Machine Learning & Software Development for Sustainable Chemistry

Joseph C. Davies

Thesis submitted to the University of Nottingham for the degree of Doctor of Philosophy

January 2025

Abstract

Sustainability represents one of the most pressing challenges for chemical synthesis in the 21st century. Traditional methods often rely on non-sustainable practices, including the use of chemicals that are harmful to human health and the environment. Significant efforts have been made to improve the sustainability of chemical synthesis and finding greener synthetic routes is a common aim for researchers.

Digitalisation presents an opportunity to embed intelligent tools into the workflows of chemists. Many academic researchers continue to use paper lab notebooks, highlighting the need for accessible electronic laboratory notebooks (ELNs) tailored to their needs. Machine learning can be used to create predictive models from high-quality data, offering a powerful approach to enhancing these tools.

In this thesis, software tools for sustainable chemistry are explored, and machine learning theory and its application to chemistry is introduced and exemplified. The development of the AI4Green ELN and the integration of machine learning models with an accompanying sustainability assessment is described. Integrating software and machine learning tools for sustainable chemistry directly into the ELN can help chemists measure and improve their sustainability without requiring duplicated data entry. The ELN captures reaction data in a structured, machine-readable format, facilitating the development of additional tools and modernising research data management.

Publications

1. **Joseph C. Davies**, David Pattison, Jonathan D. Hirst, Machine learning for yield prediction for chemical reactions using in situ sensors, *J. Mol. Graph. Model.*, **2023**, *118*, 108356
2. Samuel Boobier, **Joseph C. Davies**, Ivan N. Derbenev, Christopher M. Handley, Jonathan D. Hirst, AI4Green: an open-source ELN for green and sustainable chemistry, *J. Chem. Inf. Model.*, **2023**, *63* (10), 2895-2901
3. **Joseph C. Davies** and Jonathan D. Hirst, in *Reference Module in Chemistry, Molecular Sciences and Chemical Engineering*, **2024**. Ed. Béla Török, ISBN 9780124095472.
4. Ton M. Blackshaw, **Joseph C. Davies**, Kristian T. Spoerer, Jonathan D. Hirst, Enhancing Monte Carlo Tree Search for Retrosynthesis, *J. Chem. Inf. Model.*, **2025**, *ASAP*

Acknowledgements

First, I would like to thank my supervisor Jonathan Hirst for his constant support and for giving me this opportunity in the first place. You have given me excellent opportunities and always trusted me and encouraged me to push myself. I appreciate you always making time to support when needed and overall being a great person to work with professionally and personally. I have learnt a lot, and I am very grateful to you for that. You have also consistently and very patiently reminded me that these data are plural.

I am very thankful to everyone involved with the AI4Green project, all the stakeholders and everyone who has taken time to give feedback and engage with the project, but especially the project team. Thanks to Christopher Handley, Ivan Derbenev, and Sam Boobier who were all incredibly patient and taught me a lot when I was new to computational work. Similarly, thanks to many people in the DRS, particularly Phil Van Krimpen and Andy Rae who also helped and taught me a lot. A big thanks to the current team as well: Joe Heeley, Peace Nwafor, Zak Siddiq, Maddy Parker, Ton Blackshaw, and George Marshall. I have enjoyed working with all of you. Joe Heeley, we have worked together a lot since you joined the team and have proven Joes are stronger together. Thanks to DeepMatter for their provision of data and David Pattison for his guidance and data science expertise.

Thanks to everyone in the Hirst group and the computational chemistry department. I have enjoyed the occasional cryptic crossword, redactle and debates around the big issues in life, such as fighting swans and badgers. Appreciation also for Max Winslow and Connor Williamson in the office. I have thoroughly enjoyed our conversations,

especially all our football talk. Thanks to Copper for being my number one fan and an extra big special thanks to Ella for her constant encouragement, support, and faith in me. Finally, a thanks to my family for their support. A special mention to my Grandma, who once told me I should become a doctor (though she meant the other kind) shortly before she passed away. And to my Grandad, who passed away in May 2025, and whose quiet encouragement meant a lot.

Contents

Abstract i

Publications ii

Acknowledgements iii

List of Figures xiii

List of Tables xviii

List of Abbreviations xx

Chapter 1: Software Tools for Sustainable Chemistry 1

Abstract 1

1.1 Introduction 2

1.1.1 Green and Sustainable Chemistry 2

1.1.1.1 Green Metrics 4

1.1.2 Digitalisation of Chemistry 6

1.1.3 Data Standards 7

1.1.4 Industry 4.0 7

1.1.5 Software Availability 8

1.2 Software Tools 9

1.2.1 Sustainability Assessment 9

1.2.1.1 Synthesis 10

1.2.1.2 Analytical 13

1.2.1.3 Life Cycle Analysis 13

1.2.1.4 Integration 14

1.2.2 Benign By Design 15

1.2.2.1 Solvent Selection 15

1.2.2.1.1 Substitution Guides 16

1.2.2.1.2 Solvent Properties and their Prediction 17

1.2.2.1.3 Solvent Mapping 21

1.2.2.2 Computer-Aided Organic Synthesis 24

1.2.2.2.1 Retrosynthetic Analysis 24

1.2.2.2.2 Condition Prediction 26

1.2.2.2.3 Yield Prediction 27

1.2.2.2.4 Other Tools 28

1.3 In Silico Methods as a Sustainable Alternative 28

1.4 Future Perspectives 30

1.5 Conclusions 32

1.6 Scope and Objectives of Thesis 33

Chapter 2: Machine Learning for Chemistry – Methods & Theory 35

Abstract 35

2.1 Introduction to Artificial Intelligence and Machine Learning 36

2.2 Implementing Machine Learning 38

2.2.1 Data Collection 38

2.2.2 Data Preprocessing 39

2.2.3 Model and Feature Selection 40

2.3 Machine Learning Methods 42

2.3.1 Supervised Learning 43

2.3.1.1 Linear Regression 44

2.3.1.2 Polynomial Regression 45

2.3.1.3 Random Forest 45

2.3.1.4 Gradient Boosting Regression 46

2.3.1.5 Neural Networks 47

2.3.1.6 Long Short-Term Memory Neural Networks 48

2.3.2 Preprocessing Methods 49

2.3.2.1 Feature Scaling 49

2.3.2.2 Yeo-Johnson Transformation 49

2.3.2.3 Synthetic Minority Over-Sampling Technique for Regression with Gaussian Noise 50

2.3.3 Unsupervised Learning 50

2.3.3.1 K-means 51

2.3.3.2 Principal Component Analysis 51

2.3.3.3 Uniform Manifold Approximation and Projection 52

2.4 Chemistry Representations....53
2.4.1 Molecular Representations....53
2.4.2 Reaction Representations....56
2.4.2.1 Introduction to Reaction Representations....56
2.4.2.2 Record Keeping....58
2.4.2.3 Machine Readable Formats....59
2.4.2.4 Structured Data Repositories....61
Chapter 3: Machine Learning for Yield Prediction for Chemical Reactions Using *in situ* Sensors....63
Abstract....63
Collaboration Acknowledgements....64
3.1 Introduction....64
3.2 Methods....70
3.2.1 Data....70
3.2.2 Models....73
3.3 Results & Discussion....76
3.3.1 Data Analysis....76
3.3.2 Experiment Design....78
3.3.3 Use Cases....85
3.3.3.1 Use Case 1....85

3.3.3.2 Use Case 2 86

3.3.3.3 Use Case 3 87

3.4 Conclusion 92

Chapter 4: Development of the AI4Green Electronic Laboratory Notebook for Green and Sustainable Chemistry 94

Abstract 94

4.1 Introduction 96

4.1.1 Project Outline and Motivations 96

4.1.2 Digital Tool Development 97

4.1.3 The opportunities, the barriers, and the paper lab notebooks. 99

4.2 Methods 103

Team Acknowledgement 103

4.2.1 Application Stack 105

4.2.1.1 Backend 105

4.2.1.2 Frontend 109

4.2.1.3 Other Technologies 112

4.2.1.4 Tests 115

4.3 Functionality 116

4.3.1 Database 116

4.3.1.1 Database Seeding 116

4.3.1.2 Storing User Data 119
4.3.2 Workgroups & Workbooks 120
4.3.3 Reactions 122
4.3.4 Data Export 125
4.3.4.1 Human and Machine-readable Formats 126
4.3.4.2 Machine-Readable Formats 127
4.4 Discussion 128
4.4.1 Overview 128
4.4.2 Digital Research Environment Assessment 129
4.4.2.1 Generic Features 129
4.4.2.2 Notebooking features 131
4.4.2.3 Data Features 132
4.4.2.4 Publishing & Sharing Features 135
4.4.2.5 Collaboration & Management Features 136
4.4.2.6 Domain-based Features 138
4.4.2.7 Coding Support 139
4.4.2.8 Metadata, semantics, & AI 139
4.4.2.9 Searching 140
4.4.2.10 Customisation & Extension 141
4.4.2.11 Training & User Support 142

4.4.2.12 Overview 142
4.4.3 User Base 146
4.5 Conclusion 148
Chapter 5: Integration of a Retrosynthesis Tool with Sustainability Metrics into the AI4Green Electronic Laboratory Notebook 150
Abstract 150
5.1 Introduction 152
5.1.1 Retrosynthesis 152
5.1.1.1 Symbolic AI Approaches 152
5.1.1.2 Machine Learning Approaches 153
5.1.1.2.1 Template-free Approaches 153
5.1.1.2.2 Template-based Approaches 154
5.1.1.3 Data 154
5.1.1.4 Single-Step and Route Planners 157
5.1.1.5 Algorithm 158
5.1.1.6 Evaluation 160
5.1.2 Condition Prediction 161
5.1.3 Sustainability in CASP 162
5.1.4 This Work 163
5.2 Methods and Implementation 165

5.3 Results & Discussion 168

5.3.1 Case Study 1: Sildenafil Citrate 169

5.3.2 Case Study 2: Ibuprofen 172

5.4 Conclusions 173

Chapter 6: Concluding Remarks 175

Bibliography 178

Appendix 188

Chapter 3: Machine Learning for Yield Prediction for Chemical Reactions Using in situ Sensors 188

Chapter 4: Development of the AI4Green Electronic Laboratory Notebook for Green and Sustainable Chemistry 198

Chapter 5: Integration of a Retrosynthesis Tool with Sustainability Metrics into the AI4Green Electronic Laboratory Notebook 201

List of Figures

Figure 1.1: A: Example pictogram generated by ComplexGAPI. B: Explains meaning of hexagon segments. From ref. [19] with permission from the Royal Society of Chemistry.[19] 13

Figure 1.2: Composition of process components by percentage mass used to manufacture an active pharmaceutical ingredient. Adapted with permission from C. Jimenez-Gonzalez, C. S. Ponder, Q. B. Broxterman and J. B. Manley, *Org. Process Res. Dev.*, 2011, **15**, 912–917. Copyright 2024 American Chemical Society.[10] 16

Figure 1.3: The ACS GCIPR Pharmaceutical Roundtable solvent selection tool. Solvents which are near each other have similar chemical and physical properties and distant solvents have different properties. The tool performs PCA to provide a two-dimensional mapping of the solvents and provides multiple filters to aid searching. One solvent of the 272 has been selected here. 22

Figure 2.1: The steps involved in developing a machine learning model.....41

Figure 2.2: Simplified schematics of two neural network architectures. **A)** In a standard feed-forward neural network, data flows in one direction: from the input layer, through the hidden layers, to the output layer. Each layer applies a transformation to the data. **B)** In a recurrent neural network, the hidden layer includes a recurrent connection that feeds the previous hidden state back into the hidden layer. This feedback loop allows the network to maintain context when processing sequential data. 49

Figure 2.3: Reaction data hierarchy, where the base represents the simplest form of reaction data, and the top represents the most comprehensive. **Reactants and products** form the base as the most basic data that can be included. **Conditions**

include data such as reagents, solvents, catalysts, temperature, atmospheres, and other components or methods influencing the reaction. The **procedure** includes the specific order of steps, how they were carried out and on what scale. This can indicate important aspects for reproducibility, such as whether the reaction was done in batch or flow and how the product was purified. **Comprehensive Description** providing additional detail to describe the dataset, including metadata, such as who performed it and when, and more comprehensive data to fully describe the experiment, for example, by including spectroscopic analysis. 61

Figure 3.1: The DeviceX™ is a probe which is immersed in the reaction mixture. A) A schematic of this probe is shown. B) An example of temperature measurements for two discrete runs shown over a time-course. These runs are examples and not the experimental data used in this work. 71

Figure 3.2: Buchwald reaction between 4-chlorotoluene and benzophenone hydrazone with a palladium acetate catalyst, MePhos ligand, sodium hydroxide base and either an isopropyl alcohol or tert-amyl alcohol solvent. 72

Figure 3.3: The left panel shows the Pearson correlation values for the unmodified sensor data. The right panel shows the Pearson correlation values for cumulative sensor values, where c_features refers to cumulative features. 77

Figure 3.4: Variation over time of the recorded colour pixel component in the reaction mixture for example run Crocus-020. The top panel shows the calculated colour from the combined RGB values, and the bottom panel shows the individual RGB values. Each colour represents the colour shown, the crosses are red, diamonds green and downward arrows blue. Data are shown without error bars for clarity. 78

Figure 3.5: The left panel is the distribution of green cumulative values, and the right panel is the distribution after applying a Yeo-Johnson transformation. A normal distribution is plotted over both histograms. 80
Figure 3.6: Comparison of the mean absolute errors (indicated by the colour bar) for the different combinations of features (x axis) and models (y axis) obtained from the 10-fold cross-validation using a 9:1 train: test split. YJ stands for Yeo-Johnson. The polynomial regression model developed was not suitable for all feature sets, and hence was only used in conjunction with green cumulative. 82
Figure 3.7: The mean absolute error for product formation predictions from an LSTM model predicting 0, 30, 60 or 120 minutes ahead using a 10-fold cross-validation with a 9:1 train: test split. 85
Figure 3.8: Blue circles show mean error in predictions for reaction 17 and black crosses for reaction 25. The standard error bars are calculated from ten repeats. ...87
Figure 3.9: The blue line is the mean absolute error for predictions of a model trained only on runs which were chronologically before the test run, the number of which is shown on the bottom x axis. The black line is the mean absolute error for predictions from the cross-validation where a model was developed to predict the run by training on *all* other runs, regardless of chronological ordering. 89
Figure 3.10: The relationship between the number of runs in the training data and the accuracy of predictions for product formation in run 25. 90
Figure 3.11: *k*=3 clustering of a UMAP projection of the reaction data for all runs used in machine learning models. 91
Figure 4.1: AI4green's workgroup interface. Workgroup management options are shown on the left. In the middle, the current workbook is shown in the dropdown list

with the new reaction button below. On the right-hand side, the workbook's reactions are listed, with the option to export data shown below. 111

Figure 4.2: Simplified schematic of the AI4Green application architecture. External data sources, **PubChem** and **CHEM21** are used to populate the database**.** The Azure-hosted PostgreSQL **database** is connected to the backend through a connection string, and queries are made from the **backend** to the database using SQLAlchemy. The backend handles the application logic and is written in Python. The **Flask** web framework supports data exchange between the frontend and backend. The **frontend** is the interface the user views and interacts with and is made using HTML, CSS, and JavaScript. 111

Figure 4.3: This is the user interface of the quick access section of the home page. The user selects a workgroup, which shows a list of workbooks that belong to it. The user then selects a workbook and is shown a list of reactions belonging to that workbook. This view provides a simple method for displaying the hierarchical nature of the data. 120

Figure 4.4: Example sustainability assessment from the AI4Green interface. Metrics are colour-coded by their sustainability. 125

Figure 4.5: The proposed pipeline for exporting data to an external repository, such as the open reaction database. **1)** The user must enter reaction data during experiment creation, and this step is already present within the ELN. **2)** The export data request process also already exists within the ELN for the user to receive data in a format of their choice, and approval is required from all principal investigators within a workgroup. **3)** The data must then be converted to the repository format. No such

mechanism yet exists within the ELN. **4)** The final step is sending this data to the repository using an API. 145

Figure 4.6: Number of new users registered in each quarter. 146

Figure 5.1: The distribution of reaction classes of the USPTO dataset, omitting the 46% of reactions that were not classified. Taken from Schneider et al.[163] 157

Figure 5.2: The four steps of a Monte Carlo Tree Search. **Selection**: a node is selected using the UCT equation to balance exploration and exploitation. **Expansion**: the selected node is expanded, adding to the tree search. **Simulation**: the expanded node is simulated until a terminal position is reached, and this is assessed. For retrosynthesis, a successful simulation results in molecules belonging to the stock file. **Backpropagation**: the scores of the preceding nodes are updated positively or negatively, depending on the outcome of the simulation. Selection occurs again with the now updated node scores. 160

Figure 5.3: Schematic for the architecture showing the data exchange between the AI4Green web application and the APIs of the conditions and retrosynthesis microservices. 166

Figure 5.4: Predicted retrosynthetic route for sildenafil citrate. 169

Figure 5.5: Predicted sulfonylation conditions in the synthesis of sildenafil citrate. 170

Figure 5.6: The unoptimised and optimised medicinal chemistry routes used in the synthesis of Sildenafil Citrate as reported in the literature.[175] 171

Figure 5.7: Comparison of initial Boots and BHC routes for the synthesis of ibuprofen. 172

List of Tables

Table 1.1: The 12 principles of green chemistry developed in 1996 by Anastas et al.[1] 2

Table 1.2: Green chemistry metrics and methods for quantifying greenness and sustainability. 6

Table 1.3: The different pass tiers used in the CHEM21 toolkit to holistically assess reactions. 12

Table 1.4: Selection of exemplar software tools for sustainability assessment.....14

Table 1.5: Properties of solvents found in the solvent selection software SUSSOL.[25] 18

Table 1.6: Kamlet-Abboud-Taft and Hansen solvent parameters and their components with the commonly used symbol and corresponding definitions.[27] 20

Table 1.7: Selection of exemplar solvent selection tools. 23

Table 1.8: Selection of exemplar software tools for computer-aided synthesis design 26

Table 2.1: Different representations and identifiers of Indole.....56

Table 3.1: The chosen LSTM hyperparameters. 75

Table 4.1: Properties seeded into the compound database. 118

Table 4.2: Generic features required for a digital research environment.[132] 130

Table 4.3: Notebooking features required for a digital research environment.[132] ...132

Table 4.4: Data features required for a digital research environment.[132] 133

Table 4.5: Publishing and sharing features required for a digital research environment.[132] 136

Table 4.6: Collaboration and management features required for a digital research environment.[132] 137

Table 4.7: Domain-based features required for a digital research environment.[132] 139

Table 4.8: Coding support features required for a digital research environment.[132]139

Table 4.9: Metadata, semantics, and AI support features required for a digital research environment.[132] 140

Table 4.10: Search features required for a digital research environment.[132] 141

Table 4.11: Customisation features required for a digital research environment.[132] 142

Table 4.12: Training and user support features required for a digital research environment.[132] 142

Table 5.1: Selection of large databases recording chemical reactions. The table is taken from Weber et al.[12] 155

List of Abbreviations

ACS GCIPR	American Chemical Society Green Chemistry Institute's Pharmaceutical Roundtable
AI	Artificial Intelligence
AJAX	Asynchronous JavaScript XML
ANOVA	Analysis of Variance
API	Application Programming Interface
ASKCOS	Automated System for Knowledge-based Continuous Organic Synthesis
AWS	Amazon Web Services
BHC	Boots-Hoechst-Celanese
CAMPD	Computer-aided molecular and process design
CAS	Chemical Abstracts Service
CASP	Computer-Aided Synthesis Planning
CHEM21	Chemical Manufacturing Methods for the 21st Century Pharmaceutical Industries
CI/CD	Continuous Integration/Continuous Deployment
CID	PubChem Compound Identifier
CIR	Chemical Identity Resolver
COSMO	Conductor-like Screening Model for Realistic Solvation
CRO	Contract Research Organisation
CSS	Cascading Style Sheets
CSV	Comma-Separated Values
CT File	Chemical Table File
DENDRAL	Dendritic Algorithm
DMF	Dimethylformamide
DOI	Digital Object Identifier
DRE	Digital Research Environment
DRS	Digital Research Service
ECHA	European Chemicals Agency
ELN	Electronic Laboratory Notebook

EQ	Environment Quotient
FAIR	Findable, Accessible, Interoperable, and Reusable
GAPI	Green Analytical Procedure Index
GBR	Gradient Boosting Regression
GHS	Globally Harmonised System of Hazards
GNU	GNU's Not Unix
GPL	General Public License
GPU	Graphical Processing Unit
GUI	Graphical User Interface
HPLC	High-Performance Liquid Chromatography
HSP	Hansen Solubility Parameters
HTE	High-Throughput Experimentation
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
IP	Intellectual Property
IPA	Isopropyl Alcohol
IR	Infrared
IUPAC	International Union of Pure and Applied Chemistry
JSON	JavaScript Object Notation
JSON-LD	JavaScript Object Notation for Linked Data
KAT	Kamlet-Abboud-Taft
LCA	Life-Cycle Analysis
LC-MS	Liquid Chromatography- Mass Spectrometry
LCSS	Laboratory Chemical Safety Sheets
LHASA	Logic and Heuristics Applied to Synthetic Analysis
LIME	Local Interpretable Model-Agnostic Explanations
LIMS	Laboratory Information Management System
LLM	Large-Language Model
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error

MCTS	Monte Carlo Tree-Search
MDL	Molecular Design Limited
MLPDS	Machine Learning for Pharmaceutical Discovery and Synthesis Consortium
MSE	Mean Squared Error
NMP	N-Methyl-2-Pyrrolidone
NMR	Nuclear Magnetic Resonance
OECD	Organisation for Economic Co-operation and Development
OPRD	Organic Process and Research Development
ORD	Open Reaction Database
ORM	Object-Relational Mapping
PARIS III	Program for Assisting the Replacement of Industrial Solvents III
PCA	Principal Component Analysis
PMI	Process-Mass Intensity
PPE	Personal Protective Equipment
PSDI	Physical Sciences Data Infrastructure
QR Code	Quick Response Code
QSAR	Quantitative Structure-Activity Relationship
RDBMS	Relational Database Management System
RDF	Reaction Data File
RFR	Random Forest Regression
RGB	Red, Green, and Blue
RNN	Recurrent Neural Network
RO-Crate	Research Object Crate
RXN	Reaction File
SDF	Structure-Data File
SE	Standard Error
SELFIES	Self-Referencing Embedded Strings
SHAP	Shapley Additive Explanation
SHE	Safety, Health, and Environment
SI	Supplementary Information

SMILES	Simplified Molecular Input Line Entry System
SMOGN	Synthetic Minority Over-Sampling Technique for Regression with Gaussian Noise
t-SNE	t-Distributed Stochastic Neighbour Embedding
SQL	Structured Query Language
SURF	Simple, User-Friendly Reaction Format
SUSSOL	Sustainable Solvents Selection and Substitution Software
TAA	Tert-Amyl Alcohol
TIOBE	The Importance of Being Earnest
UCT	Upper Confidence Bounds Applied to Trees
UDM	Unified Data Model
UMAP	Uniform Manifold Approximation and Projection
USPTO	United States Patent and Trademark Office
UUID	Universally Unique Identifier
UV	Ultraviolet
UV-Vis	Ultraviolet-Visible
WLN	Wiswesser Line Notation
XML	Extensible Markup Language
YJ	Yeo-Johnson

Chapter 1: Software Tools for Sustainable Chemistry

Abstract

In this chapter, we examine the role of software with focusses on software for synthetic and green and sustainable chemistry. The concepts of green and sustainable chemistry are introduced, and the metrics used in their analysis are discussed. Making chemistry greener and more sustainable is a growing priority for researchers and software tools have been developed to aid in this pursuit. Software tools for green and sustainable chemistry have been developed to assess existing methods, propose new ones, and replace some experimental methods altogether with *in silico* approaches. We discuss the digitalisation of chemistry and the computational advances that enable software tools to play a growing role in all aspects of chemical research. Barriers and limitations of current tools are discussed along with future trajectories.

