reactions tag. Data structure of EnzymeML to handle the CSV files.

Fig. S4. Example MIRIAM RDF format of an annotated unit definition.

Fig. S5. Creating EnzymeML documents from structured spreadsheets.

Fig. S6. Creating EnzymeML documents from STRENDA DB entries.

Fig. S7. Upload of EnzymeML data to SABIO-RK.

Fig. S8. Upload of the EnzymeML document 3IZNOK_TEST via the SABIO-RK data input interface, with data describing the compounds and the reaction (A) and the kinetic parameters (B).

Fig. S9. Editing of EnzymeML and simulation of time course data from kinetic parameters followed by kinetic modelling of EnzymeML data by COPASI.