1.1 Introduction

1.1.1 Green and Sustainable Chemistry

Green chemistry reduces or eliminates the environmental impact of a chemical product or process. The concept is grounded in the 12 principles of green chemistry (Table 1.1), which were published in 1998.[1] Prevention, the first principle particularly summarises the green chemistry philosophy. This is made apparent when compared to remediation approaches which try and clean up hazardous chemicals that are released during a process. Remediation is its own distinct field with processes aimed at treating chemical waste. In contrast, green chemistry can be applied to all areas of chemistry. For any product or process, it is possible to analyse their impacts and compare them to researched alternatives.

Table 1.1: The 12 principles of green chemistry developed in 1996 by Anastas et al.[1]

Principle
Prevention
Atom Economy
Less Hazardous Chemical Syntheses
Designing Safer Chemicals
Safer Solvents and Auxiliaries
Design for Energy Efficiency
Use of Renewable Feedstocks
Reduce Derivatives
Catalysis
Design for Degradation
Real-time analysis for Pollution Prevention
Inherently Safer Chemistry for Accident Prevention

The 12 principles give rise to a qualitative understanding of green chemistry. In practice, particularly when using software approaches, it can be beneficial to attempt some quantification to create “green metrics”. Not all these principles can be quantified with the same level of ease. For example, atom economy can be

straightforwardly calculated as a numeric metric from an equation, but it is more challenging to quantify the level of hazard associated with a chemical synthesis.

Sustainable chemistry is less well-defined than green chemistry and this in part leads to a lack of standardisation of measuring the sustainability of a process. A broader definition of sustainability can be found from the 2030 Agenda for Sustainable Development from the United Nations, which balances the three pillars of sustainable development – economic, social, and environmental.[2] The concept of sustainability being dependent on three components is also reflected in the triple bottom line assessment, which evaluates an organisation based on its impact on people, the planet, and profit.[3]

Sustainability can be challenging to measure, as it is a multifactorial problem. Green metrics can aid the assessment of the impact of environmental harm. Economic cost can be measured through the materials purchased, but overheads, such as energy and labour, can be more challenging to measure, particularly for an individual researcher. The social impact is also hard to measure for an individual researcher.

Interest in sustainability has grown in recent years and it is now a priority for chemical manufacturers. This has happened alongside a better understanding of climate change and is a trend that is likely to accelerate as regulatory deadlines approach, such as the UK's aim to achieve net zero emissions by 2050.[4] Another motivating factor is the potential cost savings. The annual production of active pharmaceutical ingredients (APIs) generates 10 million metric tons of waste and $20 billion in disposal costs.[5]

Computers have been used in scientific research since the 1930s by scientists such as Douglas Hartree to solve quantum mechanics equations, and their use has since

undergone a radical transformation. Chemistry databases were developed and enabled the sharing of data in a way that was previously not possible. More specialist chemistry software was developed in the 1970s. One example is the Logic and Heuristics Applied to Synthetic Analysis (LHASA) program which was developed for retrosynthetic analysis.[6] Computational methods became integrated into the workflows of other branches of chemistry. ChemDraw, released in 1986, allowed chemists to create publication-quality structures more easily than manual drawing with templates.[7] It is has since become extensively used and, its integration within the ubiquitous Microsoft Office software reflects this.

As interest in green and sustainable chemistry has grown in recent decades, software has been developed to support researchers to meet their ambitions in this domain.[8] Computational power and researchers' access to computers and internet connectivity has grown exponentially over time. These conditions have enabled the development and use of increasingly powerful and accessible software tools, capable of supporting cutting-edge research and industrial processes. In the following discussion, our emphasis is on software capable of supporting green and sustainable chemistry.

1.1.1.1 Green Metrics

Making a chemical process sustainable is challenging, as many considerations must be made simultaneously. The 12 principles of green chemistry encourage thinking around the environmental and safety impact of all aspects of a process. Metrics to quantify aspects of sustainability have continuously been developed. This enables the comparison of processes through metrics which can help decision-making and provides a rational framework for selecting processes by sustainability. There is no

universally agreed set of metrics for measuring the sustainability of a process. Mass-based metrics, atom economy and the E factor were invented in the early 1990s.[9] In the following years, these metrics have been widely adopted and modified for different use-cases. Additional mass-based metrics have since been developed. Process mass intensity (PMI), for example, was promoted as the key metric for sustainable manufacturing by the American Chemical Society Green Chemistry Institute's Pharmaceutical Roundtable (ACS GCIPR) in 2011.[10] It has been pointed out that mass-based metrics alone are insufficient for providing a holistic assessment of environmental harms, as they do not differentiate based on the composition of waste, only the mass.[9] The toxicity and other implications that arise from the specific masses used and generated by process must also be considered. An environmental quotient (EQ) was introduced in 1994 to overcome this limitation by multiplying the E factor by an arbitrarily assigned "unfriendliness" quotient.[9] Since then, many more metrics and frameworks have been developed. Life cycle analysis (LCA) is a framework developed in the 1990s which assesses the environmental impacts of a process using a set of impact factors within a designated scope. Table 1.2 summarises the green chemistry metrics and frameworks discussed in this section.

Table 1.2: Green chemistry metrics and methods for quantifying greenness and sustainability.

Metric	Equation/Description
E Factor[9]	$\text{E Factor} = \frac{Mass\ of\ waste\ (kg)}{\text{Mass of final product (kg)}}$
Process mass intensity[10]	$\text{Mass Intensity} = \frac{Mass\ of\ raw\ materials\ (kg)}{\text{Mass of final product (kg)}}$
Environmental quotient[9]	$\text{Environmental Quotient} = \text{E factor} \times \text{Q}$ $(\text{Q} = \text{"unfriendliness" factor})$
Life cycle analysis	Framework for holistically assessing the environmental impacts of a process.
Social life cycle analysis[11]	Framework for holistically assessing the sustainability of a process.

Variation in metrics arises due to differences in preferences and priorities at an individual and institutional level. One example of a reason for differing priorities is in the manufacturing of APIs. Organic solvent usage accounts for 56% of the mass of a process.[10] This would indicate solvents have a disproportionate effect on the sustainability of a process in this field. Therefore, in these processes sustainable solvent selection may be of greater priority.

1.1.2 Digitalisation of Chemistry

Chemistry has become increasingly digitalised in recent years in a trend that is expected to continue. Resources including literature, searchable databases, and a plethora of web-based or downloadable software tools are all accessed digitally. In industry, use of electronic laboratory notebooks (ELNs) or a laboratory inventory management system (LIMS) is common for scientists. Many companies utilise their

intranet as a facile way to disseminate information through files accessible to all employees. Digitalisation has many potential benefits including aiding the transition to more sustainable chemistry.[12]

1.1.3 Data Standards

Key to the role of digitalisation in this area is the effective collection, handling, and analysis of chemistry data. There are calls for the data generated by research to be made FAIR (Findable Accessible Interoperable and Reusable), and, when reasonable, for this to be mandatory as part of the publishing and funding process.[13] Software tools, such as ELNs and LIMS, can help facilitate this data capture. Crucially, these systems capture negative results which are often unpublished. A 2017 survey of chemists in academia reported 76% of respondents had a strong interest in either implementing or finding out about them more but only 11% currently used an ELN in their research group.[14] These data highlight an opportunity to increase ELN usage among chemists in academia. Widespread adoption of methods for disseminating experiment data could help facilitate the development of new software tools for sustainable chemistry.

1.1.4 Industry 4.0

Industry 4.0 is the concept of manufacturers integrating modern technologies into their existing workflows. Software is integral to these recent technologies which include the Internet of Things (IoT), analytics, and AI. These innovative technologies can lead to improvements in efficiency, which may improve process sustainability by reducing emissions and energy usage. The IoT is a network of things that have integrated digital devices such as sensors which can communicate and exchange data

with other integrated digital devices or computer systems. Pipelines and automatic data cleaning processes can enable the collection of these data that may previously have been uncollected and monitored with a standalone device, such as standard thermometer. The importance of data has already been discussed. For example, in pilot plants software has been used to train machine learning models on past data to predict when to perform preventative maintenance negating unexpected downtime or premature maintenance.[15]

1.1.5 Software Availability

Software availability refers to the ease with which users can access it and is contingent on the chosen distribution model. The distribution model covers how the software is made available. This includes factors such as whether it is open-source, commercial, restricted to internal use within an organisation, and what delivery model is used.

Open-source refers to making the source code available for anyone to see and the opposite is the case for closed-source. Open-source code is typically stored in an online repository, GitHub being the most used. Multiple open-source licenses have been developed, such as the GNU (which stands for GNU's Not Unix) General Public License (GPL), to define the exact terms with which the software can be reused.

Software delivery models describe where the software is installed and how one would access it. Two common examples of delivery models include on-premises and cloud-based. On-premises requires the software be installed on the user's machine to be used. In contrast, cloud-based software is hosted online, often by a third-party provider, and is accessible over the internet, commonly through a web browser.

Commercial software typically requires a monetary payment for usage and is

accompanied by a type of commercial software license. This license often relates to the payment model used. For example, software available for a one-time purchase may come with a Perpetual License - allowing indefinite use after the initial payment. However, software which has a subscription model does not grant indefinite usage and requires continued payments. Tiered approaches are common, with tiers often linked to constraints on the number of simultaneous instances within the client organisation or varied charges based on factors like institution type. This may include offering the software at a lower price or for free to educational institutions.

Software may be restricted to use internally within one or more organisations for various reasons. These include maintaining a strategic advantage and confidentiality of intellectual property (IP). Alternatively, the software may be unsuitable for external use for reasons such as being overly customised and integrated with existing systems. All these factors have an impact on the reach of the software and its lifecycle.

1.2 Software Tools

There are many ways one can look at software tools for green and sustainable chemistry due to the high level of interest in both areas and their overlap and integration into all aspects of chemistry.

1.2.1 Sustainability Assessment

A basic example is where the chemist inputs particular details of their work and the software returns feedback on sustainability. This type of software will often perform calculations on the inputs to calculate specified metrics and provides a standardised way for chemists to compare processes and make a more informed decision. Commonalities across many of these tools include a foundation in the 12 principles of

green chemistry or variations tailored for specific domains like analytical chemistry or chemical engineering. Many also make use of visually engaging methods of presenting the assessment which may be words, numerical, colour-coded, pictograms, plots, or a combination.

1.2.1.1 Synthesis

The software can either be designed to attempt to provide a holistic assessment of sustainability or calculate a single metric. Holistic assessments are advantageous because sustainability is a multifaceted problem, but not all metrics may be relevant for a particular procedure. However, using a single metric may be misleading and using multiple software for single metrics can require repeated data entry.

One way to make assessments and provide holistic feedback is to explicitly score the safety, health, and environmental aspects.[16] This integrates with the well-established Globally Harmonised System of Classification and Labelling of Chemicals (GHS) hazard codes used to standardise phrases about the hazards of substances where hazards are classed as physical, health, or environmental. Another consideration for a synthesis sustainability software tool is whether it measures the entire route or just a single reaction. Longer, non-convergent syntheses tend to have poorer efficiency; so by only evaluating a single reaction, important context may be lost.

Spreadsheet templates which calculate a specific series of green metrics have been developed for this purpose.[17] While not standalone software, these still merit discussion due to their similarity to other standalone tools and the low barrier to their development and use. This low barrier to development enables green chemists who are not proficient in software development to share their sustainability assessment

methods. Chemists will typically already have access to and be comfortable using spreadsheet software, negating the need to make an account, download software, or worry about where the data will be stored.

For this type of sustainability software there is also a trade-off between the level of detail in the feedback and the amount of input data required. Data input can be both time-consuming and impractical because not all chemical compounds, particularly synthetic intermediates, are fully characterised, nor is this characterisation always feasible. This trade-off is best determined by the level of detail required. For reactions which are to be performed repeatedly or on a large scale or for researchers for whom green chemistry is their primary focus, the extra detail in feedback can be worth the additional data input. For reactions which are to be done once on a small scale this is less likely to be the case and it is more feasible to use simpler metrics. CHEM21 uses a 0-3 tier system, shown in Table 1.3. This provides a pragmatic approach by still scrutinising the sustainability of each step but in a way which does not alienate chemists with unrealistic expectations. Zero tier reactions may only be performed once, so excessive scrutiny at this stage could be a poor use of the chemist's time and may discourage uptake. However, still applying some basic scrutiny avoids trying to scale up reactions that have serious issues which may have longer-term negative consequences such as necessitating extensive optimisation and radical route redesign. This idea of all chemists having a basic awareness of greenness from the offset is one which recurs throughout this work and can be seen in multiple tools and research in this area.[5] Their assessments use colour-coding and wording ranging from 'Recommended' to 'Highly Hazardous'.

Table 1.3: The different pass tiers used in the CHEM21 toolkit to holistically assess reactions.

Pass Tier	Description
Zero	For use in the discovery phase when screening large numbers of reaction on a small scale. Sustainability is assessed by health and safety from hazard codes, solvent choice, and mass-based metrics. Product formation is estimated from analytical methods to avoid the need for workup and purification. Suitable analytical methods depend on the specific molecules in the reaction, e.g., the ratio of high-performance liquid chromatography (HPLC) peak areas corresponding to starting material and product. By only requiring easily accessible data the additional researcher workload is minimised.
First	For use in the discovery phase for reactions that have passed the zero pass and are performed on the scale of hundreds of milligrams. Sustainability is assessed on the same metrics as the zero-pass, but the mass-based metrics are calculated from isolated product following workup and purification. PMI is added and includes all mass inputs including workup and purification. Additional metrics include element sustainability, workup procedure, batch/flow, temperature, catalyst/enzyme use, compound availability, and applicability of the reaction class. Most of these metrics are obtainable from data the chemist would enter in their experimental writeup.
Second	For use in reactions performed at pilot scale which produce industrially relevant compounds with commercial value and have perform well on the first pass metrics. The metrics used on the first pass are now applied more judiciously. Reduction and recovery of inputs should be considered at this scale to improve sustainability. The carbon footprint of the synthesis of the inputs and whether these are renewable (bio-based) or not (petrochemical). Before the scale up at this stage a safety assessment should also be completed. LCA is a suitable framework for this stage but may be too time-consuming due and necessary data may be unobtainable. Few processes will be performed on this scale and so require data beyond that which would be entered in a standard experimental writeup.
Third	For assessing the feasibility of performing the reaction at an industrial scale for manufacturing. Additional considerations on the waste generated, processing of this waste, and upstream and downstream energy use. Very few processes will be performed at this scale and intensive research into characterising the process is warranted.

Green Motion groups the 12 principles of green chemistry into seven fundamental concepts.[18] Each of these concepts is scored and an overall score can be obtained by

summing these to provide a quantitative method of comparison, whereas in the CHEM21 toolkit the authors deliberately avoid giving an overall numerical score to a reaction to encourage holistic thinking.[17]

1.2.1.2 Analytical

The complementary green analytical procedure index (ComplexGAPI) is a freely available open-source tool that was developed on the back of the green analytical procedure index (GAPI).[19] ComplexGAPI is designed to assess an entire analytical methodology including prior to the analysis itself. It generates a pictogram with a colour-scale as seen in Figure 1.1.

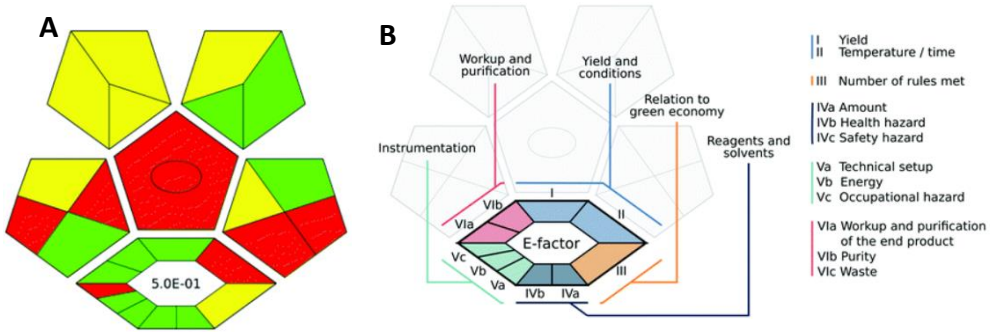


Figure 1.1: A: Example pictogram generated by ComplexGAPI. B: Explains meaning of hexagon segments. From ref. [19] with permission from the Royal Society of Chemistry.[19]

1.2.1.3 Life Cycle Analysis

The scope of LCA can span from "cradle to grave", encompassing the entire life cycle from raw material extraction to product end-of-life, or it can be confined to "gate to gate", focusing solely on the manufacturing process. The first provides a fuller picture of the impact of a process but requires significantly more data than the latter. For a chemical process to be entirely sustainable, economic and social impacts should also be considered and LCA can be adapted for this by using a social life cycle assessment (S-LCA) which also considers social and socio-economic aspects.[11]

Software is used throughout the LCA process. Curating a dataset of suitable quality can be a barrier when trying to apply LCA. Databases can provide high quality data but may be unattractive due to cost or stipulations preventing the training of machine learning models. Alternatively, simulation software can be used to generate process data. The challenges in obtaining suitable, high-quality datasets are compounded by the observation that different LCA software output different results. A recommendation to overcome this was for all software tools to update their databases from a Global LCA Data network.[20]

1.2.1.4 Integration

Integration of sustainability assessments into existing workflows provides the chemist with feedback with minimal additional overhead. One method of doing this is to integrate green chemistry into an ELN. Industrial solutions to this have been published as early as 2012.[21]

Exemplar tools discussed in this section are summarised in Table 1.4.

Table 1.4: Selection of exemplar software tools for sustainability assessment.

Software Tool	Description
CHEM21 Spreadsheet[17]	A spreadsheet template which takes user input to assess reaction sustainability using a traffic light colour coding system. https://learning.acsgcipr.org/guides-and-metrics/metrics/the-chem21-metrics-toolkit/
Green Motion[18]	Commercial software which outputs an overall score based on individual scores from each of the 12 principles of green chemistry. Originally developed to rate and enable comparison between the environmental impact of flavours and fragrances.
Complex GAPI[19]	Open-source tool used to assess the greenness of an analytical process. https://git.pg.edu.pl/p174235/complexgapi
OpenLCA	Open-source tool for life cycle analysis. https://www.openlca.org/

1.2.2 Benign By Design

Software tools can go beyond assessment and help chemists work more sustainably by suggesting alternative chemicals or entire procedures. 'Benign by Design' is a well-established idea that was conceptualised in the 1990s as part of the green chemistry paradigm shift in proactively preventing environmental harm rather than finding the best way of dealing with the consequences. This involves considering the synthesis and degradation and designing products so that these processes are as environmentally benign as possible. This approach shares a similarity with LCA as it focuses on the product's lifecycle, but it does not incorporate a quantitative framework.

1.2.2.1 Solvent Selection

Solvents are the chemicals used to dissolve a solute. A mixture of solvents may be used, and mixtures of solutes are likely to be used. Solvent purposes and their desired properties vary hugely, depending on application. For example, it is often intended for the solvent to be removed from the process after a particular duration. For a reaction in a laboratory the scientist can actively do this. The solvent may be removed by filtration or evaporated under reduced pressure and heating. However, for paint the solvent cannot be actively removed in such a way. The paint must have suitable properties to evaporate safely within an acceptable timeframe.

As shown in Figure 1.2, organic solvent usage makes up over 50% of the percentage mass of API synthesis.[10] This makes solvents significant contributors to the environmental impact of these processes. Solvent selection tools can assist the chemist in substituting non-sustainable solvents with more sustainable alternatives.

These can range in complexity from tools that suggest single replacements to tools which enable comparison between hundreds of solvents simultaneously.

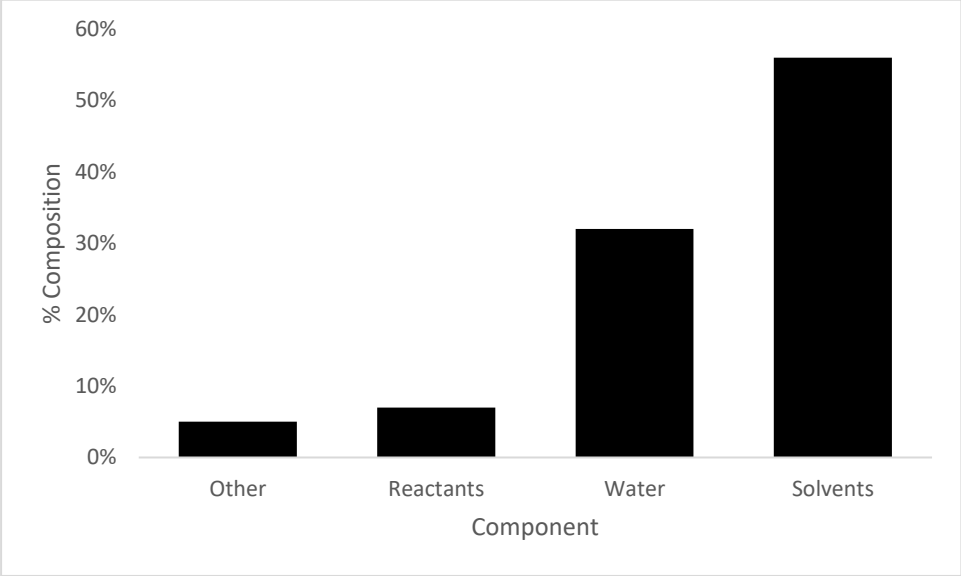


Figure 1.2: Composition of process components by percentage mass used to manufacture an active pharmaceutical ingredient. Adapted with permission from C. Jimenez-Gonzalez, C. S. Ponder, Q. B. Broxterman and J. B. Manley, *Org. Process Res. Dev.*, 2011, **15**, 912–917. Copyright 2024 American Chemical Society.[10]

At multiple organisations there have been considerable efforts to reduce usage of harmful solvents. A recent case study from Syngenta revealed they managed to halve their consumption of seven hazardous solvents over a two-year period without a loss in productivity.[22] A bias of precedence and status quo have been suggested as motivations for chemists to seek out solvents they know are reliable. High quality solvent selection tools can help facilitate overcoming these biases.[22]

1.2.2.1.1 Substitution Guides

Solvent selection guides are simple tools that help chemists select more sustainable solvents. They often use a traffic-light system which enables the chemist to immediately have an idea of the sustainability of a solvent. Since 1999, pharmaceutical companies have released solvent selection guides.[23] These tools are simple but

effective to use and typically include robust alternative solvent suggestions, utilising molecularly similar substitutions, such as replacing carcinogenic benzene with toluene.[16] To disseminate these guides within an organisation they have been shared as word processing or spreadsheet files within a company's intranet and displayed in labs as posters. Research comparing these different guides has found they generally agree with one another.[24] Limitation of these guides include chemists already being aware of these key substitutions and they lack flexibility if either no substitution is suggested or the suggested one fails. Additionally, failure to update these guides may result in recently developed green solvents not being present within the guide.

1.2.2.1.2 Solvent Properties and their Prediction

Solvent substitution requires finding green solvents with similar desirable properties to the currently used unsustainable solvent. Physical properties such as boiling point, density, or log P can be either measured or predicted. These properties are important to know when trying to rationally find a new solvent. Solvent properties try to provide a way to quantify the behavior of a solvent. These include physical properties like boiling and melting points, which must be known as typically the solvent must be a liquid at the desired temperature. Log P is not specific to solvent selection and is originally from drug discovery but has been used in solvent selection tools. This gives a measure of how hydrophilic or hydrophobic a compound is, and this information can help predict what type of compounds a solvent can dissolve. These properties and more are described in Table 1.5.

Table 1.5: Properties of solvents found in the solvent selection software SUSSOL.[25]

Property (unit)	Description and Utility
Boiling Point (°C)	The temperature at which the solvent turns from liquid into vapor. Without additional measures, solvent will evaporate and leave the system if this temperature is exceeded.
Melting Point (°C)	The temperature at which the solvent turns from liquid into solid. Exceptions withstanding, the solid will not uniformly dissolve the solute and mixing cannot occur.
Log P	The log of the ratio of solubilities partition coefficient for a compound in a mixture of two immiscible solvents (Normally water and octanol). A higher value is hydrophilic and lower is hydrophobic. This can help find 'like' solvents for compounds.
Vapor Pressure (Pa / mmHg)	Vapor pressure depends on temperature and is the pressure that the vapor that evaporates from the solvent exerts. This is particularly noticeable in a closed system.
Flash Point (°C)	The lowest temperature at which the vapor that evaporating from the solvent can be ignited. This is an important safety consideration.
Autoignition Temperature (°C)	The lowest temperature at which the solvent will combust spontaneously without an ignition source. This is an important safety consideration.
Viscosity (Pa.s)	The resistance to movement or changing shape, i.e., thickness. This is important as it affects solvent mixing and handling.
Density (kg/L)	The density of the solvent is the mass of a defined volume. Density can have many effects on a solvent's behaviour such as in layering in multiphasic solutions.
Molecular Weight (g/mol)	The molecular weight is the mass of a single mole of a compound. This can influence the ability to dissolve solutes.

Conductor-like Screening Model for Realistic Solvation (COSMO-RS) is software which uses quantum mechanics to calculate thermodynamic properties and has been utilised for *in silico* design of new solvents and solvent selection.[26]

Solvatochromatic parameters have been developed to describe the interactions between solutes and solvents and predict outcomes such as reaction rate, making them suitable for solvent selection. Kamlet-Abboud-Taft (KAT) parameters are an example of these. They consist of three components: π^*, polarisability; α, hydrogen-bond donating ability (acidity); and β, hydrogen-bond accepting ability (basicity).

These were developed for use in linear solvation energy relationships as shown in equation 1.1.[27] *Y* is a physiochemical property which changes based on the interactions between solvent and solute. Y_0 is a solute property for a solvent when π* = α = β = 0. Coefficients *a, b,* and *s* are dependent on the solute; thus, their magnitude indicates the strength of relationship between the solvent and solute for the corresponding parameter component.

$$Y = Y_0 + a\alpha + b\beta + s\pi^* \qquad (1.1)$$

Traditionally, these parameters are obtained from the ultraviolet-visible (UV-Vis) absorption spectra of solvatochromatic dyes. KAT parameters have been used to try and find alternative green solvents.[28] Inexpensive computational methods have been developed to calculate KAT parameters *in silico* using COSMO-RS.[29] Solvent mixtures can expand possible functionality. Blending solvents provides a method to attain desirable properties that otherwise may not be obtainable. However, solvent mixtures add complexity to calculations because in addition to solvent-solute interactions, solvent-solvent interactions must also be considered. Additionally, solvent mixtures can affect sustainability because solvent recovery or recycling is easier in single solvent systems.[17]

The Hansen Solubility Parameters (HSP) were developed in the 1960s and consist of three components: dispersion, *δD*; polarity, *δP*; and hydrogen bonding, *δH*.[30] These three components are used to obtain a solubility parameter, *δ*, as shown in equation 1.2.

$$\delta^2 = \delta_D^2 + \delta_P^2 + \delta_H^2 \qquad (1.2)$$

HSP was designed for and has mostly been applied to polymer solvation. Polymer solvents account for 78% of solvent usage in the EU and dipolar aprotic solvents such as N-methyl-2-pyrrolidone (NMP) and dimethylformamide (DMF) are commonly used.[31] Hence, there is an opportunity for further research aimed at discovering more environmentally sustainable alternatives. The HSPs have been integrated within the Hansen Solubility Parameters in Practice software (HSPiP). The HSPs and KAT parameters are summarised in Table 1.6.

Table 1.6: Kamlet-Abboud-Taft and Hansen solvent parameters and their components with the commonly used symbol and corresponding definitions.[27]

Conceptual Theory	Symbol	Definition
KAT	α	Hydrogen-bonding donating ability (acidity)
KAT	β	Hydrogen-donating accepting ability (basicity)
KAT	π*	Polarisability
Hansen	δD	Dispersion
Hansen	δP	Polarity
Hansen	δH	Hydrogen bonding

Machine learning has been used to predict the solubility of small molecules in aqueous and organic solvents.[32] This has also been applied to polymers.[33] Computer-aided molecular and process design (CAMPD) methods can be used to explore chemical space and design the most suitable solvent molecule and optimise the process, simultaneously.[34] COSMO-susCAMPD has been developed to holistically design sustainable solvents by integrating predictive LCA which utilises an artificial neural network with CAMPD.[34]

A different approach can be found in the Program for Assisting the Replacement of Industrial Solvents III (PARIS III) software. The PARIS III software contains over 5,200 chemicals and calculates properties for mixtures of these. A user enters the solvent or mixture of solvents used in their process. The software then returns to the user a list

of the 500 closest mixtures. The different impact factors in the software can be weighted allowing the user to select a replacement solvent mixture with similar functionality but improved sustainability.[35] The software is free to download and install and has been used to find potential replacements for solvents like carbon tetrachloride, toluene, and NMP.

1.2.2.1.3 Solvent Mapping

One challenge of solvent selection is the number of properties of interest. Due to the high number of properties of interest in solvent selection, visualisation and identification of substitute solvents can be challenging. Principal component analysis (PCA) reduces the number of dimensions of data by making new features called principal components. These principal components will maximise the variance they account for within the dataset, and it is possible to see the variance that each principal component accounts for. Using two principal components has obvious visual benefits as two-dimensional graphs are easily interpretable. A PCA mapping tool was developed by AstraZeneca (Figure 1.3) and is hosted by the ACS GCIPR and is free to use in a web browser, making it very accessible.[36]

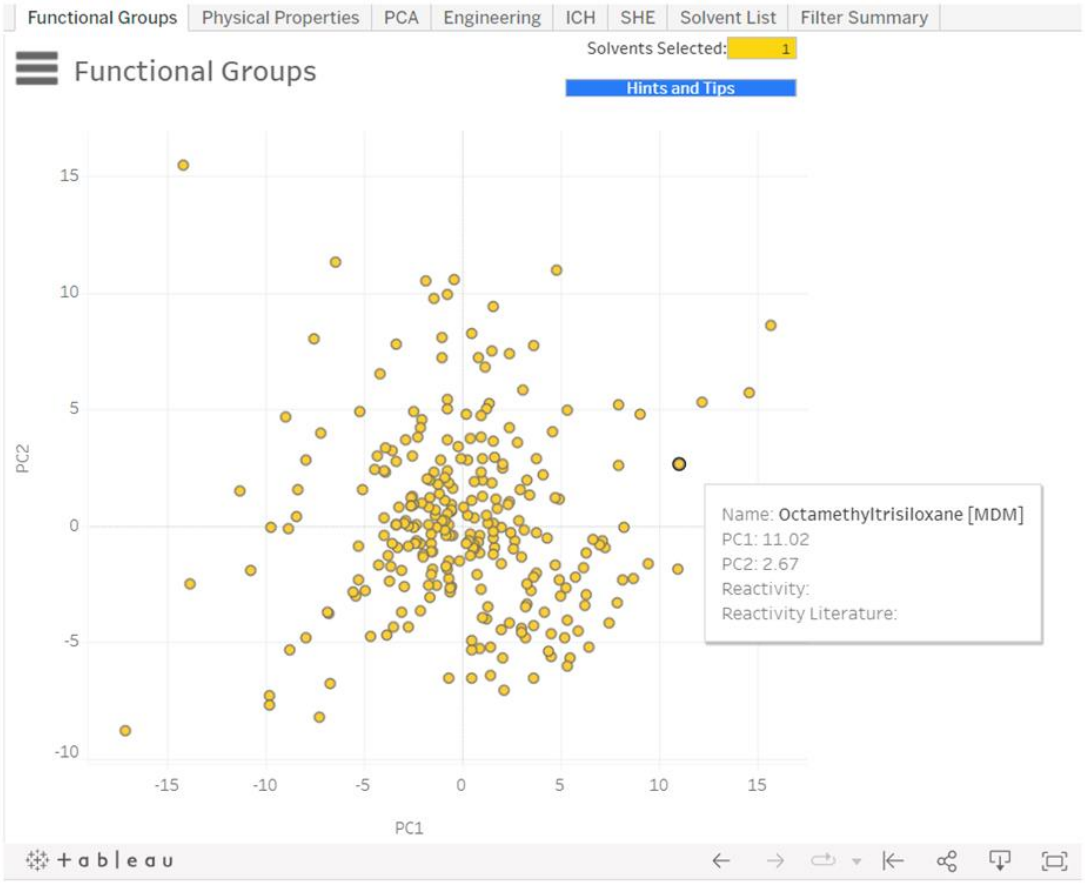


Figure 1.3: The ACS GCIPR Pharmaceutical Roundtable solvent selection tool. Solvents which are near each other have similar chemical and physical properties and distant solvents have different properties. The tool performs PCA to provide a two-dimensional mapping of the solvents and provides multiple filters to aid searching. One solvent of the 272 has been selected here.

The ACS GCIPR tool provides flexibility by allowing chemists to map solvents using PCA or other chosen properties. It also provides options to filter including by functional group, and safety, health, and environmental (SHE) data. Limitations to the tool include the inability to add new solvents, although 272 is a significant initial number to choose from. Additionally, different reaction classes may favour different properties and incorporation of logic to reflect this into the tool could further boost its effectiveness. Interactive PCA interfaces have been developed and these have been adapted for solvent selection, enabling chemists to impart their own knowledge onto

the tool by moving solvent points when the default mapping does not reflect the reality for their use-case.[37,38]

Sustainable solvents selection and substitution software (SUSSOL) uses a neural network to cluster solvents into groups based on their physical properties on a two-dimensional map.[25] SUSSOL has a dataset of 500 compounds with 22 physical properties. An advantage compared to other tools is its support for comma-separated value (CSV) files, meaning that the user can add their own solvents. The software is designed for commercial purposes, but the authors provide a limited version of the software and dataset on their GitHub repository which is free to use. A summary of exemplar tools discussed in this section is shown in Table 1.7.

Table 1.7: Selection of exemplar solvent selection tools.

Software Tool	Description
CHEM21 solvent selection guide[16]	A solvent selection guide based on scoring criteria then adjusted by an expert panel. https://learning.acsgcipr.org/guides-and-metrics/solvent-selection-guides/the-chem21-solvent-selection-guide/
HSPiP[30]	Commercial solvent selection software based on Hansen solubility parameters. https://www.hansen-solubility.com/HSPiP/
PARIS III[35]	Free to download software for finding more sustainable replacement solvent mixtures. https://www.epa.gov/chemical-research/program-assisting-replacement-industrial-solvents-paris-iii
ACS solvent selection tool[36]	Free to use in the web browser PCA-based solvent selection tool. https://www.acs.org/greenchemistry/research-innovation/tools-for-green-chemistry/solvent-selection-tool.html
SUSSOL[25]	Machine learning based solvent selection tool. Commercial software with a limited free version. https://github.com/SUSSOLKDG/Sussol

1.2.2.2 Computer-Aided Organic Synthesis

1.2.2.2.1 Retrosynthetic Analysis

Within Computer-aided organic synthesis, computer-aided synthesis planning (CASP) refers to using tools developed for planning synthetic routes. There are typically multiple routes to synthesise a target compound. Route planning is often done through retrosynthetic analysis where a target compound is iteratively "disconnected" into ever simpler intermediates and eventually commercially available building blocks. Corey introduced the concept of retrosynthesis in the 1960s, a contribution recognised by the 1990 Nobel Prize in Chemistry. During this period, early computational approaches to retrosynthesis were also developed.[6] The systematic nature of retrosynthesis makes it a suitable problem to be tackled computationally. It should be noted that retrosynthesis does not provide the conditions required for the transformations. However, knowing the conditions typically associated with different chemical transformations could contribute to assessing the sustainability of a synthetic route and may be implicitly learnt.

In 2005, Synthia was created and used hand-encoded rules; according to a 2020 publication the software included over 100,000 rules.[39] The merits of this approach can be seen in the success when applied to finding routes for complex natural products.[39] Following seminal work in 2018 on the use of deep neural networks for retrosynthesis, many tools for this purpose have been developed.[40] These tools require reaction data, of which there are a few options. Pharmaceutical companies may train them on proprietary in-house data. Other options include commercial

reaction data providers or open-source datasets, the largest of which is the US Patent and Trademark Office (USPTO) dataset.[41]

A chemist's motivations when developing a route will depend on the context. In the discovery phase of chemical research less emphasis may be put on sustainability due to a greater focus on selecting the route which is most feasibly going to yield product in the desired quantity in the least amount of time. However, this initial win may be costlier in the long-term if the synthesis must be re-worked later.[5] Integrating sustainability directly into CASP software using machine learning techniques could enable optimisation of route sustainability at an earlier stage without additional overhead for the chemist.

Integration of sustainability into CASP software has been implemented by Wang *et al*.[42] This implementation led to identification of shorter routes with greener solvents. This methodology requires integration of sustainability considerations into the search algorithm used to find retrosynthetic routes. This is a challenge as in retrosynthesis only the reactants and products are defined, meaning the sustainability of the solvents, reagents, and other conditions would not be defined at this point. Wang *et al*. overcame this by using a trained neural network which returns predicted conditions for a reaction and assigned scores to solvents based on their biosafety and flammability.[43]

Retrobiosynthesis is a new emerging area of CASP tools. These use enzymes for biocatalysis of reactions avoiding harmful reagents such as traditional heavy metal catalysts. A dataset consisting of over 62,000 unique enzymatic reactions was created

by extracting data from multiple sources.[44] Exemplar tools discussed in this section are summarised in Table 1.8.

Table 1.8: Selection of exemplar software tools for computer-aided synthesis design

Software Tool	Description
ASKCOS[42,43]	Open-source computer-aided synthesis design toolkit. Including retrosynthetic analysis and condition prediction. https://askcos.mit.edu/ https://github.com/ASKCOS/
Synthia[39]	Commercial rules-based computer-aided synthesis design software. https://www.synthiaonline.com/
RXN for Biochemical reactions[44]	Open-source data-driven retrosynthetic analysis software for bio-catalysed reactions. https://github.com/rxn4chemistry/biocatalysis-model

1.2.2.2.2 Condition Prediction

A retrosynthetic route only breaks down the target molecule into building blocks. It does not describe the methods required to achieve to the desired transformations. One example would be ester hydrolysis. A reaction which can be done in either basic or acidic conditions. Empirical evidence can be used to find suitable conditions. On reaction literature curation services, such as Reaxys and Sci-Finder, a chemist can search for molecules structurally similar to their own and see what methods most commonly give high yields.

Additionally, a theoretical approach can be used to select the conditions that are most compatible with the molecule. For example, avoiding using any conditions that other regions of the molecule would be sensitive to and give an unwanted additional reaction. Rational reaction design requires enough energy and reactive species to affect the desired change and yield the new product. Some of the factors molecules can be sensitive to include air, water, heat, acid, base, light and other reactive species.

In addition to molecular stability and sensitivity under different conditions, stereo and electronic effects may reduce or increase reactivity, which may require a change to the conditions. This can become challenging when dealing with the synthesis of complex molecules, such is often the case with pharmaceuticals and many other molecules of interest. Protecting group strategies can be particularly important for this and this relies on the retrosynthetic planner to learn these. Protecting group strategies involve concealing a reactive moiety, such as an amino group with a less reactive group that can be removed under relatively mild conditions. Machine learning models have been trained on reaction data where each component is labelled with its role in the reaction to predict conditions. The data should allow the models to learn what conditions different reactions require and what conditions different chemical moieties can tolerate. ASKCOS includes a context recommender tool which predicts the temperature, catalyst, solvent, and reagents required for a chemical reaction.[43]

1.2.2.2.3 Yield Prediction

Yield prediction tools can either be generalised or specific. Data quality is a significant barrier to making a generalised yield prediction tool. A reason for this is the variation in motives for synthesis. For example, when looking at patent literature yields may be artificially low for many reactions where the aim is to isolate a small mass of high purity compound for testing. This contrasts with much of the chemical literature in academic journals where the aim is to maximise the yield. Additionally, reproducibility is a factor where different chemists following the same procedure may get varying yields.

1.2.2.2.4 Other Tools

Tools to predict the product of a reaction have been developed. These include general tools suitable for all reaction types such as ASKCOS, and tools with a specified reaction scope. General product prediction tools can be used to identify the likely major product of a reaction and can provide some reassurance in validating a predicted retrosynthetic reaction. Tools with a narrower scope are developed to deal with complex regioselectivity problems. An example of this is C-H bond functionalisation. Normally, this reaction involves the replacement of a C-H bond of an aromatic or heteroaromatic molecule with a C-X bond, where X depends on the moiety being added. This can be predicted by calculating the energies of the transition states.[45] However, these calculations are expensive and machine learning methods have been developed to provide computationally cheaper alternatives.[46]

1.3 In Silico Methods as a Sustainable Alternative

In this chapter the application of *in silico* methods for property prediction and their beneficial impact on software tools for green and sustainable chemistry have been discussed. *In silico* methods can improve sustainability by addressing one of the core elements of green chemistry, reduction. Replacing experimental processes with *in silico* methods can improve the sustainability of a process, but it is important to emphasise this is not a trivial task. Historically, quantitative structure-activity relationship (QSAR) and, more recently, machine learning models have been developed to predict a wide range of properties, which normally would be obtained through more costly experimental methods. These methods, like many others mentioned throughout this chapter, rely on high quality datasets for accurate results.

QSAR is a modelling method for discovering the relationship between structural features in a molecule and properties of interest. This is commonly used in drug discovery in relation to the biological activity of a compound. For QSAR models to be beneficial, they should be capable of aiding smarter compound design and reducing the number of compounds that must be synthesised and tested. If successful, this approach has sustainability benefits by reducing the impact associated with the synthesis and testing of compounds.

QSAR uses either regression or classification techniques. This contrasts with machine learning, which includes additional statistical techniques such as clustering and generative models capable of outputting molecules. QSAR can make use of machine learning techniques, and in recent years, this has become increasingly common. Classical QSAR modelling used less data-intensive methods and focused on the analysis of small sets of data, Hammet plots, for example.[47] Integrating machine learning involves applying machine learning algorithms to large volumes of data to make models. This includes algorithms like random forest and neural networks. Machine learning involves using data to train a model according to an algorithm. The most basic example of a machine learning model is linear regression. Equation 1.3 shows an example of multiple linear regression, where *y* is the predicted value, *a* is the coefficient or weight which has been optimised during training, *x* is the variable and *c* is the intercept or bias and is also optimised during training.

$$y = \{a_1 x_1 + \ldots + a_n x_n\} + c \qquad (1.3)$$

The use of a_n *and* x_n is because multiple variables can be used. Initially the model will use a random weight and bias. During training, a loss function will determine how far

away the predicted outputs are from the true outputs using the training data. The result of this loss function will be used to determine how to update the weights. This process repeats for a set number of iterations or until the error is below a particular threshold. These core principles hold true for other machine learning models, but the use of more sophisticated algorithms and models can uncover more complex relationships between the input variables and outputs. For example, linear regression is unsuited for capturing non-linear relationships. These models can then be assessed on previously unseen data to determine their accuracy.

Predictions of toxicity to reduce reliance on animal testing is an example of an area of interest. A long-standing concept in the field of animal testing is the three Rs of replacement, reduction, and refinement. These principles are provided as guidelines by major organisations such as the OECD.[48] These three Rs are a good example of an area where *in silico* methods are a much sought after alternative. LHASA have produced software for *in silico* prediction of drug toxicity in addition to other properties.[49]

1.4 Future Perspectives

Software relies on the availability of computational power. The continual progress in hardware, exemplified by Moore's Law since its conceptualisation in 1965, has contributed to a rapid escalation in computing capabilities. Moore’s law states the number of transistors on a microchip doubles every two years, which increases the computational power relative to hardware size. This observation made in 1965 has held true for a long time but recently there has been debate as to whether it is still true.

The growth in computational power is especially noticeable in the recent development of advanced large language models (LLM). ChemCrow is an LLM designed for chemistry use and uses specialised tools to provide specialist chemistry functionality and compensate for weaknesses inherent to LLMs.[50] There are no green or sustainability tools within this model, but it is foreseeable that these could be integrated into this or other models into the future. The LLM could be trained on resources on sustainability such as related literature papers, textbooks, or encyclopaedia. This could enable the LLM to give expert feedback to chemists who are not themselves experts in green and sustainable chemistry.

However, these technologies are not without concern. There are multiple social, economic, and environmental concerns related to artificial intelligence (AI). The financial and energy cost of training large models makes it prohibitive to many, potentially increasing inequality.[51] For example, in 2020 Alphabet (Google) had an R&D budget of $27.57 billion higher than that of India or Spain; this has potential implications on public university research.[51]

The development of more sustainable methods in chemistry has been a huge focus with numerous advancements. It can be challenging for these to reach chemists who are not experts in green chemistry, especially those reluctant to change from the methods they are used to. Software offers great utility in reducing the burden of optimising sustainability for chemists. It can be integrated into existing workflows and new increasingly digitised workflows with many potential benefits for chemists.

1.5 Conclusions

This chapter has introduced the concepts of green and sustainable chemistry and contextualised the development of this field and how software supports this. The software which has been looked at can be grouped into: 1) Sustainability assessment software; 2) Software for the design of benign experiments; and 3) Software as an alternative to non-benign experiments. Various software applications for green and sustainable chemistry have been presented. The pharmaceutical industry, drug discovery, and organic chemistry have been covered extensively. However, attention has also been drawn to examples from other areas.

The fields of sustainable chemistry and software development have both seen simultaneous growth in recent decades. The potential of exciting new methods such as LLMs and retrobiosynthesis provide an indication that the growth trajectory for software tools for green and sustainable chemistry will continue. A greater focus on sustainability will continue due to regulation and an increasing prioritisation of minimising climate change.

The increasing capability to effectively use large volumes of data has prompted research aimed at curating better datasets through automated chemistry.[52] Automation and robotics have not been discussed in this chapter, but the future development of these technologies could have could have significant effects on sustainability.

There is still work to be done. Getting academic researchers to adopt digital methods such as ELNs is far from a finished task but is worth pursuing as it is the simplest way to integrate sustainability within existing workflows. Behavioural barriers to green

chemistry have been discussed, and work has been done to break down these barriers by focusing on the benefits that green chemistry can offer to a chemist's workflow. Another barrier is the potentially confusing myriad of options when it comes to green metrics. This may be overcome as green and sustainable chemistry is further integrated into undergraduate studies; future chemists may feel more confident in interpreting and selecting the appropriate metrics for their work.

The limitations of the software tools have been highlighted. A research and market gap for easy-to-use and integrated sustainability tools has been identified. This gap is more significant for open-source and non-proprietary software, which are far more accessible for most academics and students. Concerns over IP are an anticipated obstacle to ELN adoption. However, work on federated machine learning methods to anonymise data and prevent reverse engineering has been investigated and could overcome this barrier.[53]

1.6 Scope and Objectives of Thesis

This chapter has established the landscape of software tools for sustainable chemistry and the strengths and weaknesses of the current tools have been discussed. Subsequent chapters build on this landscape analysis with work which seeks to address the raised issues. Chapter two introduces the core concepts of machine learning and cheminformatics. In chapter three, novel methods are applied to predict the yield of reactions using *in-situ* sensors, providing a potentially greener, higher-throughput, and more data-rich alternative to traditional analytical instruments. Chapter four discusses the development of the AI4Green ELN which aims to integrate the sustainability tools described in this chapter into an ELN that captures data in a

FAIR way that can be used for the development of machine learning tools in the future. Chapter five discusses the integration of retrosynthesis with a sustainability assessment into the AI4Green ELN.

Chapter 2: Machine Learning for Chemistry – Methods & Theory

Abstract

Machine learning is a set of computational techniques that can use the underlying patterns in data for predictive power. These algorithms have been applied to problems in multiple domains, including chemistry. When applied to chemistry, suitable representations must be used for molecules and reactions. These representations can significantly impact performance as the quality of training data is directly related to model performance. This chapter will introduce and discuss the machine learning methods used in chemistry, data representations and the theory behind them.

2.1 Introduction to Artificial Intelligence and Machine Learning

Artificial intelligence (AI) refers to the concept of machines exhibiting human-like intelligence. One application of AI is machine learning, where systems learn knowledge by processing large amounts of data. Machine learning is a statistical approach to AI. An alternative approach is symbolic AI, which relies on logic and a knowledge base encoded through a series of rules. Symbolic AI was the dominant approach from the 1950s until the late 1980s.[54] Symbolic AI was used to develop expert systems such as DENDRAL, which was capable of suggesting potential molecular structures from mass spectra.[55] Another notable example is IBM's Deep Blue, which defeated chess world champion Garry Kasparov in 1997.[56] Despite these successes, these systems face issues that limit their utility. Knowledge acquisition, the process of extracting all the relevant information that a human expert would use when solving a problem, can be challenging. Additionally, there are "fuzzy" problems, which are poorly suited for explicit rules.[57] An advantage of Symbolic AI is the transparency of its workings due to the use of explicit rules and logic, meaning the results of symbolic AI can be more easily interpreted. Model interpretability is important for many reasons, including trust, detecting biases that may have ethical implications, and gaining useful insights that can be used to improve the model.

Machine learning methods have also existed since the 1950s, with significant advances occurring throughout the latter half of the 20th century. However, it was not until the 2000s that a more favourable environment for machine learning emerged. The volume of data available grew beyond human interpretability, along with decreasing costs and the increasing power of computational hardware, such as graphical

processing units (GPUs) (initially designed for rendering graphics in video games). Additionally, improvements in machine learning algorithms led to more robust models.[58] These factors - more data, better hardware, and more algorithmic advancements - enabled higher quality machine learning models with greater utility. These trends are still ongoing, likely leading to further advancements in machine learning. An example demonstrating the potential of machine learning is the development of AlphaFold and its subsequent iterations, which can accurately predict the three-dimensional structure of a protein from its amino acid sequence.[59] Machine learning is typically less explainable than symbolic AI due to the use of statistical correlations rather than explicit rules. This tends to be increasingly true for more complex models, with deep learning models often described as a black box. However, techniques such as local interpretable model-agnostic explanations (LIME) and Shapley additive explanation (SHAP) values can help with this.

Machine learning involves training algorithms to identify correlations within data. This can be divided into supervised, unsupervised, and reinforcement learning. Supervised learning works with labelled data where both the input features or variables and output are known. This contrasts with unsupervised learning. Unsupervised learning works with unlabelled data, which consists only of input features and has no associated output. Unsupervised learning techniques often involve identifying patterns and structures in the data. Supervised learning identifies correlations between the input features and the output, enabling predictions on previously unseen data. Reinforcement learning involves an agent learning through the feedback obtained through trial and error. Examples of applications include robotics and self-driving cars, and these can also be applied in simulated environments.

Many machine learning algorithms exist within these broad categories. They have differing objectives, levels of complexity, associated computational costs, and limitations and assumptions. This means that different algorithms are use-case dependent on factors such as volume of data, data structure, and desired objective, which will all play a role in determining suitable models for a specific situation.

2.2 Implementing Machine Learning

2.2.1 Data Collection

As perhaps the most famous adage in machine learning goes, “garbage in, garbage out”. The most important step to build a predictive machine learning model is to prepare a suitable dataset for the intended goal. In many situations, relevant data may have already been collected and made available in an online resource such as a database or online repository. If this is the case, its usability may depend on how FAIR the data are. For example, the data can be open-source and available or behind a paywall. If no suitable data already exist, then data must be collected. Traditional data collection can be challenging; for example, collecting a large enough volume of reaction data for a target problem can take a long time. In chemistry, methods such as high-throughput experimentation (HTE) can rapidly generate data. Sensors and data-collecting devices related to the IoT can gather a large volume of data at a low cost. Alternative sources of data include file parsing or web scraping. Data can be parsed from a file, or a script can be used to extract data from the Hypertext Markup Language (HTML) content of a website.

Data from multiple sources can often be combined. However, this can come with associated risks. For example, data from different biological assays should not be

combined to try and predict drug activity, as biological assays vary considerably and are typically not comparable.

2.2.2 Data Preprocessing

Once collected, data can be pre-processed to make them more suitable for the machine learning task. This can involve adding, modifying or removing features, or generating new datapoints through synthetic data generation methods or dropped datapoints. The volume of data used should balance the predictive power of the model with the computational resources required to train it. Redundant data may slow training and add noise, reducing model accuracy. In many domains, data scarcity means that this problem is not a cause of concern but can lead to other issues. The number of datapoints used should be significantly higher than the number of features used to avoid the curse of dimensionality.[60,61] It is also possible to condense features into fewer features by unsupervised learning techniques, which aim to capture the variance and relationships within a dataset in the minimum number of features.

This is a consideration when removing or adding features. Features can be added or modified to make new features more relevant and useful for the given task. For example, if a sensor records the time of a measurement, it can be converted to seconds after a defined start time. Other times, more complex features can be generated from the data. Molecular descriptors, for example, can be generated from structural representations and provide vectors with encodings of features which describe the molecule, with data such as the functional groups present.

The data can also be cleaned by identifying erroneous datapoints and removing them from the data. Features that are not relevant can be dropped. This can include

features with no relationship or correlation to the predicted output. Analysis can also identify strongly correlated features; therefore, removing one or more of these features may reduce the volume of potentially redundant data. The input feature chosen should be related to the output. Domain knowledge can assist with this, and features with a known relationship to the output are preferable. However, it can be worth exploring features without ground-truth causation to the output, as there may still be a correlation.

Exploring and understanding the data before using them in a machine learning model can be valuable. This can give an understanding of what features correlate with one another and which correlate with the predicted output. It can also indicate the space the data occupies. Space that is unoccupied by the data is unlikely to be accurately predicted by the model. Additionally, over-occupied space may lead to a bias in the model. Ideally, the data should be evenly distributed over the target domain.

To summarise, the quality, quantity, coverage, and potential bias of the data should be evaluated and influence the selection of a suitable model.

2.2.3 Model and Feature Selection

After preprocessing, different models and features can be assessed. Typically, data are split roughly 70:20:10 into training, testing, and validation data. The model is trained on the training data, and then its performance is tested with the test data. The validation data are held back at this stage to avoid over-optimisation for a specific test data set. If data are scarce, the validation set can be dropped, and just a training and test set can be used. In this scenario, alternative methods can be used to provide adequate scrutiny, such as cross-fold validation where, if the train: test split used is

90:10, the 10% used as test data changes each time. This slows down the assessment, but this tends to be less problematic as the volume of training data is small. Validation is the last step in developing a machine learning model. The full process is shown in Figure 2.1. The model may then be published, deployed, or used internally for the intended task.

Figure 2.1: The steps involved in developing a machine learning model.

More complex models, such as neural networks, typically require more data than less complex models, such as linear regression. It can generally be advisable to start with less complex models and move to more complex models as needed. Less complex models tend to be more explainable in their predictions and require fewer computational resources.

In addition to the model and features, model hyperparameters can also be changed. These can influence factors such as model size and algorithms used, which can significantly impact model performance. Before use in a model, feature scaling is performed to get the features within a similar magnitude to prevent features with larger values from disproportionately affecting the output. Additionally, this step is performed after splitting the training and test data to prevent data leakage between the two sets.

Various metrics can be used to assess the model performance. Metrics used for classification and regression problems differ significantly due to the difference in

output type. Some of these metrics overlap but can provide slightly different insights depending on the specific aims and use case. One or more of these metrics can be used to provide a framework with which the model performance can be assessed and the best combination of models, features, and hyperparameters found.

Pitfalls include issues with over or underfitting. Overfitting is when the model too closely matches the training data and performs significantly worse on the test set than during training. This can potentially be avoided by methods such as dropout layers in neural networks where a percentage of information is discarded during training. Underfitting is when the model does not capture the relationship between the input features and the output and features, and hyperparameters should be carried forward and tested on external or validation data to test model generalisability.

2.3 Machine Learning Methods

Machine learning models rely on parameters or structures that influence how input variables are processed to generate predictions. During training, the model is optimised to minimise the discrepancy between its predictions and the actual outputs. The optimisation process varies depending on the type of model.

Gradient-based methods, such as neural networks and linear regression, iteratively adjust parameters to minimise a loss function. This process is repeated until a desired level of accuracy is achieved or an iteration limit is reached. Different types of loss functions are required for different problem types. For example, regression models can use mean squared error (MSE) or mean absolute error (MAE) as a loss function. These loss functions have subtle differences in how they influence model training; MSE penalises larger errors more heavily, which makes it more sensitive to outliers

than MAE. Classification models require different loss functions, and other specialisation loss functions are also used for tasks involving images. Custom loss functions can also be developed to achieve a specific goal within an application.

Other machine learning methods do not rely on loss functions in the traditional sense. For example, tree-based methods use criteria to split data at each node and ensemble methods aggregate predictions from multiple trees to improve accuracy.

Hyperparameters also vary by model type. For tree-based models, key hyperparameters include the number of trees and the maximum depth. For gradient-based models, the learning rate (α) can control the magnitude of updates to the iteratively updated parameters. If the learning rate is too high, the weight adjustments will repeatedly overshoot, preventing the model from converging to the loss function's minimum. Too small of a learning rate will require more iterations of training to the minimum of the loss function and, depending on the iteration threshold, may fail to reach the minimum of the loss function.

2.3.1 Supervised Learning

Supervised learning algorithms are either classification or regression models. Classification models predict a discrete class from the inputs, whereas regression models predict continuous values. Only regression methods are used and discussed further in this work. Different regression models are suited to different tasks. This depends on factors such as model complexity and volume of data, with more complex models often requiring more data but being able to make powerful predictions when given that data. Models have different assumptions regarding the data and vary in the type of relationships they can capture between the inputs and the output.

The bias-variance trade-off is a problem in machine learning, where bias is a systematic error that occurs to assumptions made by the model. Linear regression is an example of a high-bias model as it assumes a linear relationship between features and the output which can lead to it failing to capture meaningful complex relationships. On the other hand, variance is how well the model learns the relationships in the training data. High variance can lead to overfitting as it may have learned relationships between noise in the data and the outputs. This means the model's performance is sensitive to fluctuations in training data and the predictions change when trained on different training data. Ideally, bias and variance will be low, meaning the model learns complex relationships in the data whilst not being overly sensitive to noise in the data.

2.3.1.1 Linear Regression

In linear regression, it is assumed that there is a linear relationship between the input variables and the output. It also assumes that the input variables are independent of one another. Linear regression can have one or more input features. Linear regression with *n* variables is represented by Equation 2.1. *y* is the output, x_n is the input variables or features, *and* is the gradient coefficients, and *c* is the intercept. The intercept, *c*, can either be free or be set to 0.

$$y = \{a_1 x_1 \ + \ \dots \ + \ a_n x_n\} + c \tag{2.1}$$

To develop the linear regression model, the values of a_n and *c* are optimised to calculate the outputs for the training data. The differences between the correct outputs and the predicted outputs are the residuals. *a* and *c* are optimised using the least squares method, which minimises the sum of the squares of the residuals. Linear

regression is a useful model due to its low level of complexity and ability to identify linear relationships between the input variables and the outputs. However, linear regression models perform poorly on complex data, which often do not exhibit a linear relationship between the input variables and the outputs. Linear regression models are typically high-bias low-variance models.

2.3.1.2 Polynomial Regression

Polynomial regression operates similarly to linear regression. However, instead of assuming a linear relationship between the input features and output, a suitable order polynomial relationship is found and assumed to be the relationship. A third-order polynomial using a single feature to predict the output *y* is represented in Equation 2.2. The coefficients *a*, *b,* and *c,* and the intercept *d*, are optimised using the least squares method. This polynomial regression equation assumes there is no dependency between the features. However, cross-terms (such as x_1x_2, where both x_1 and x_2 are input features) can be introduced to capture the dependencies between features. This increases the complexity of the model but does enable it to represent more complex relationships. This enhanced ability to capture complex relationships means the bias of the model is lower compared to linear regression.

$$y = ax + bx^2 + cx^3 + d \tag{2.2}$$

2.3.1.3 Random Forest

Random forest models can be applied to either classification or regression problems. In this work, only random forest regression (RFR) is used and discussed. RFR is an ensemble method that constructs a specified number of decision trees to make predictions. Increasing the number of trees typically improves model performance

with diminishing returns as the number of trees increases. The model uses random sampling to reduce the risk of overfitting by only training each tree on a random subset of the data, often around two-thirds of the training data. Since each tree is trained on a different subset of data, the risk of overfitting is reduced.

As the decision trees build, they branch using a split criterion to find the optimal feature and threshold to split the data. The criterion for this is typically a measure of the difference between the predicted and actual outputs, such as mean squared error (MSE). Each tree outputs a continuous value, and the average of these predictions is computed to provide the output of the random forest.

The Random Forest model addresses the bias-variance trade-off by combining the predictions of multiple decision trees. Individual trees, trained on samples of the data, can capture complex relationships and exhibit low bias. However, these trees may have high variance because each is trained on a different subset of data and splits are based on randomly selected features, leading to diverse predictions. By averaging the predictions across all trees, Random Forests reduce variance while maintaining low bias, resulting in a model that generalises better. This strategy effectively balances the bias-variance trade-off.

2.3.1.4 Gradient Boosting Regression

Gradient Boosting Regression (GBR) is an ensemble method that combines multiple decision trees. In each iteration, the algorithm builds a new tree that attempts to reduce the residuals or the errors made by the previous trees. This process uses functional gradient descent to optimise its predictions by minimising a loss function. Initially, the first tree predicts the target variable from the raw data. Each subsequent

tree learns from the residuals of the prior predictions, progressively increasing the model's accuracy. This iterative process transforms a collection of weak learners into a strong learner, an essential concept of ensemble methods.

The initial tree is a weak-learner, a shallow decision tree with high-bias that struggles to capture complex patterns. The process of sequentially making new decision trees which aim to correct the previous errors lowers the bias of the model as it progressively fits more complex patterns in the data. However, this process can increase the model's variance as it becomes increasingly fitted to the training data. The bias-variance trade-off within the model can be managed through hyperparameter tuning.

This method can be contrasted with RFR. GBR is better suited for complex datasets where capturing complex relationships and low model bias are high priorities. RFR is more adept at handling noisy data and is better-suited when robustness and low-variance are priorities.

2.3.1.5 Neural Networks

Neural networks consist of multiple layers of interconnected neurons or nodes. The layers consist of an input layer, an output layer, and potentially multiple hidden intermediate layers. Each node has an associated weight and bias which are randomly initialised and optimised during training. The first layer receives the input features, and each node computes an output by obtaining the product of multiplying the input features with the weight, adding a bias term, and applying a non-linear activation function. The output is passed to all nodes in the next layer. In the initial layers, nodes with substantial weights for specific input features can indicate a strong relationship

between the input feature and the model output. This process continues, with the data becoming increasingly abstract, until the output layer is reached.

In regression tasks, the output layer typically contains a single node with a linear activation function that produces a continuous numerical value. During training, this output is evaluated by a loss function such as MSE. The backpropagation algorithm then updates the weights and biases based on how much they contribute to the model's error.[62] This leads to adjusting the weights and biases to minimise the loss function and improve model accuracy. This describes the mechanism of a basic feed-forward neural network.

This can be contrasted with linear regression, which does not apply non-linear transformations, and polynomial regression, which only applies a fixed-order non-linear transformation. This flexibility in training allows complex relationships to be captured.

2.3.1.6 Long Short-Term Memory Neural Networks

Long short-term memory (LSTM) neural networks are a type of recurrent neural network (RNN) designed for learning sequential data.[63] RNNs have recurrent connections that allow information from one datapoint to be retained for the next datapoint within a layer, which acts as a memory for the network (Figure 2.2). LSTMs make use of gated memory cells, which control the flow of new memory being added to the cell, old memories being removed, and how much memory is output to the next layer. This enables LSTMs to store relevant data from earlier in the sequence and makes them capable of capturing long-term dependencies and sequential relationships in data.

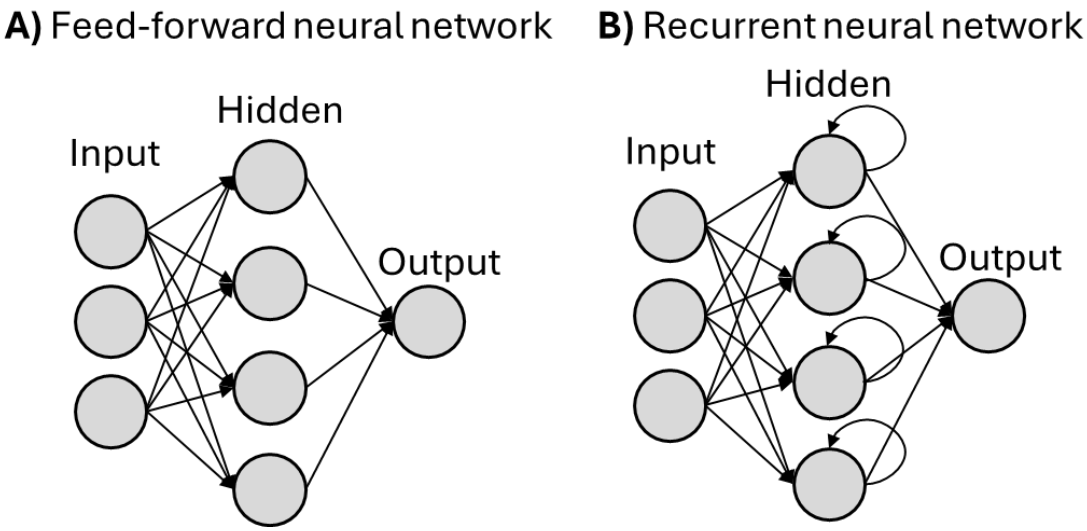


Figure 2.2: Simplified schematics of two neural network architectures. **A)** In a standard feed-forward neural network, data flows in one direction: from the input layer, through the hidden layers, to the output layer. Each layer applies a transformation to the data. **B)** In a recurrent neural network, the hidden layer includes a recurrent connection that feeds the previous hidden state back into the hidden layer. This feedback loop allows the network to maintain context when processing sequential data.

2.3.2 Preprocessing Methods

2.3.2.1 Feature Scaling

There are multiple feature scaling methods with subtle differences. The general concept of feature scaling is to transform the input variables such that they are of a similar magnitude. This means the model has the potential to learn equally from the features, irrespective of their original magnitude. Min-max scaling rescales all the values of a feature to be within a specified range, normally between -1 and 1 or 0 and 1. Standardisation rescales the values of a feature to have a mean of zero and a standard deviation of one.

2.3.2.2 Yeo-Johnson Transformation

The primary purpose of a Yeo-Johnson power transformation is to make data more Gaussian-like. This can be particularly useful for models, such as linear regression, that

assume the data is normally distributed. The Yeo-Johnson power transformation can handle negative and positive values, whereas the similar Box-Cox power transformation requires all values to be positive, as it cannot handle negative values. The Yeo-Johnson power transformation raises the variable to the power of parameter λ. The maximum likelihood estimation shown in Equation 2.3 is used to determine the optimal value of λ that is most capable of transforming the data into fitting a Gaussian distribution.

$$y_i^{(\lambda)} = \begin{cases} \big((y_i+1)^{\lambda}-1\big)/\lambda, & if\ \lambda \neq 0,\ \mathrm{y} \geq 0 \\ log(y_i+1), & if\ \lambda = 0,\ \mathrm{y} \geq 0 \\ -\big((-y_i+1)^{(2-\lambda)}-1\big)/(2-\lambda), & if \lambda \neq 2,\ \ \mathrm{y} < 0 \\ -\log(-y_i+1) & if\ \lambda = 2,\ \ y < 0 \end{cases} \tag{2.3}$$

2.3.2.3 Synthetic Minority Over-Sampling Technique for Regression with Gaussian Noise

The synthetic minority over-sampling technique for regression with Gaussian noise (SMOGN) algorithm can be used to address an imbalanced dataset. In many use cases, it can be important to predict edge cases where a failure may occur.[64] However, as these are edge cases, they tend to be underrepresented in the data. This can be particularly problematic if the dataset is small. SMOGN identifies underrepresented space in the data and transforms the dataset by over-sampling the underrepresented and under-sampling the over-represented data. This results in a more balanced dataset that can improve model performance, particularly for edge cases.

2.3.3 Unsupervised Learning

Unsupervised machine learning algorithms do not aim to predict an output. Instead, the identified patterns and structures within the unlabelled data are used for other

purposes. Clustering and dimensionality reduction are two examples. Clustering involves the algorithm trying to fit the data into groups. Clustering algorithms can use a specified number of clusters in methods, such as K-means, or the optimal number of clusters can be determined as part of the algorithm.

2.3.3.1 K-means

The K-means algorithm uses a cluster centroid for each cluster. A datapoint belongs to the cluster nearest to the centroid. Initially, each centroid is placed into the graph randomly. The algorithm has two iterative steps. In the first step, each datapoint is assigned to its closest centroid. The second step is calculating the mean position of all datapoints within a cluster and then moving the centroid to this position. These two steps occur iteratively until an iteration limit is reached or either the centroid position or the datapoint assignments stabilise. The resultant clusters can be visualised by colouring each cluster and showing them in a two-dimensional graph.

2.3.3.2 Principal Component Analysis

Principal component analysis (PCA) is a linear dimensionality reduction method. PCA can be used on a dataset with many independent variables or input features to reduce the number of dimensions. The new dimensions obtained are called principal components. The principal components can be used instead of the original dataset, which may be advantageous for visualisation and model performance. PCA reduces dimensionality whilst preserving the maximum amount of information in the data. It does this by creating new features that capture the maximum amount of variance from the data. The first principal component captures the largest amount of variance, and each subsequent principal component is orthogonal to the previous principal

components. This means the different principal components are uncorrelated, reducing redundancy in the data. Once PCA has been applied to a dataset, it is possible to see what percentage of the variance has been captured, and the number of principal components can be adjusted accordingly if desired.

2.3.3.3 Uniform Manifold Approximation and Projection

Uniform manifold approximation and projection (UMAP) is another dimensionality reduction technique with similar use cases.[65] Unlike PCA, UMAP can be used for non-linear dimension reduction, which makes it more suitable for certain datasets as it captures these more complex relationships. UMAP assumes the high-dimensional data lie on a manifold and constructs a topological representation of the data's structure. The aim is to generate a low-dimensional representation of the data that preserves the relationships between datapoints.

This is achieved by optimising the low-dimensional representation to minimise the cross-entropy between the low and high-dimensional representations using stochastic gradient descent. Cross-entropy measures the difference between datapoints in the low and high-dimensional spaces by comparing the probability distributions of their similarities. Cross-entropy is particularly useful in high-dimensional settings where metrics such as Euclidian distance become less meaningful. Minimising cross-entropy ensures the transformed data retain the structural relationships of the original high-dimensional space.

Unlike PCA, UMAP uses stochastic methods, meaning results can vary between runs unless the random seed is fixed. In addition to this difference, UMAP has tuneable hyperparameters that can be tailored to the specific dataset. t-distributed stochastic

neighbour embedding (t-SNE) has a similar function to UMAP.[66] Both methods preserve local relationships within clusters, but UMAP also preserves global relationships between clusters. Comparisons between the methods have found UMAP to be significantly less computationally expensive and quicker to run.[67]

2.4 Chemistry Representations

To apply machine learning to problems in the domain of chemistry, suitable representations of molecules and reactions are required. Reactions involve one or more molecules interacting to form a new product molecule. Hence, molecular representations will be explored first as these pave the way for building reaction representations.

2.4.1 Molecular Representations

Molecules can be represented in many ways, with different use cases that suit different representations; this is demonstrated in Table 2.1 using Indole as an example. Molecules can be drawn in two dimensions using different bond types to indicate their three-dimensional structure and stereochemistry with little ambiguity to a chemist. This is the most common way molecules and reactions are represented in the chemical literature. However, such representations are not practical for computational use. However, software has been developed to interpret images of chemical structures using machine learning computer vision methods.[68]

String identifiers are another way to describe molecules. Common names of ubiquitous chemicals, such as benzene or ammonia, are well-known and easy to understand. However, due to the vastness of chemical space, most chemicals lack a recognisable common name. The International Union of Pure and Applied Chemistry

(IUPAC) provides a systematic nomenclature, enabling the derivation of a name from a structure or vice versa. However, this method becomes complex for larger molecules. Opsin is an open-source tool that can convert an IUPAC name to a structure, but there are no open-source tools that reliably reverse the conversion of structures to IUPAC names. Although early versions of these tools have been developed using machine learning, these are still evolving.[69] Chemical Abstracts Service (CAS) numbers are another commonly used identifier. They are useful due to their brevity, recognisable pattern, and uniqueness. A compound may have multiple names, but it can only have one CAS number. This is particularly important in applications where data are queried, as a one-to-one relationship between identifier and molecule means only a single query is required. This can be useful in many contexts, such as searching within a database or chemical supplier website. These identifiers provide a link to the chemical structure but rely on additional data to get the chemical structure. However, the string identifiers described thus far do not inherently robustly convey the molecular structure and require additional data to associate the structure with the identifier. This can limit their application in some computational tasks as patterns between the chemical properties and names cannot be learnt. Additionally, although many molecules have CAS numbers, again, due to the vastness of chemical space, not all molecules have CAS numbers.

Molecular representations describe the structure of a molecule as opposed to identifiers which are effectively labels. This enables a person or machine capable of decoding the representation to determine the molecule. Wiswesser line notation (WLN) was invented in 1949 and was one of the first methods used to encode chemical structures. A key use of this was to enable efficient computational search of

molecules. By encoding the structure and using a line notation representation of it, substructures can also be searched for, something not as readily achievable using identifiers such as CAS numbers.[70] WLN is rarely used now; modern line notations have taken its place in the majority of applications.

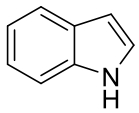
Another early example is chemical table (CT) files. These were developed by Molecular Design Limited (MDL) and first publicly described in a 1992 publication, but they have been used at MDL since 1979.[71] The simplest is a Mol file, which contains information for a single molecule, whereas a structure-data file (SDF) can contain multiple molecules. CT files contain a header with metadata and a connections table block, which describes the chemical structure and contains information on the atoms and bonds of a structure.

Another example is the Simplified Molecular Input Line Entry System (SMILES). SMILES was published in 1988, and the author attributed its timing to advancements in computational hardware and the application of graph theory to chemical notation.[72] SMILES are short strings of linear text used to represent molecules. SMILES provide a representation that is efficient for humans and machines to read. However, SMILES have some disadvantages. One molecule can be represented as different SMILES strings. This issue can be handled by canonicalising the SMILES strings using software such as RDKit.[73] Another disadvantage of SMILES is that when generating SMILES, invalid SMILES strings that do not correspond to real molecules can be generated, limiting their suitability in generative AI models.

The InChI was published in 2005 to address the issue of molecular degeneracy observed with SMILES. It achieves this by ensuring a molecule has only one InChI

string.[74] Since then, the InChI has been further developed to improve support for more complex chemical representations, with version 1.07 released on GitHub in August 2024.[75,76] InChI is more challenging for humans to read but is not unreadable. Their advantages include unique strings that aid database searching, more information encoded using multiple layers, and the ability to hash as a 27-character InChIKey to aid retrieval, where the raw InChI may be much longer. Due to their complexity, it is also possible for generative AI to produce invalid InChI.

Table 2.1: Different representations and identifiers of Indole.

Representation	Indole as an Example
Common Name	Indole
IUPAC Name	1*H*-indole
Synonym	Benzopyrrole
CAS	120-72-9
SMILES	C12=CC=CC=C1C=CN2
InChI	InChI=1S/C8H7N/c1-2-4-8-7(3-1)5-6-9-8/h1-6,9H
InChI Key	SIKJAQJRHWYJAI-UHFFFAOYSA-N
Structure	

2.4.2 Reaction Representations

As reactions consist of molecules, most reaction representations build upon serialised forms of the molecular representations with methods to indicate the role of a compound in a reaction. Compounds in reactions can have one of a few roles. Reactants, reagents, catalysts, solvents, and products. However, these roles can overlap, complicating the issue.

2.4.2.1 Introduction to Reaction Representations

Anything to the right-hand side of the arrow in a reaction is treated as a product. This includes byproducts, which are often omitted. This can lead to missed health and

safety concerns and issues with atom mapping and reaction balancing, where all the atoms on the left-hand side of the arrow are mapped to their corresponding new position after the reaction. Byproducts typically refer to small molecules that are expected to be produced as part of the reaction, such as water in a condensation reaction. Additionally, side products are often also omitted. Many molecules can undergo more than one reaction in a particular set of conditions, and a small amount of this product may be formed in addition to the desired product. This side product should be removed during purification, which can often be challenging as it will likely be chemically similar to the main product. Related to this is data on purity and yield, which are two key metrics for the success of a reaction. These missing data related to the outcome of a reaction can impair the ability to judge the quality of a reaction. Reactants and reagents can have significant overlap. Reactants are generally defined as molecules with atoms which appear in the desired product. However, there are ambiguous cases. For example, it is common for small molecules, such as many reducing agents which contribute a hydrogen atom to the product, to be treated as reagents. This also applies to solvents, where protic solvents can protonate a molecule in the final step of a reaction mechanism to yield the product. Reagents and catalysts are sometimes treated differently, and their roles are specified separately. However, they are often combined with reagents, including bases. Catalysts can potentially be identified, as coordinated metal species with electron-donating ligands are typically catalysts. In some situations, reagents and solvents can be used interchangeably, where the reagent is used in significant excess and is effectively the reaction solvent. These overlaps, interchangeability, and variations in reporting methods make representing reaction data a significantly more complicated task than representing

molecular data. Many representations give space for the inclusion of additional reagents, catalysts, and solvents. However, these are often grouped as agents, and judgement is required to deduce which role each agent performs. Practically, these are often excluded, with only the reactants and products included in the representation. Although this method does simplify the process, a significant amount of data is lost. Most methods also ignore factors such as temperature, pressure, and other conditions, such as stir rate, reaction method including photochemistry, vessel type, whether microwave heating was used, whether the reaction was performed in batch or flow and more. The loss of these data can hinder the reproduction of the reaction since the conditions can often control which reaction, if any, proceeds.

2.4.2.2 Record Keeping

The first step is the recording of the reaction data. Traditionally, this is done in a paper lab notebook belonging to the chemist. These can be personal and extend beyond simple data input and contain notes, ideas, and hypotheses. Paper and pen provide the chemist with a canvas that is as flexible as possible. Often, the reaction data are recorded in multiple sessions. A plan is drawn up with a visual reaction scheme and a table or list of all other components and their relative amounts, masses, or volumes. Then, the reaction is set up, and its progress is monitored. At this point, the chemist can record their actions so far, along with any observations and analysis. Finally, the product can be isolated and analysed, and the lab notebook entry completed with yield data, an experimental procedure, including purification, and analysis to indicate the identity and purity of the product. Whether these data are adequately recorded to reproduce the experiment is a long and ongoing discussion within the chemistry community. For example, it is not unexpected for a second chemist to obtain a

different yield when repeating a reaction with the same procedure. The aim will be to publish the reaction data as part of a wider investigation into a research journal or patent, depending on the context of the work. The format required for both is similar. The experimental procedure for each reaction is typed and accompanied by a scheme, with all the required data normally available in the experimental and the processed analytical spectra attached. Alternatively, a general scheme and protocol are written, and R groups are utilised to describe similar reactions where only a few parts of the reaction vary. This is normally done in a PDF format and included in the supplementary information if part of a research paper. It is possible to extract these data. This is done by commercial online reaction databases such as Reaxys and Sci-Finder. It is common for organisations to have subscriptions to one or both platforms. They often act as a first point of call when trying to identify suitable reaction pathways and conditions.

These current standard methods have some issues. Firstly, the data are not in a machine-readable format. Although there has been work into automatic data extraction using machine learning techniques, these have not yet found widespread adoption. One challenge is the variability within research articles, and patents, while more methodical, tend to lack the latest developments.[77,78] Secondly, not all of the data recorded in a project make it to publication. Failed or low-yielding reactions, known as negative data, often fail to make it to publication.

2.4.2.3 Machine Readable Formats

Machine-readable formats for reactions have been developed. As discussed earlier, their level of detail can vary as they balance simplicity and accuracy. Machine-readable formats can broadly be divided into these two categories: simplistic and complex.

Simplistic representations require additional data to be reproduced and other more complex representations capture these data by use of an extensive schema. Many of the more simplistic machine-readable formats group chemicals in a reaction to be reactants, agents, or products. Chemical table file formats also exist for reactions. These include Reaction (RXN) and reaction data file (RDF) formats. The RXN file format is more widely used and represents a single reaction by using a connections table block for each molecule in the reaction and labelling each molecule as either a reactant, agent, or product. Agents include reagents, solvents, catalysts or any other non-reactant substances used in the reaction. The RDF format can contain multiple reactions and additional metadata stored as key-value pairs. Reaction SMILES operate similarly. These consist of SMILES strings delimited by a dot and reactants and products separated by two arrows '>>' with agents, if present, appearing between the arrows. There is also the RInChI (reaction InChI) project. The RInChI operates similarly. It has six layers, layers one, two, three, and four contain the version of the software and the InChIs of the reactants, agents, and products. Layers five and six provide additional detail describing the direction of the reaction, including whether it is an equilibrium and whether any of the compounds used lack structures. It is also possible to have starting materials with no product in a RInChI format.[79] These formats, despite missing large amounts of data relevant to the reaction, still have strong use cases. For example, retrosynthesis uses only reactants and products and does not describe the conditions required. Therefore, these representations are suited for use cases such as that. Figure 2.3 depicts a hierarchy of different levels of detail for reaction data.

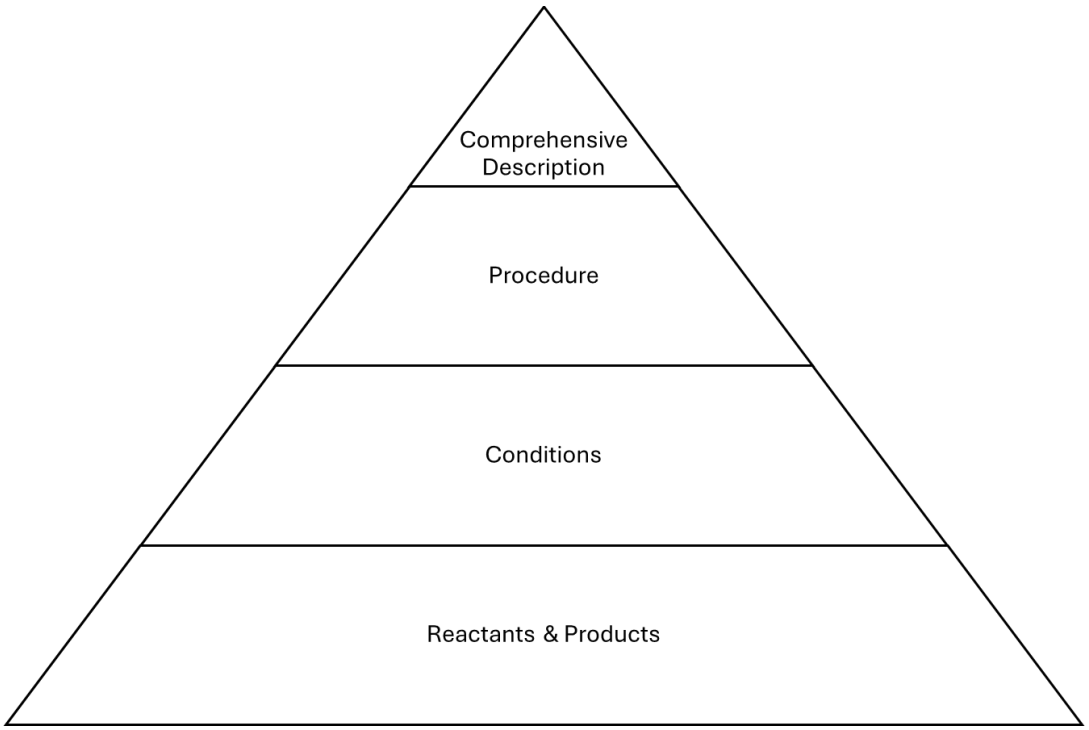


Figure 2.3: Reaction data hierarchy, where the base represents the simplest form of reaction data, and the top represents the most comprehensive. **Reactants and products** form the base as the most basic data that can be included. **Conditions** include data such as reagents, solvents, catalysts, temperature, atmospheres, and other components or methods influencing the reaction. The **procedure** includes the specific order of steps, how they were carried out and on what scale. This can indicate important aspects for reproducibility, such as whether the reaction was done in batch or flow and how the product was purified. **Comprehensive Description** providing additional detail to describe the dataset, including metadata, such as who performed it and when, and more comprehensive data to fully describe the experiment, for example, by including spectroscopic analysis.

2.4.2.4 Structured Data Repositories

Open-source repositories for chemical data have been developed; Chemotion and the open reaction database (ORD) are two such examples.[80,81] However, there is little incentive for busy chemists to digitise and suitably format their data for publication in one of these repositories. The barrier to publishing in these can be large. Electronic laboratory notebooks can begin to address this issue if they initially capture data in a structured way and provide suitable data export to these repositories. If the ELN manages this process using an extract, transform, load pipeline to extract the reaction data, transform it to the schema of the online repository and then load it into that

repository, then the barrier for the chemist is significantly reduced to potentially only a single button click. This workflow addresses the issues of missing negative and non-machine-readable data. Chemotion provides an open-source ELN, enabling anyone to create an instance capable of exporting data to the repository.[82]

The Chemotion and ORD repositories aim to use a comprehensive schema and store the data in a FAIR method. One reason for this is an interest in reaction automation, which requires the data to be sufficiently detailed for a machine to carry out the reaction.[83] These schemas provide a structure for the data, in contrast to the unstructured text and images used in a research paper's supplementary information.

Chapter 3: Machine Learning for Yield Prediction for Chemical Reactions Using *in situ* Sensors

Abstract

Machine learning models were developed to predict product formation from time-series reaction data for ten Buchwald-Hartwig coupling reactions. The time-series data were obtained using sensors, including an *in-situ* reaction probe. The data were provided by DeepMatter and were collected in their DigitalGlassware cloud platform. The reaction probe has 12 sensors to measure properties of interest, including temperature, pressure, and colour. Machine learning models were able to learn which of the properties were important, and colour was a found to be a good predictor of production formation for this reaction. Predictions for the current product formation (in terms of % yield) had a mean absolute error of 1.2%. For predicting 30, 60 and 120 minutes ahead the error rose to 3.4, 4.1 and 4.6%, respectively. The work here presents an example into the insight that can be obtained from applying machine learning methods to sensor data in synthetic chemistry.

Collaboration Acknowledgements

This work was done in collaboration with DeepMatter. The data were provided by DeepMatter and generated by chemists working at the contract research organisation O2H. Regular meetings were held with David Pattison, of DeepMatter at the time. David Pattison was responsible for the first iteration of the sliding window algorithm used in this work. This work has been published and is listed as Publication 1.[84]

3.1 Introduction

Machine learning can be applied to identify patterns and trends in large volumes of data with a trained model that generalises and is able to make predictions for unseen examples. Machine learning applied to chemistry is a growing field of research.[85] Many factors, such as advancements in graphics processing units, larger dataset collections and new algorithms have contributed to this renaissance.[86] Baum *et al.* observe that this growth is not uniform and postulate the reason fields such as analytical chemistry have seen faster developments compared to others like organic synthesis is due to the availability of large training datasets in areas which are traditionally more data intensive.[85,87] For example, in analytical chemistry, machine learning algorithms have been used to find chemical species at concentrations below the usual limit of detection by finding hidden patterns of signals within the noise.[88,89] However, considerable progress has been made in applying machine learning to organic chemistry. For example, machine learning techniques have been used effectively in CASP by training on reaction data from Reaxys or the United States Patent and Trademark Office dataset.[40,90,91]

Sensor data such as *in-situ* temperature and pH can be a good source for machine learning algorithms in time series modelling. Interest in sensor usage has grown with the development of the *internet of things* (IoT) – which is the exchange of data between internet-connected devices, and can be extended to include chemistry equipment and chemically relevant data.[92,93] The concepts of *cloud chemistry* or *telechemistry* have been introduced and involve remotely monitoring reactions by uploading the results from analytical equipment to the cloud.[94,95] IoT is part of the wider concept of industry 4.0, which has received attention in recent years and refers to the current trends of interconnectivity, data, and automation.[96] The application of machine learning techniques to real-time data, including sensor data, for predictive maintenance is an example of industry 4.0 practice.[97] Within process chemistry, these principles have been applied to enable predictive maintenance for pilot plants and self-optimisation of reactions.[15,98,99]

The use of sensors in organic chemistry is an emerging area, fuelled by advances in flow and automated synthesis.[100–102] Using sensors inside organic chemistry reactions generates data which could be valuable in the development of machine learning tools to augment the synthesis process, ultimately helping the chemist. In the field of reaction kinetics, hybrid models, which combine machine learning with traditional modelling methods, have found success at predicting the chemo- and regioselectivity of substitution reactions.[103,104] Sensors in chemistry can be used to monitor the reaction or to control the processes involved in performing the reaction. Mettler Toledo's ReactIR™ is used for monitoring reactions in real time using infrared (IR) spectroscopy, while FlowIR™ is an adaptation of this designed for use in flow

chemistry where additional sensors measure pressure and temperature to monitor the flow within the system.[105–107] In automation, conductivity sensors have been used to detect the phase boundary between two immiscible solvents during extraction.[108] There has been work to standardise the hardware and code used to run experiments to improve reproducibility of results and data sharing.[109] Despite significant advances in automation, most synthetic chemistry in the lab is done manually in glassware as batch chemistry. In this study, we explore the utility of using sensor data gathered from hand-performed reactions. We use data collected by DeepMatter's DigitalGlassware platform and a mix of proprietary and original equipment manufacturer sensor devices. Specifically, these consist of a DeviceX™ reaction probe (temperature, ultraviolet (UV), pressure, stir rate and camera) and a Vernier thermometer, both suitable for a multi-necked flask, and an environmental sensor which would go adjacent to the reaction setup. The sensors used in this study recorded and more importantly saved the data in an open extensible markup language format.

The proposed utility focuses on predicting current and future product formation to track the reaction progress. Nuclear magnetic resonance (NMR) and chromatography are two methods commonly used for monitoring reaction progress. To be quantitative, UV chromatography requires calibration of the UV detector with the analyte. This can be a challenge because authentic samples of the analyte may not be available. Alternative detection methods such as evaporative light scattering detection and chemiluminescent nitrogen detection are less accurate, but do not need calibration with the sample.[110,111] NMR provides quantitative results with use of an

internal standard but can be more challenging to interpret and may require more extensive sample preparation and interpretation. In hand-performed reactions, these methods take time for the chemist to perform and valuable machine time. Sample preparation means these methods are limited to being applied during chemists' worktime, whereas reactions often run overnight or over non-workdays.

Once enough data have been collected for a reaction, sensors could be employed to send data to a trained model which can monitor the product formation and predict the future product formation. This would inform the chemist when the reaction is nearing completion or if the reaction has stagnated. Sensor choice can be tailored to the specific chemistry. For example, pH sensors could be used in a pH-dependent reaction or colour data could be used if a colour change occurs in the reaction. This can be seen as a step towards automating, quantifying, and recording the data from some of the simpler tasks a human chemist does to understand their reaction. A litmus paper pH test is replaced by a quantitative pH sensor with a time-series and the same for a colour change, qualitatively observable by eye but can be quantitatively defined by red, green, and blue (RGB) colour values.[112,113] Sensors also offer the possibility for monitoring aspects which cannot be monitored easily in traditional ways. Data which can typically be harder to discern includes activity status of catalyst or reagents which may have poor ionisation, high reactivity, or not show under UV and common thin-layer chromatography stains.

Chemists can use a variety of methods and intuition to analyse the reaction mixture to predict progression at the current instant in time or the future progression. For example, it may be determined a reaction has plateaued if two measurements in

succession show no change in the reaction profile. Alternatively, a chemist may have experience with the reaction and develop an empirically based intuition of when the reaction has ended. For example, a reaction may typically reach completion at a variable conversion but within the same timeframe. The research here aims to test the suitability of machine learning for this objective.

Machine learning for yield prediction using the reaction scheme and conditions is an approach which has been implemented by others.[114,115] This research has been applied to palladium catalysed reactions, including Suzuki and Buchwald-Hartwig cross-coupling reactions.[116] However, the methodology struggled when applied to patent data which had too much inconsistency for accurate yield prediction.[115] Two issues with chemical reaction data that make predictive tasks challenging are the sparsity of the data within chemical space, and a lack of reported failed experiments.[117] This approach is complementary to the one we developed in this research as both have different aims. Our strategy aims to tackle the variability in yield that can be seen for a specific reaction. This can be a non-trivial problem for reactions that suffer from poor reproducibility. The method we have developed treats each repeat of a reaction as a single instance that can be distinguished from other instances by differences in the time-series data. Our hypothesis is that patterns within these differences can be exploited by a suitably trained machine learning model, thereby allowing accurate predictions of product formation.

The reaction studied in this work is a Buchwald-Hartwig cross-coupling reaction between benzophenone hydrazone and 4-chlorotoluene. Two experienced chemists at a contract research organisation carried out 15 repeats of this reaction and used

the hardware described earlier, provided by DeepMatter to generate time-series data for each repeat. The dataset will be processed to a suitable format for training and testing machine learning models to predict the current and future product formation. A series of machine learning algorithms will be used to evaluate the suitability of different models for the predictive task. Product formation is a continuous variable; hence, regression models are of interest including linear and polynomial regression models, decision tree models, and neural networks. The main limitation of the models being developed is they will only relate to this dataset, and therefore, only this specific reaction. However, extending this work to include larger datasets of multiple related reactions would be worth exploring in future work.

3.2 Methods

3.2.1 Data

We begin by describing the experiments that were performed to generate the raw data that forms the basis of our study. A DeviceX™ reaction probe and a Vernier thermometer were inserted into the reaction vessel for each reaction (computer-rendered image shown in Figure 3.1). The DeviceX™ contains a series of immersed sensors, exposed sensors, and a camera at the end for recording colour. Additionally, an environmental sensor, placed in the fume hood, measured the ambient temperature, humidity, and light levels. The data collected from this equipment were saved to a cloud storage and comprises several time-series.

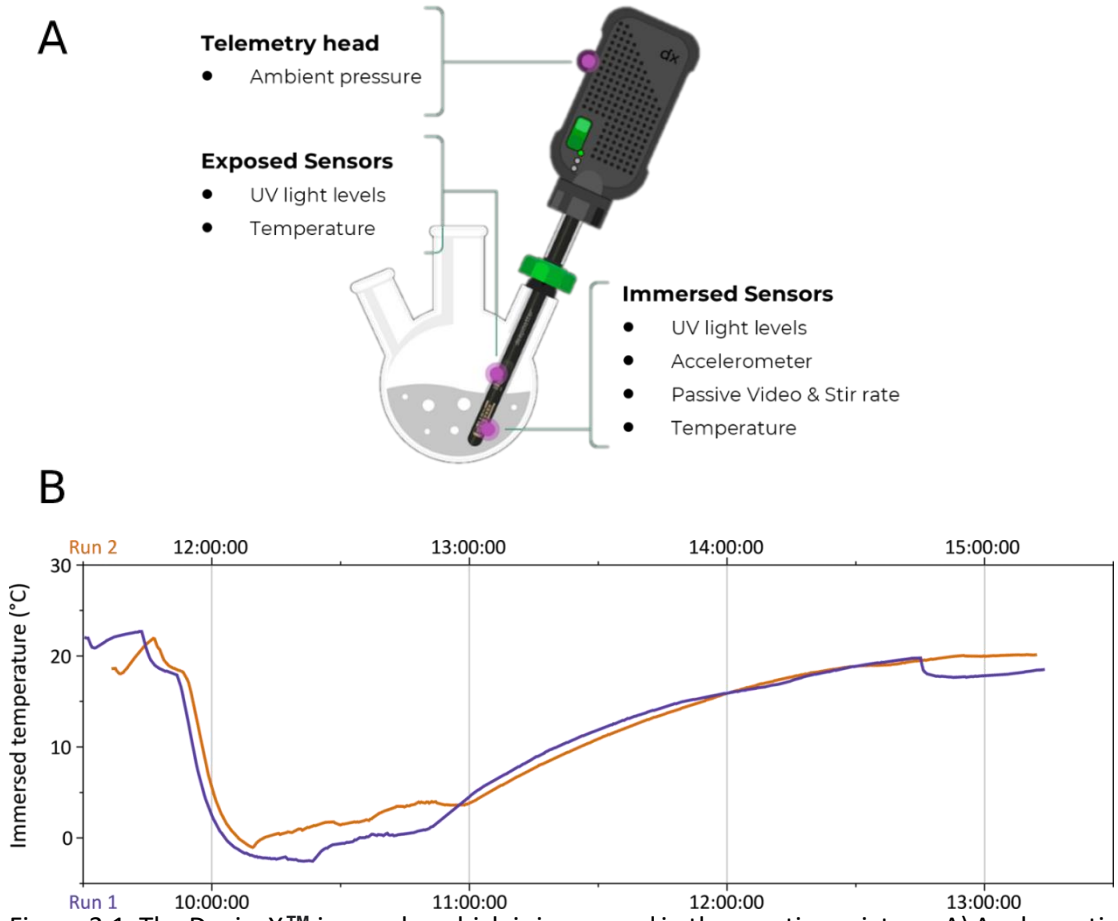


Figure 3.1: The DeviceX™ is a probe which is immersed in the reaction mixture. A) A schematic of this probe is shown. B) An example of temperature measurements for two discrete runs shown over a time-course. These runs are examples and not the experimental data used in this work.

The 15 reactions in the dataset are Buchwald-Hartwig coupling aminations. The reaction here is the cross-coupling coupling of benzophenone hydrazone and 4-chlorotoluene to form a new C-N bond (Figure 3.2).

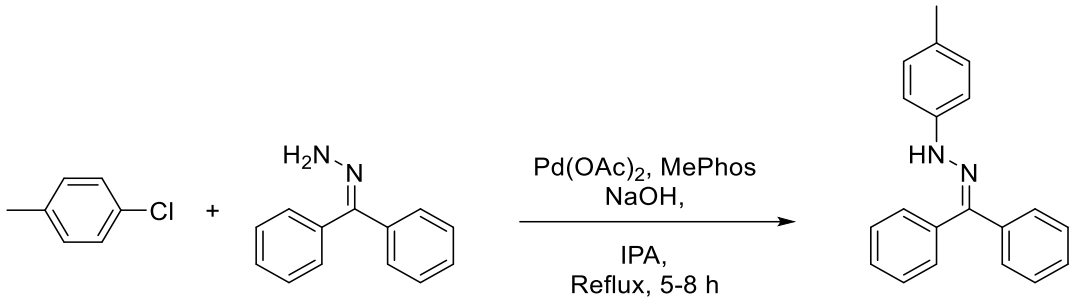


Figure 3.2: Buchwald reaction between 4-chlorotoluene and benzophenone hydrazone with a palladium acetate catalyst, MePhos ligand, sodium hydroxide base and either an isopropyl alcohol or tert-amyl alcohol solvent.

The experimental procedure was based on an optimised reaction obtained from the literature.[118] This exact literature route was repeated four times. Then the modified version shown in Figure 3.2 was used in the next 11 runs. The original route used *tert*-amyl alcohol (TAA) in place of isopropyl alcohol (IPA) and only ran for two hours. One difference between the solvents is that IPA's boiling point is 82.5 °C compared to 102 °C for TAA. Other variations between reactions included: the version of DeviceX™ probe which was used and what sensor data were recorded, the type of thermometer used and the rate of hydrazone addition. All of these are variations are displayed in Table A1. The first four experiments (001 to 011) and 024 were excluded, because the reaction probe used in these did not collect colour data (Figure A2). This left ten reactions to use for modelling and evaluation of the models. The first eight runs ran for a duration of five hours and the last two were left for longer (eight hours) as there was evidence that the reaction was still progressing.

The sensor data were collected by four different instruments. The reaction probe measured UV A and UV B wavelengths, stir rate, temperature, and pressure. The system extracted the average RGB components of the images captured by the

submersed camera. The environmental sensor measured light, humidity, temperature, and pressure. The Vernier thermometer provided a more accurate measure of temperature than the reaction probe (±1 °C versus ±3 °C). These temperature data was therefore used in preference to the temperature data from the probe.

The dataset also included liquid chromatography-mass spectrometry (LC-MS) data collected at 30-minute intervals to determine the conversion of the reaction. The peaks were assigned by their molecular weights. Specifically, these were benzophenone hydrazone (starting material), the product, and an impurity from hydrazone reacting with IPA which was identified by ^{1}H NMR (Figure A1). All the monitored reaction components contain the highly conjugated phenylhydrazone group. Therefore, it was expected their UV absorbance coefficients would be similar enough to allow the percentage area of the peak integrals to be directly compared and used for monitoring reaction progression.

The data were processed by changing the timestamps in coordinated universal time to time in seconds relative to the start of the reaction to give a standardised start time. Datapoints were kept if they fell in the window between the first and final LC-MS measurement. The sensor data were down-sampled to every ten seconds and the LC-MS outcome data were up-sampled by linear interpolation also to every ten seconds.

3.2.2 Models

The problem was approached as a regression problem with a continuous spread of outputs representing product formation. Many widely used time-series specific

approaches, such as autoregressive integrated moving average, were unsuitable for this problem as they require the time series to be stationary, which is often done by removing trends or seasonality. In contrast, this task involves predicting a non-stationary output, and the increase over time is meaningful and should not be removed. However, the approaches we use, such as recurrent neural networks, are suitable for application to time-series and do not assume a stationary output.

Due to the time-series nature of the data, cumulative and change values between measures were calculated. Cumulative values were calculated by successive addition of each time instance of a feature. Change values were calculated by measuring the difference between the previous and current time instance of a feature. Power transformations were applied to the features and to the cumulative features to increase the linearity of their relationship with product formation.

Models were made using linear regression, cubic polynomial regression, random forest, gradient boosting regression, and LSTM neural networks. These approaches include parametric and non-parametric techniques; whilst not an exhaustive set, they provide a reasonable coverage and variety of methods. The linear regression, random forest, and gradient boosting regression models were all implemented in Python using conventional sci-kit learn libraries. The cubic polynomial regression model was constructed using a custom function in Python and was considered because the product formation curve appeared from visual inspection to be describable by a cubic polynomial function. Splines could have been used as a more flexible alternative to polynomial regression. While this work did not explore them, they could be considered in future studies. The decision tree models each used 1000 trees and the

same random state was used. The LSTM neural networks were made using the Python package Keras. Two LSTMs were constructed. The first LSTM predicts the current product. A second LSTM was constructed to predict future product formation. Both LSTMs used a sliding window of data to make predictions. LSTMs are often suitable for sequential data.[63,119] Kinetic models were considered but were not chosen due to the lack of accurate product formation data in the early stages of the reaction. With the first LC-MS being performed after 30 minutes, the interpolated product formation values would have been strongly relied upon and may have resulted in inaccuracies when applied to kinetic modelling. This would be a significant area of interest for future work.

A sequential LSTM model was used and its hyperparameters are shown in Table 3.1. The model had three LSTM layers with 50 nodes with the first two using a ReLu activation function and the third layer using a sigmoid activation function. A dense layer with 50 nodes was then used and finally a dropout layer which randomly dropped 20% of input units to prevent overfitting. Additionally, if the model failed to converge during training, the training was repeated using a loop based on the loss value obtained in training.

Table 3.1: The chosen LSTM hyperparameters.

Hyperparameter	Value
Batch size	128
Optimiser	Adam
Loss	Mean Squared Error
Epochs	20

Features of interest were identified by manually calculating Pearson correlation coefficients with the product and by visual inspection. Four datasets were generated

based on manual analysis. These were denoted as *all features, green cumulative, Yeo-Johnson transformed green cumulative* and *non-colour features.*

To evaluate the predictive performance of the models, the data was divided into a training and test set, according to run rather than aggregated samples. Due to the limited number of runs within the dataset, the test set consisted of a single run and all datapoints from a run were kept together within the same partition to prevent leakage from the training set into the test set which would give an overly optimistic prediction. UMAP was used in conjunction with K-means clustering algorithm to visualise and group similar runs together.[65]

3.3 Results & Discussion

3.3.1 Data Analysis

The sensor data were averaged across the runs. From these averages the Pearson correlation coefficients, *r,* between pairs of features and between a feature and product formation were calculated (Figure 3.3). The coefficients between a feature and product formation were used as a starting point to determine the utility of a feature in modelling.

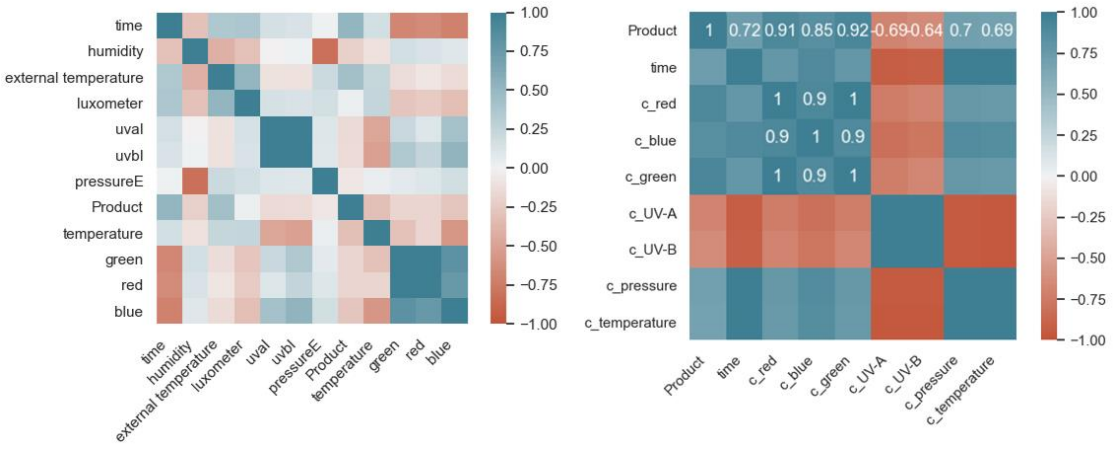


Figure 3.3: The left panel shows the Pearson correlation values for the unmodified sensor data. The right panel shows the Pearson correlation values for cumulative sensor values, where c_features refers to cumulative features.

To contextualise the coefficients, most cumulative features were expected to show some correlation to product, as if the feature is always positive or negative, both product and the cumulative feature will continually increase in value until product formation has stopped. The features most correlated with product formation are the cumulative colour features, particularly green and red, and a more negative correlation for blue. From the average values across the runs of these three features for the first five hours of the reaction, it can be observed that green and red steeply decrease over the course of a reaction and blue shows a more shallow decrease (Figure 3.4). A hypothesis for this is the reaction tends to black (where RGB would be 0,0,0) as the palladium catalyst is lost from the reaction and is reduced to palladium (0) black from the activated palladium (II) in the palladium acetate and the palladium ligand complex. This fits with the chemists' observations of the initial mixture being described as cream coloured but turning black or darker brown later.

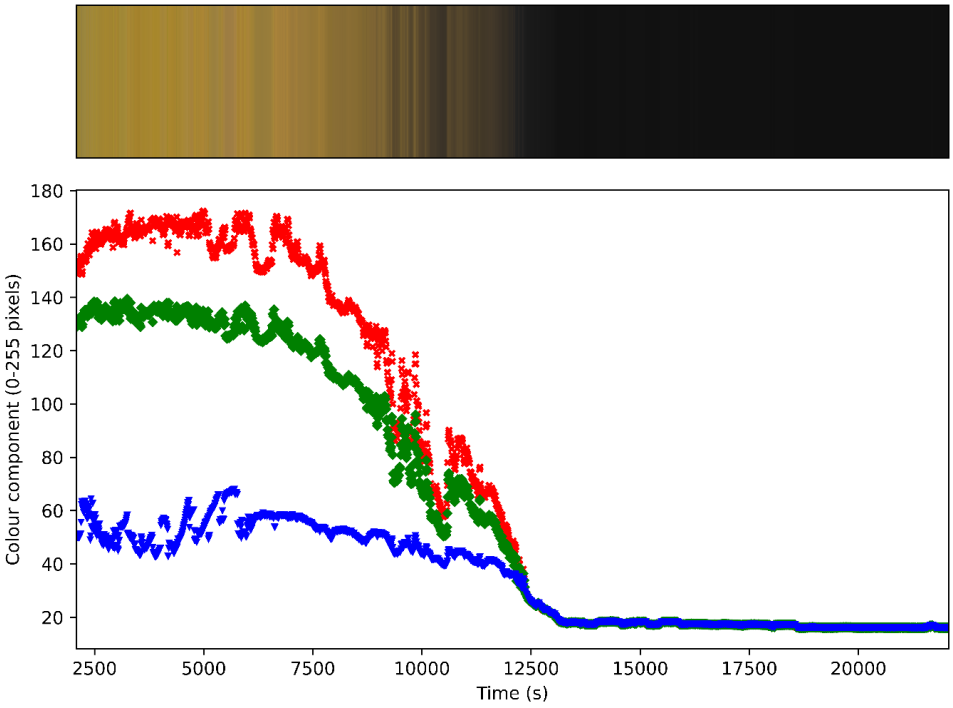


Figure 3.4: Variation over time of the recorded colour pixel component in the reaction mixture for example run Crocus-020. The top panel shows the calculated colour from the combined RGB values, and the bottom panel shows the individual RGB values. Each colour represents the colour shown, the crosses are red, diamonds green and downward arrows blue. Data are shown without error bars for clarity.

3.3.2 Experiment Design

The data was split into training and test sets using a 9:1 ratio, keeping entire reaction runs intact. This resulted in using 9 reactions in the training set and 1 in the test set. This prevented data leakage between reaction runs and more accurately simulated a real-world scenario, where the model would encounter completely unseen test runs. The cross-validation architecture used was based on all possible splits; evaluations of the predictions are reported as an average across all the possible splits. Initially, a linear regression model was developed on the cumulative green values from nine runs to predict the product for the tenth run. Assessing the model using cross-fold validation provides a robust estimate, in data-limited situations, of the accuracy of the

model and its ability to generalise by testing its predictions on all the datapoints. Models were evaluated by measuring the MAE of the predictions. All product yields are reported as percentage points Thus, despite MAE being reported with units of % it is an absolute and not a relative error.

Linear regression is a natural starting point. However, it was observed the relationship between the cumulative green and product formation was not linear. To remedy this different power transformations were applied, and they were evaluated by the improvement in accuracy of the transformed features in a linear regression model compared to the non-transformed feature. After evaluating the different power transformations, a Yeo-Johnson transformation was selected as the most promising. The purpose of a Yeo-Johnson transformation is to reduce the skewness and make the distribution of the data more Gaussian.[120] Applying the Yeo-Johnson transformation to cumulative green yielded a more Gaussian-like distribution, as can be observed in Figure 3.5. An unexplored alternative would be the use of machine learning-based transformations, which could be compared to power transformations in terms of their ability to normalise the data and their impact on model performance.

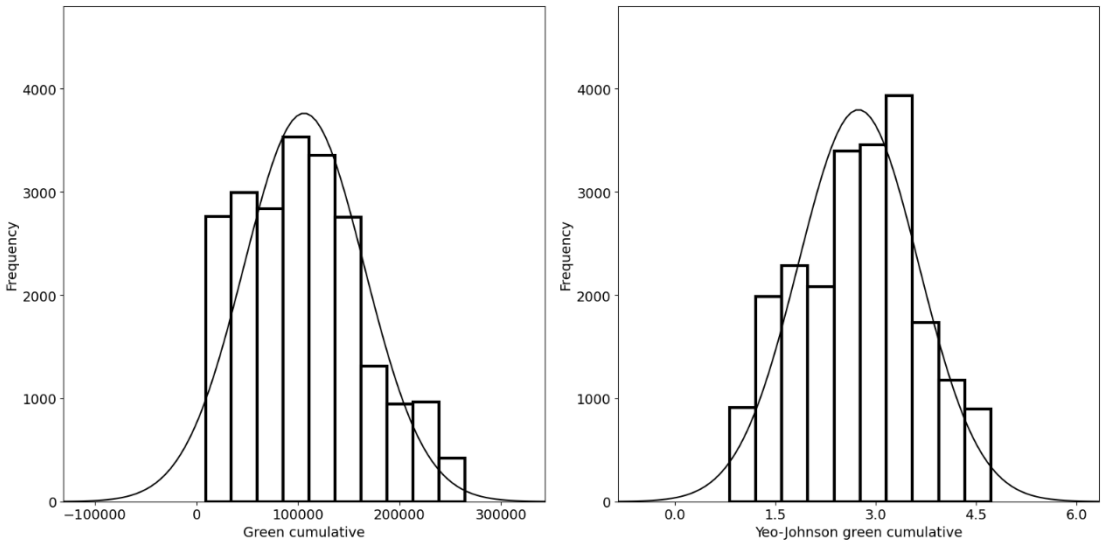


Figure 3.5: The left panel is the distribution of green cumulative values, and the right panel is the distribution after applying a Yeo-Johnson transformation. A normal distribution is plotted over both histograms.

Other models explored included polynomial regression. The polynomial used was a cubic function. This model was considered suitable due to the non-linear relationship between cumulative green and product. Random forest and gradient boosting regression, both based on decision tree models, were also explored.

The four datasets, each containing different features, (*all features, green cumulative, Yeo-Johnson transformed green cumulative* and *non-colour features)* were developed to investigate the effects and significance of the power transform, colour, and other sensor data. Use of *all features* would allow an evaluation of predictions without prior manual identification of features. *Green cumulative* and *Yeo-Johnson transformed green cumulative* compare the effect of the power transform and to what extent a single feature can provide accurate predictions. The *non-colour* dataset investigates how predictive the non-colour data is.

As a baseline, we calculated the MAE of predictions obtained from just the mean product values of the training runs. This gave an MAE of 7.1%. The four datasets were combined with the four models to produce all possible combinations. The results of these are shown in Figure 3.6. Analysis of variance (ANOVA) and t-tests and were used to determine statistical significance. The data are given in Tables A2, A3 and A4 of the Appendix. The Yeo-Johnson transformation improved results when using a linear regression model, reducing the MAE from 4.2% to 3.8%, but the standard error (SE) was unchanged at 0.5% and the improvement was determined not statistically significant by a t-test (*p*-value 0.55). The results for each algorithm with the *no colour* dataset (MAE 7.0, 7.1 and 8.2%) were all comparable to or worse than the baseline results (7.1%). This suggests the Pearson correlation coefficients identified the useful features in this context, and there is little value in the non-colour data in isolation. ANOVA showed there was a statistically significant difference in prediction accuracy between the methodologies. T-tests showed that eight of the 13 methodologies had a significant improvement in predictive accuracy compared to the baseline: these eight methodologies were all those with a MAE equal to or less than 4.4%. ANOVA indicated that there was no significant difference in the predictive accuracy between those eight. Results of interest include the lowest MAE obtained by polynomial regression with cumulative green (MAE 3.6%, SE 0.6%) and the only multivariate method which used all features with a gradient boosting regression algorithm (MAE 3.9%, SE 0.4%). Using all features has the advantage of obviating the need for prior feature selection. This does not interfere with the interpretability of the model either, as feature importance values can be extracted from the decision tree models. Feature

importance using all features in a gradient boosting regression model (Figure A3) further confirms colour to be the most informative feature and suggests that it is likely all four models identify similar patterns in the relationship between cumulative green and product formation which had the highest correlation with product formation as shown in Figure 3.3.

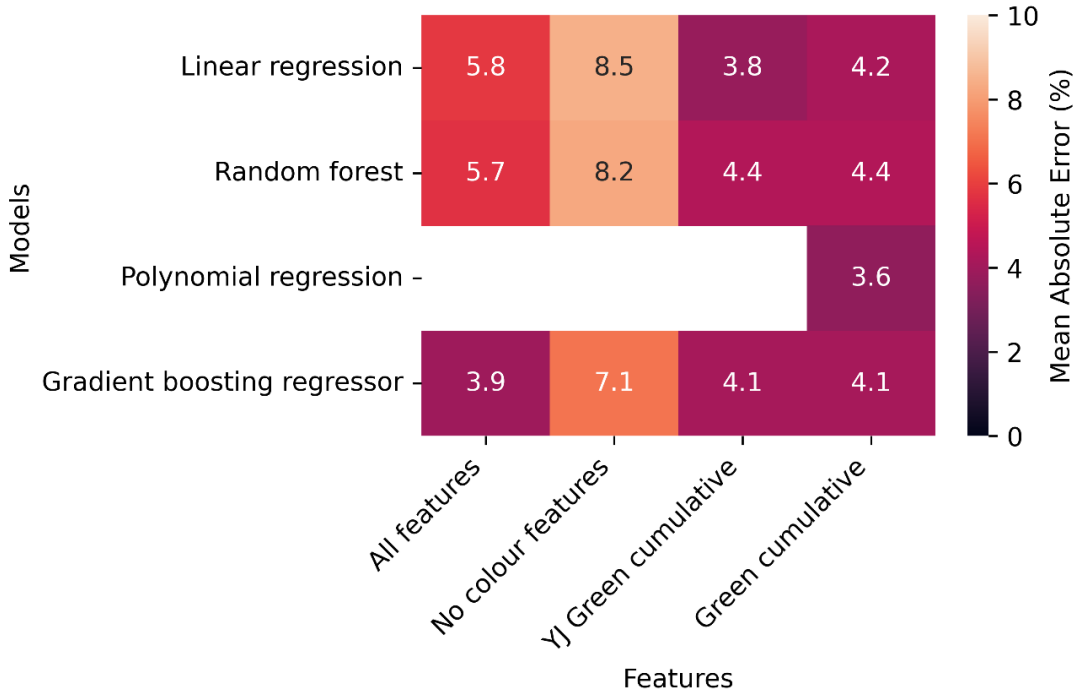


Figure 3.6: Comparison of the mean absolute errors (indicated by the colour bar) for the different combinations of features (x axis) and models (y axis) obtained from the 10-fold cross-validation using a 9:1 train: test split. YJ stands for Yeo-Johnson. The polynomial regression model developed was not suitable for all feature sets, and hence was only used in conjunction with green cumulative.

None of the above models is time-based, and no explicit concept of time has been encoded in the associated datasets. Given that the input data are a time-series signal, this raises the question of whether a time-series model would be more accurate.

Recurrent neural networks, such as LSTMs, are well suited to time-series problems and other sequential tasks such as natural language processing and have been

successfully applied in other areas of chemistry.[121,122] The LSTM model constructed uses a sliding window approach, whereby a window of fixed size is formed over the data, and this window slides over the data to capture different portions of it. This method ensures the volume of data used in the model remains consistent. The method was employed to use the previous 20 minutes of sensor data. Changing to an LSTM model using *all features*, improved prediction accuracy with a MAE of 1.2%; this is compared to a best of 3.6% for the non-neural-network models. These results can be contextualised by comparison to the yield range in addition to the earlier described baseline. The observed yield was between 7.5 and 38.1%, giving a range of 30.6%. Therefore, a predictive accuracy with a MAE of 1.2% was considered useful in this context.

Following on from the promising results obtained for the instantaneous predictions from the LSTM, a second LSTM was designed for the more ambitious aim for predicting the future product. The second model uses sensor data and predictions from the first model between times $t_1 - y$ and t_1, where t_1 is the current time and y is a fixed time interval, to predict product at time $t_1 + z$ where z is a variable time interval.

To balance ambition (i.e., the extent of the forward time interval), accuracy, and computational cost, y = 2 hours, and z = 0, 30, 60, or 120 minutes. These time interval values can be put into context by comparison to reaction duration which was between five and eight hours (300-480 minutes). The expected behaviour is the larger the value of z, the harder it is to predict product formation. A range of values were selected for z to allow an evaluation of the relationship between the MAE and z. Because of the greater challenge associated facing the second model, y was assigned a higher value

than *x*. This meant sensor data over a longer duration of time was used in the second model. In this task, it is more important to be able to see the larger context of the reaction progress. Having two models was advantageous, as it allowed the assessment of what would have been the intermediate output when $z = 0$.

Since the LSTM gave a significant improvement, this model was chosen for the more ambitious target of predicting future product formation. Because current product can be predicted relatively accurately (MAE of 1.2%), a time-series of predicted product values was obtained. These predicted product values and the sensor data are then fed into a second model to predict the product conversion in the future. The results from the cross-validation of the two LSTM models are shown in Figure 3.7. There is a large (but unsurprising) increase in error when predicting just 30 minutes ahead. For further scrutiny of the model, an additional cross-validation architecture was employed, based on all 45 possible splits from an 8:2 test: train split. The MAEs for these were 0.95, 5.3, 5.8, and 6.3% for predictions 0, 30, 60, and 120 minutes ahead, respectively (Figure A4).

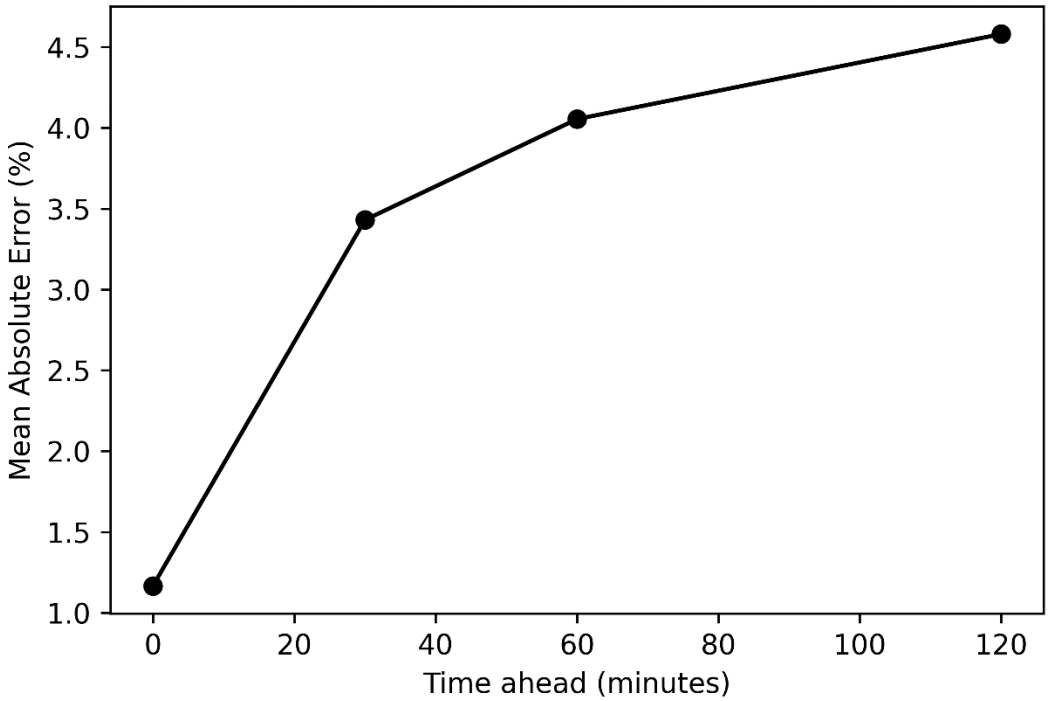


Figure 3.7: The mean absolute error for product formation predictions from an LSTM model predicting 0, 30, 60 or 120 minutes ahead using a 10-fold cross-validation with a 9:1 train: test split.

3.3.3 Use Cases

3.3.3.1 Use Case 1

To assess the model, a series of real-world scenarios were identified. One scenario was to split the data, so the last chronological run was the test data, and the previous runs were the training data. This corresponds to the training data that would exist whilst the final reaction was being performed. The trend of the results from this use case closely align with those from cross-validation (Figure A4). Compared to the cross-validation, the model performed slightly better for all values of *z* greater than zero.

3.3.3.2 Use Case 2

Another example use-case would be for predicting when product formation has stagnated in a reaction at a lower conversion than expected. The goal in this scenario would be to detect a failed reaction at an earlier stage, and the live data in combination with predictions would give an indication of this to a chemist. This can be rationalised as, if the colour change occurs to indicate the catalyst has been consumed, no more product will be formed. Reactions 16 and 17 were both low yielding and the chemists performing these reactions observed a potential exposure of the hydrazone to atmospheric moisture.

An 8:2 train: test split was used and reactions 17 and 25 were used as the test runs. The rationale for using this split is twofold. The first and primary aim is to demonstrate that if a failed reaction, 16, is included in the training set, the model can identify future failed runs, such as 17, early. The second aim is keep run 25 in the test set to ensure the model is still capable of generalising and accurately predicting successful runs.

A weakness identified in the model was the imbalanced dataset due to the training data containing only one failed reaction. To address this, oversampling was implemented. A condensed version of the reaction data was interpreted by a SMOGN (Synthetic Minority Over-Sampling Technique for Regression with Gaussian Noise) algorithm.[123] The SMOGN algorithm can be particularly useful when the values in the interest of predicting are rare or uncommon, such as the failed reactions in this scenario. The algorithm identifies under-represented reaction runs from the condensed dataframe and returns a new dataframe with oversampling of under-represented data and undersampling of over-represented data. The algorithm

performed as expected, with the newly returned dataframe oversampling the only low yielding reaction in the training set, reaction 16. The results for this can be seen in Figure 3.8. The predictions for run 25 are worse compared to those shown in Figure A4; however, they are within an acceptable range and the predictions for run 17 demonstrate an ability to predict a low-yielding reaction.

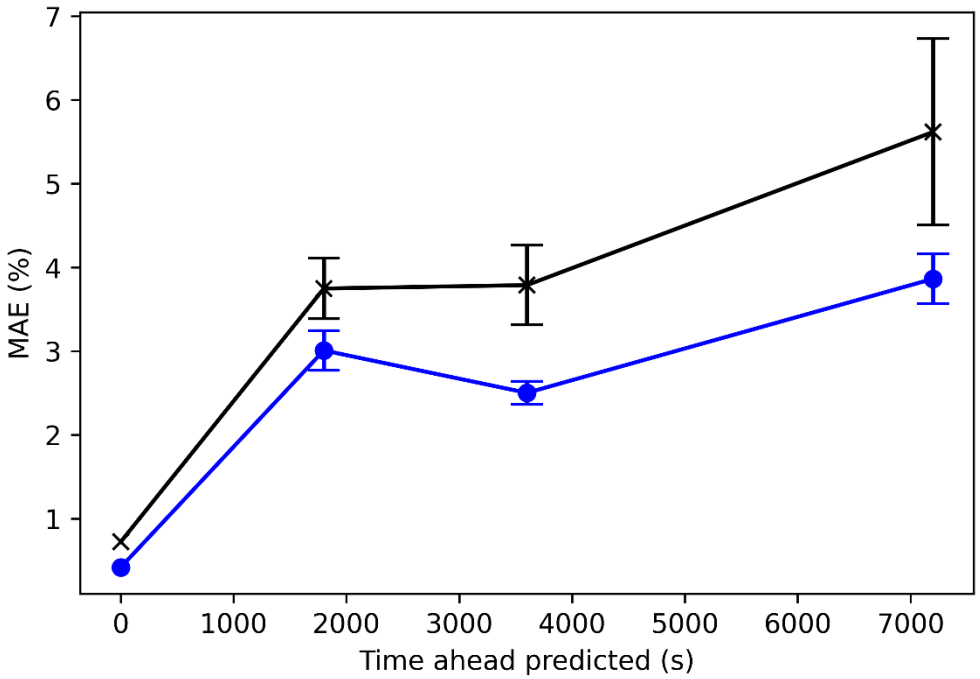


Figure 3.8: Blue circles show mean error in predictions for reaction 17 and black crosses for reaction 25. The standard error bars are calculated from ten repeats.

3.3.3.3 Use Case 3

A third use-case is to test the model chronologically and explore the relationship between the size of the training data and prediction accuracy. To test the model chronologically, this would mean training only with runs that come chronologically before the test run. This is a useful test to do, because it closely mimics the real-life scenario of a developing dataset wherein the chemist may be learning subtleties of the problem as they proceed. The model produced reasonable results when $z = 0$. However, in early runs with little training data, the model was unable to make

accurate predictions when $z > 0$. Interestingly, the relationship between the size of the training data and prediction accuracy was inconsistent with the MAE increasing as the training size increases before decreasing again. The reason for this could be due to the differences between runs, with some being more challenging to predict than others. This is also seen in the results from cross-validation. Figure 3.9 compares the chronological and cross-validation results. The cross-validation results show the error at time zero for each run with a training set size of nine other runs, whereas the chronological results show the error at time zero for each run with a training set of an ascending size only including runs which were performed earlier chronologically. In both scenarios runs 22 and 23 have the greatest MAE. This suggests that these runs are more challenging to predict. Some runs perform better in the chronological results, despite having a smaller training dataset. This could be due to the runs in the dataset being more similar. This can be observed with run 17 as it is similar to run 16, since both had poor product formation.

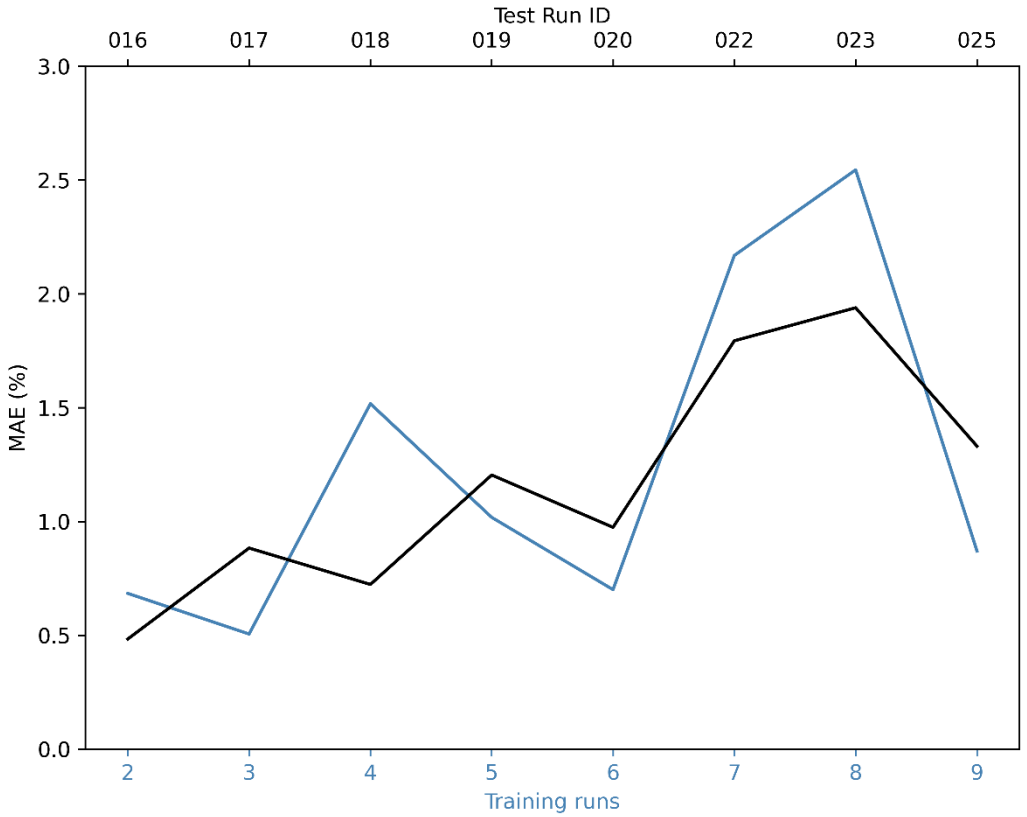


Figure 3.9: The blue line is the mean absolute error for predictions of a model trained only on runs which were chronologically before the test run, the number of which is shown on the bottom x axis. The black line is the mean absolute error for predictions from the cross-validation where a model was developed to predict the run by training on *all* other runs, regardless of chronological ordering.

An alternative approach to investigate the relationship between training dataset size without the additional variable of different test runs, was to keep run 25 as the test run and increase the training size in chronological order (Figure 3.10). This gives a more expected relationship with smaller increases in MAE as the training size increases. The other variable that may affect the relationship here is the different runs added to the training set and how useful they are for learning how to predict the product formation in run 25.

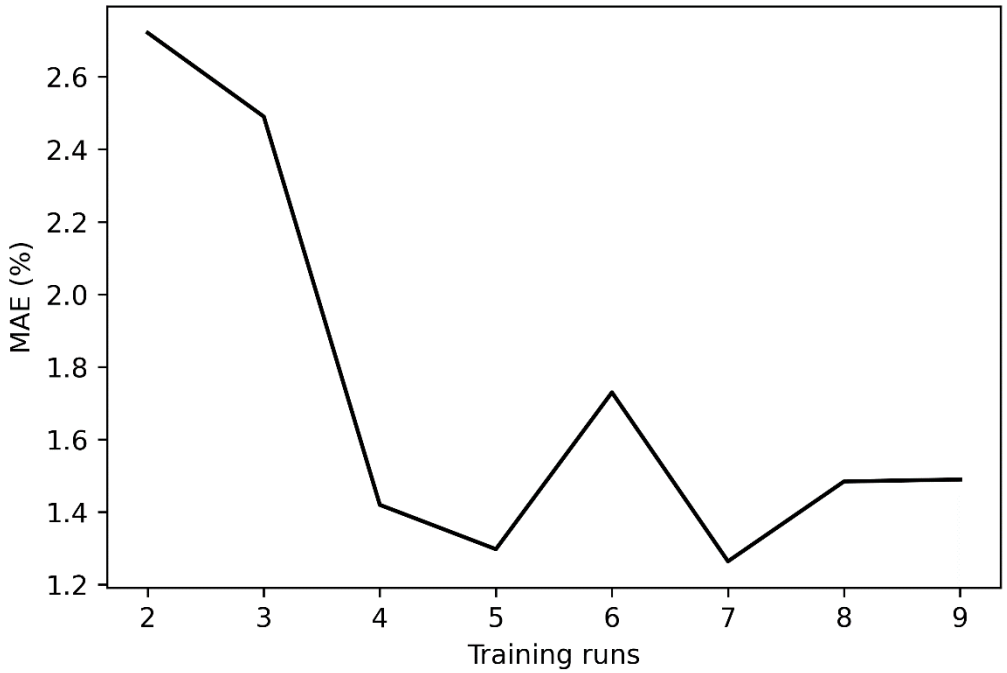


Figure 3.10: The relationship between the number of runs in the training data and the accuracy of predictions for product formation in run 25.

It was hypothesised that individual runs could be tracked within the 2D projection and similar runs could be near one another To investigate this the data were projected onto a two-dimensional manifold using UMAP and a K-means clustering algorithm applied to obtain distinct groups. After trying different values of *k*, *k=3* was used and the three clusters obtained from this can be observed in Figure 3.11. The dark blue cluster in the top right correlates with low product formation. Runs 16 and 17 (both ending with below 20% product formation) are mostly in this cluster. In comparison, run 25 initially occupies space in the top right, but as product formation increases, it has more datapoints in the middle and bottom left cluster. (Figure A5). This spread of data suggests the model may struggle to generalise, as runs all occupy different

spaces, and hence may struggle to predict runs further from the space of the training data.

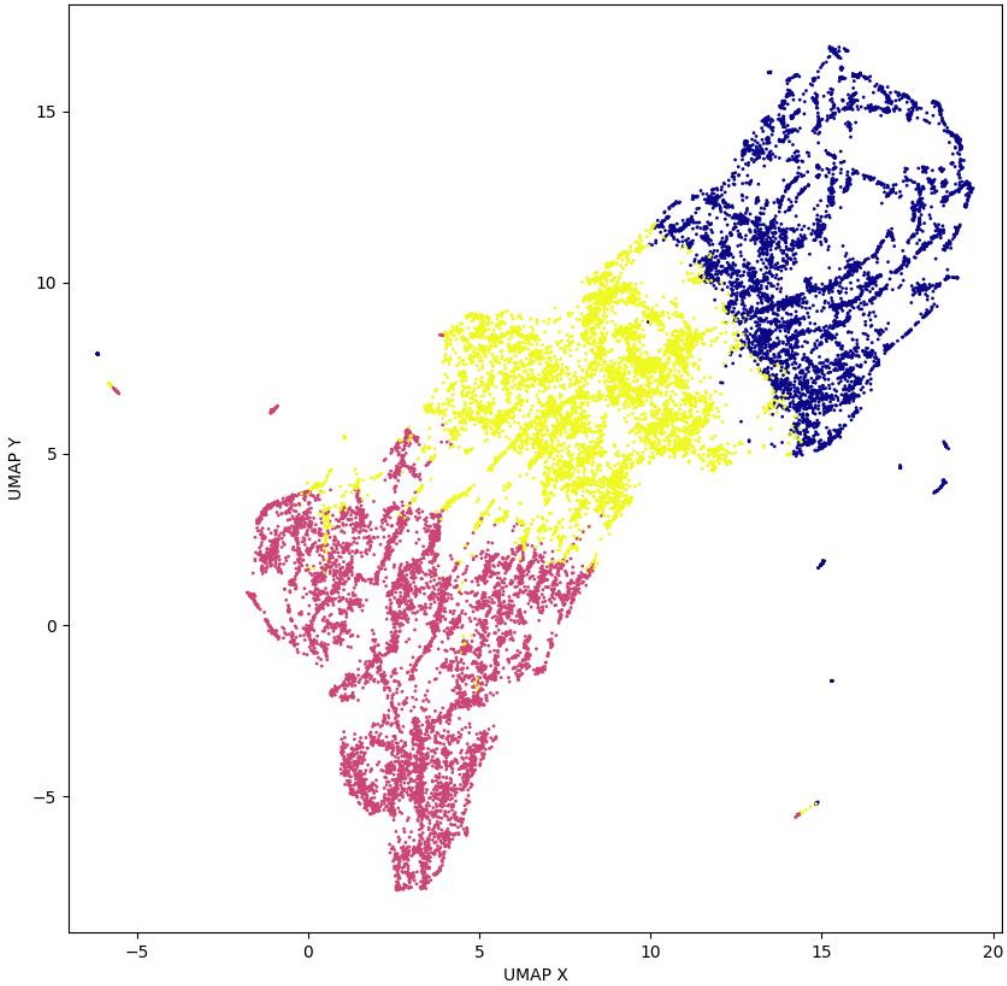


Figure 3.11: *k*=3 clustering of a UMAP projection of the reaction data for all runs used in machine learning models.

In a larger dataset, it could be envisaged that this could allow similar runs or reactions to be identified. This could allow the use of tailored models which would seek to place emphasis within the training data on runs identified as similar to the run of interest and this could create models which could give more accurate results. A small-scale example of this can be observed earlier in the accurate results of run 17 in the

chronological use-case, which included also failed run 16 in the small volume of training data.

3.4 Conclusion

Models to predict the current and future product formation from reaction sensor data collected by an *in-situ* reaction probe were constructed and evaluated. In this work we have demonstrated different use cases for these models. By using cross-validation and three realistic use cases, the predictive accuracies were assessed.

The reaction used here was suitable due to the reliable product formation curve shape and profile of the reaction. Mild changes between runs, such as the rate of hydrazine addition, did not have a noticeable impact on prediction. However, large differences between runs, such as in concentration or temperature, could affect the rate of reaction. Future work will investigate how more significant changes affect the accuracy of the model. To assess further the potential and utility of using machine learning to predict product, more examples would need to be examined. This methodology could enable AI augmentation of reaction monitoring to assist synthetic chemists and facilitate a greater understanding of the reaction by identification of correlations between sensor features and reaction outcomes. Insights into the chemistry being performed could also be developed. For example, the correlation between cumulative green and product formation in this work provides a quantitative description of the colour change in the reaction.

As use of sensors in synthetic organic chemistry grows, more data will become available and allow greater insights into the chemistry. This will also permit a more

thorough investigation into the relationship between dataset size and accuracy. The models demonstrate that information recorded by specialised reaction probes can be exploited by a neural network for product prediction.

Data and software availability are provided in the appendix.

Chapter 4: Development of the AI4Green Electronic Laboratory Notebook for Green and Sustainable Chemistry

Abstract

Electronic laboratory notebooks can offer significant advantages over traditional paper lab notebooks, which are still widely used in academia. In this chapter, the development of the AI4Green ELN is presented. AI4Green offers core ELN functionality, which can support chemists' workflows by providing methods to complete molar calculations automatically, search reactions by structure, and organise reactions and users hierarchically, facilitating controlled data sharing within a research group. A compound database has been created using data from PubChem, which instantaneously gives users information on the compounds used in their reaction, including hazard codes. The ELN encourages good data practices by saving reactions in a digital format which is backed up in the cloud and by supporting multiple data export formats.

AI4Green encourages sustainable chemistry by automatically calculating green metrics and using colour coding to indicate the sustainability of the various aspects of a reaction. The app has built and integrated tools to support solvent substitution and retrosynthetic route planning. It is intended for further tools to be developed and integrated in the future and for further development of existing tools. It is anticipated

that the collection of new reaction data within the ELN may contribute towards the development of future tools.

4.1 Introduction

Recording and sharing data is essential for scientific progress, as it allows the transfer of knowledge. While ideas can be exchanged verbally, written communication provides a lasting and detailed record that reduces errors associated with human memory. Accurate data recording close to the event enables others who are not present to understand what occurred and provides a foundation for data sharing.

The methods for data recording and sharing have evolved over time. Writing is among the earliest and most robust methods, and this fundamental approach to knowledge preservation has independently emerged multiple times throughout human history; as early as 3300 BC, the Sumerians wrote on clay tablets. While these methods have evolved, constants remain, as chemists today still write in paper lab notebooks.

Data sharing is effectively a method of communication where the accuracy of the shared data depends on the reliability of the method used for the original recording and sharing. Data sharing requires transferring data from one person to another, often in different geographical places. As the printing press revolutionised the process of text duplication, the internet and computers have enabled near-instantaneous global data transfer in recent decades. This chapter discusses the opportunities, barriers, and practical realities of developing and using ELNs to digitise chemistry and how ELNs can aid data recording and sharing in modern scientific practices.

4.1.1 Project Outline and Motivations

The AI4Green project goals involve making an ELN. There is a gap for an ELN tailored for synthetic chemists in academia focused on encouraging sustainability. There were

then multiple motivations for making this ELN. The first was to collect reaction data, both positive and negative, in a structured format. Reaction data are a valuable resource as they can reduce duplication of synthetic effort if these data are made FAIR, and other chemists can find, access, and use them. Additionally, if the data are recorded in a structured format, they can be used as machine-readable data to train machine learning. Secondly, sustainability assessments should be integrated into the workflow to provide opportunities for improvement and simplify the process to avoid adding to the chemist's workload. Thirdly, there was a motivation to provide chemists with a "one-stop shop" of digital tools. By doing this, we can reduce the burden required for chemists to set these tools up themselves or register and sign into accounts on multiple websites. It is intended for these tools to include externally developed tools accessible via an application programming interface (API), internally hosted versions of these tools, with potential modifications to incorporate sustainability, and internally developed tools to help sustainability.

4.1.2 Digital Tool Development

The development of a scientific research software platform presents unique challenges. Initial requirements must be identified and translated into executable tasks that achieve the intended functionality. Flexibility and adaptability are crucial for the software to meet continuously evolving priorities. Every stage of development involves critical decisions on the architecture, framework, and tools used. Suboptimal choices can result in technical debt, which hinders future development and maintenance and may require significant reworking in the future.

Software development, like chemistry or data science, is a continuously evolving complex field where techniques, tools, and processes are continually refined. The inherent complexity of software engineering can be exemplified by the recurring concept of a software crisis. First discussed at the 1968 NATO Software Engineering Conference, it describes the observations that projects frequently exceed time and budget constraints, deliver poor final products, and require excessive maintenance.[124] More recently, the term has also been used to describe modern software development challenges.[125] These issues are often attributed to inadequate development processes and a growing gap between the rapid advancements in hardware capabilities and the expertise required to utilise them effectively.[124,125]

To achieve the aims in the project outline, the software requires research on sustainability and the metrics used to measure it, organic chemistry and its digitisation, and suitable methods of implementing this functionality in software and delivering it to users. The software must be engaging, intuitive, and practically suitable for chemists to adopt it. The methods required to achieve FAIR data standards must also be researched, including collecting, storing, and sharing data.

The development of the ELN lies at the interface of chemistry, software engineering, and data science. ELNs are complex software as they must handle specialist data and perform the relevant scientific operations on it. Health and safety are important considerations within chemistry. These requirements mean the software must perform to a high standard and be reliable to give users trust in safety features and data security.

4.1.3 The opportunities, the barriers, and the paper lab notebooks.

Chemists have traditionally used paper lab notebooks to record their experiments. Whilst these are incredibly flexible tools, paper notebooks have significant drawbacks. They are poor at preserving data long-term, and the data require manual digitisation to be shared, such as for publication. When the experimental procedures are published as part of the supplementary information, it is typically provided as unstructured text and, therefore, not machine-readable. Additionally, failed or low-yielding reaction data, often called “negative data”, are typically excluded from publications despite its value.[126] In today’s digitised world, ELNs can offer numerous advantages. To fully realise these potential advantages, ELN developers must incorporate features to enable them. Interoperability is crucial for data sharing but is not widely supported. While no accepted universal standard exists, efforts can be made to prioritise using open and widely used data formats over proprietary formats to aid data sharing. With an ELN, data can theoretically be shared with anyone, irrespective of location and proximity. Practically, this requires options and secure methods for different levels of sharing permissions. Searching can be quickly performed on the institutional or research group scale, which can build a trusted in-house knowledge base. Ultimately, ELNs can offer a modern alternative to paper lab notebooks, but they do not come without their challenges.

Barriers to ELN usage include cost, product lifetime, data security, and suitable functionality, and whilst time-savings may be made, these must be balanced with potential time costs. The costs incurred depend on whether the ELN is commercial and requires a subscription, open-source and requires hosting, or is freely accessible

online. Monthly subscriptions are considered SaaS (software as a service) and vary in price, typically dependent on factors and require a quote. An approximate value can be obtained from the Amazon Web Service (AWS) marketplace, where Revvity Signals costs $1,110 a year per user in a multi-tenant environment.[127] This is much more expensive than a paper lab notebook but comes with an ELN's advantages and minimal implementation time and effort. Alternatively, an open-source ELN can be hosted within an institution. This option requires more upfront cost and effort and continual server costs with potential risks of storing the data locally. It may also limit data sharing outside the institution and remote access to the ELN. An example is eLabFTW, which can be hosted for free as the code can be downloaded and the application configured to run on a server. Alternatively, and to provide a guide cost, eLabFTW offers hosting services for €4985 per year for up to 128 users.[128]

The lifetime of an ELN can be a concern as if the product stops being offered, the data may be trapped, and new costs may be associated with setting up a new service and moving data to this new service. An analysis found that the median lifetime of an ELN was seven years, and of 172 products, 96 were active, and 76 had been defunct.[129] Interoperability with another ELN can mitigate and minimise data transfer costs to a new service. But even so, the risk of a product becoming defunct is a concern. In comparison, the user can exercise more control over paper lab notebooks. However, their long-term storage can be challenging, and finding a specific book from seven years ago may be a significant challenge, and this would be amplified if searching for a specific reaction. A positive aspect of the paper lab notebook is the extent to which they are personal items and fully customisable to the user. However, a downside is

that it also contributes to difficulties in recovering data after the lab notebook's owner has left the institution, as other team members may be unfamiliar with the notebook's contents and style.

Data security is another potential concern. Some institutions may prohibit storing data on external servers, which can rule out some ELNs. Concerns over IP are strong within the drug discovery industry and may also affect the decision of contract research organisations (CROs) working on behalf of external clients when implementing an ELN. This can restrict the choice of ELN rather than prevent ELN implementation by either hosting on internal servers or using a trusted vendor with high levels of security on an external server. Provided they remain on site, paper lab notebooks can be seen as more secure as they are inaccessible by design, as is seen with the challenges in extracting data from them.

ELNs also offer different functionality. Some, such as eLabFTW, are general-purpose and can be used in any field but lack domain-specific functionality. Alternatively, ELNs, such as Chemotion, are designed for use within a specific domain, and typically, more extensive commercial solutions may offer strong support for multiple domains.[82,130] This requires the organisation to decide its priorities. For example, multi-disciplinary institutions may wish to select a single ELN and either lack features designed to support a discipline, pay more for a single ELN that supports multiple disciplines or choose multiple ELNs that may require additional work. Even within a single domain, not all workflows may be supported equally. Using organic chemistry as an example, many ELNs offer much more support for single reactions than HTE work. Paper is very flexible in comparison but comes with no supporting tools.

Due to these documented challenges, ELN selection can be daunting and time-consuming. Reflecting this is the development of tools such as ELN-finder to help organisations find suitable solutions.[131] Institutional and personal inertia and a 'don't fix it if it isn't broken' attitude towards replacing paper lab notebooks with electronic lab notebooks can all contribute to these barriers. Despite these barriers, ELNs have become more widely adopted within industry. These can involve in-house implementations, especially for larger companies, such as AstraZeneca, or using a commercial ELN, such as Signals by Revvity (previously Perkin Elmer). The development of ELNs by many large pharmaceutical companies contrasts with the low ELN usage in academia, with a 2017 survey by BioSistemika where only 7% of academic respondents used an ELN in their daily lab routine.[14] The most cited barrier in this survey was cost, followed by concerns about changes to working habits, the time required for implementation, and storing data in the cloud. 72% of respondents expressed an interest in using ELNs, and only 21% said they were not interested and preferred paper lab notebooks. A 2023 survey by the Physical Science Data Infrastructure (PSDI) on whether paper or electronic devices were used for recording different aspects of an experiment found that paper-based methods were still frequently used in the lab.[132] Research in this area suggests there is still work to get ELNs in the hands of the researchers and barriers to be overcome.

A 2022 review of ELNs found there were far more commercial than open-source options, 147 compared to 24.[129] This may contribute to the more significant barriers faced in academia, where funds are more limited. Related to costs are hardware barriers where the PSDI survey found paper was used more in the lab stage of an

experiment than the later analysis stage, which could be due to the challenges and costs of having electronic devices in hazardous labs that may be lacking in space.[132] Additionally, academic research groups are more decentralised than industry and have less of a data-sharing culture between groups.[133]

The lack of uptake of ELNs within academia suggests that these barriers are significant, and multiple approaches from different stakeholders are required to overcome them. Labs need to be equipped for the digitised world with suitable computational hardware to support ELN usage. ELNs need to better meet the needs of their users in terms of access, usability, and cost. Finally, work must be done to link the developers and distributors of ELNs to the lab chemists to encourage and support the uptake of unfamiliar software. Within the AI4Green project, all three of these approaches have been taken. As part of the project, the AI4Green ELN has been developed to meet the user's needs, hardware has been bought and installed in selected labs to enable ELN usage, and users have been given training and support in their use to increase the success of their adoption of the AI4Green ELN. The focus of this chapter will be the development of the AI4Green ELN to meet the user's needs, with some mentions of the work done to support and encourage successful uptake through providing training resources.

4.2 Methods

Team Acknowledgement

The development of the application has been a collaborative effort within the AI4Green team. Some features have been implemented independently by me, and

others have been in strong collaboration or done entirely by others. Change management to the code base is controlled by pull requests, the process where code changes. Each pull request requires two approvers. Therefore, all of my changes have been checked, with changes requested when necessary and approved by others on the project. The inverse is also true; I have approved pull requests made by others and requested changes when necessary.

When I joined the project, the web application was in the prototyping stage and only ran on a local server. This early version contained a more basic version of the reaction constructor with the integrated CHEM21 sustainability metrics. The application did not have a database. It was not possible to save reactions or use compounds not in PubChem, and there was no user hierarchy or project management system. An API was used to get compound data from PubChem, which is much slower than querying a database.

This prior development was completed by Dr Ivan Derbenev and provided the foundation upon which the subsequent development was built.

The user account system was built by Dr Ivan Derbenev (4.3.1.2).

The initial workgroup and workbook structure was developed by Dr Christopher Handley (4.3.2)

The most recent methods for adding users to a workgroup by email or QR code were implemented by Dr Joe Heeley (4.3.2).

The highlighting of required cells of the reaction table in red was completed by Dr Samuel Boobier (4.3.3)

The solvent substitution tools, flashcards, and the interactive PCA were developed by Dr Samuel Boobier and Dr Joe Heeley.[38,134] Therefore, these tools are mentioned when describing the application, but their methods are not described.

The initial continuous integration/continuous deployment (CI/CD) pipeline was developed by Andy Rae in the University of Nottingham Digital Research Service (DRS).

4.2.1 Application Stack

When describing the technical details of an application, it can be helpful to describe it as a stack. The stack consists of different technologies that interface to form the application. The stack can be divided into frontend and backend. The frontend is the part the user sees and interacts with. The backend is the part the user does not see but handles most of the logic and data of the application. Data can be exchanged between the frontend (browser client) and the backend (server), enabling the server to send information to the user's browser and receive data from the client in return. The stack is illustrated schematically in Figure 4.2.

4.2.1.1 Backend

AI4Green has a Python backend which utilises the Flask framework for web applications. According to The Importance of Being Earnest (TIOBE) index, Python has become the most popular language in the world in recent years.[135] It is comparatively easy to learn, versatile for use in different domains, and memory secure. However, it is slower than other languages, such as C++ and Rust. Python also has excellent support for data science and scientific programming. This, combined with its

versatility, memory security, and ease, made it an appropriate choice for this project, where data science, cheminformatics, and web development were required.

Web frameworks manage communication between the frontend (web browser) and backend (server) by handling Hypertext Transfer Protocol (HTTP) requests and responses and integrating with other parts of the stack. Flask was chosen as the framework as it is more lightweight and flexible than larger frameworks such as Django. Flask can receive data from HTTP requests and send it to the specified route, meaning the intended function in the backend receives the data. After the route processes this request, it must return a response that the web browser understands; common examples include a status code indicative of outcome, e.g., success (200) or unauthorised (401), an HTML page, a status code indicating), or a JavaScript Object Notation (JSON) object with data used to update the current page. It also provides a structured way to make the application using blueprints. Each blueprint covers an area of functionality and one or more related endpoints. Using blueprints allows related endpoints and functions to be grouped and organised into modular components. This makes the code more maintainable. For example, there is an authorisation blueprint which covers registration and login.

Generally, the backend handles most of the logic of the application. This includes checking whether a user has permission to view a page, whether their login details are correct, and the transfer of reaction or compound data, for example. Flask also comes with best practices for structuring a larger project with standard directory names, meaning the code is understandable by someone with working knowledge of Flask and similar frameworks but unfamiliar with the project. There are also extensions

made for Flask to support database operations, such as Flask-SQLAlchemy and Flask-Migrate, for interacting with and updating (migrating) the database schema. When handling an HTTP request, the next step is often to interact with the database and then send a response based on this. This is where SQLAlchemy and PostgreSQL are used.

PostgreSQL is a relational database management system (RDBMS) that stores data in structured tables. PostgreSQL uses a query language called Structured Query Language (SQL). SQLAlchemy is an object-relational mapping (ORM) library in Python, allowing the database and its contents to be treated as Python objects. It does this by translating the Python code into SQL queries. This simplifies development by keeping the code for querying or updating database records consistent with the rest of the application code. The database schema is written in Python, and a model describes a table, corresponding to a type of record, such as reaction, users, and compounds. Each table has columns, and these and their data type column are specified in the model. Examples of data types include strings, integers, and dates. An ORM database can also map related objects from different tables. For example, a user can be mapped as the creator of a reaction, linking the entries from the different tables. These relationships between records in tables can be specified as one-to-one or one-to-many in the table model. When changes are made to the database schema by editing the model files, a migration file must be generated to provide instructions to the database on how to apply these changes. The Alembic Python package generates this script, which can then be applied to upgrade or downgrade the database to a newer or older schema version.

The application uses a collection of dependencies, of which SQLAlchemy and Flask are examples. Dependencies, like SQLAlchemy and Flask, are software components packaged to be easily installed in a project. Dependencies can be open-source, or they can be commercial or in-house. Open-source dependencies for Python are primarily made available through the Python Package Index (PyPi), which has over 570,000 packages available. This means developers don't have to needlessly "reinvent the wheel". RDKit is an example of a chemistry-specific dependency originally written in C++ with a Python wrapper, making it straightforward to use in Python projects. RDKit provides a way to treat molecules as Python objects, and its ubiquity provides somewhat of a common standard across chemistry software. RDKit supports the conversion of molecules between representations. For example, RXN, SMILES, and InChI molecular representations are all used in AI4Green.

The barriers to using packages are increased memory as the scripts and data associated with a package must be installed, increased project build complexity, and compatibility with other dependencies. Build complexity and compatibility can be mitigated by using a dependency manager to avoid installing dependencies one at a time. The simplest method is to list all dependencies in a file and install them using a command from pip install packages (pip). However, this requires all versions listed to be compatible with one another, including the Python version. Poetry is a Python package that can be used to manage a project's dependencies. Poetry takes a list of dependencies in a TOML file. It will seek to solve compatibility issues and find a list of suitable package versions with the required download links saved to a LOCK file, which can be updated and managed as the project's dependencies grow. Poetry uses these

files to create a virtual environment that isolates dependencies, making it reproducible across machines.

Another method used to organise the codebase is a services directory. This services directory is split into files supporting different functionality. These service functions can be imported throughout the application and used where needed. In general, it is good practice to break code up into functions for readability, particularly if these functions are reused. By removing complex logic from the blueprints' route functions, their narrative and purpose can be more easily understood. This can also help debugging as the source of errors can be tracked to a single function.

4.2.1.2 Frontend

The frontend of AI4Green is written in HTML, JavaScript, and Cascading Style Sheets (CSS). HTML is used to create hierarchical elements on the web page, which are what the user sees and interacts with. Buttons and text fields are examples of HTML elements. These elements are styled according to their CSS properties, affecting size, colour, and positioning. Bootstrap, a free and open-source front-end toolkit, is used and built upon to create visually appealing web pages. Jinja templates allow for dynamic content generation by combining static HTML with server-side data. This can enable the page to load with information specific to that user, such as their name or reaction data.

Interactive web pages are a key part of modern web design and are expected by users. To achieve this, JavaScript was used. JavaScript is a frontend scripting language that

can be executed in the browser, unlike many other programming languages. JavaScript functions can perform specific actions when a user clicks a button or interacts with the page in a certain way. These functions can be based purely on the frontend and could be used to edit HTML in some way, or they can perform a fetch or asynchronous JavaScript and XML (AJAX) request to exchange data with the backend. For instance, if a user clicks the 'add reagent' button, this will execute a JavaScript function to add HTML with inputs for a new reagent. When they enter the name or CAS number of the reagent, this will execute a JavaScript function that will send the CAS number to the backend and receive data about that compound, which is then inputted into the HTML fields. This means the webpage content can be updated with content from the backend and database without reloading the page, making the user experience more responsive. A user-friendly and intuitive frontend (shown in Figure 4.1. Additional images are shown in the appendix) makes the application more accessible and lowers the barrier to effective use of the software. This reduces the barrier to use and can increase user uptake, retention, and satisfaction, increasing the project's impact.

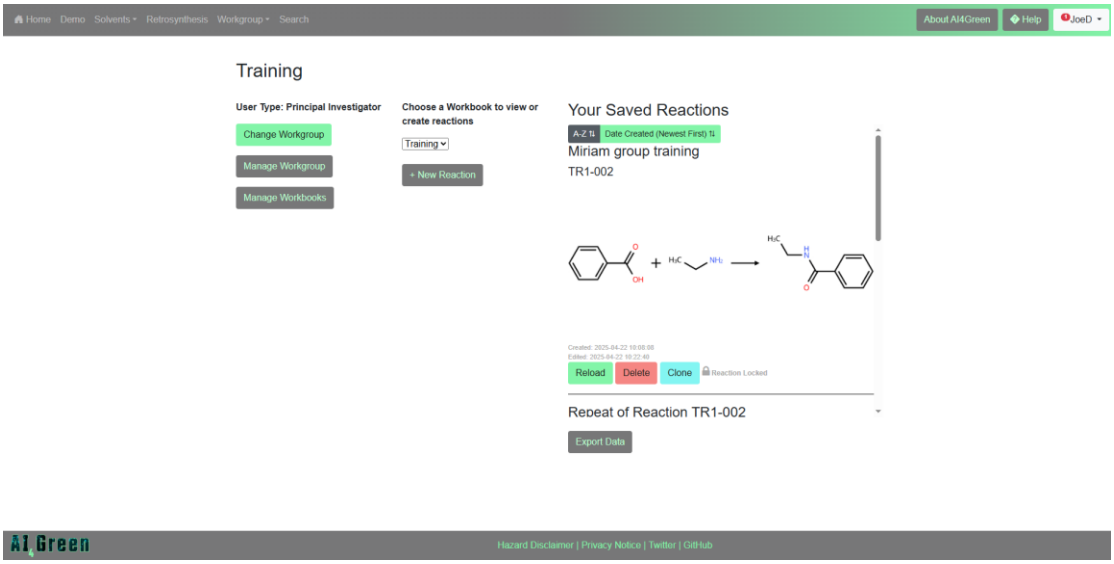


Figure 4.1: AI4green's workgroup interface. Workgroup management options are shown on the left. In the middle, the current workbook is shown in the dropdown list with the new reaction button below. On the right-hand side, the workbook's reactions are listed, with the option to export data shown below.

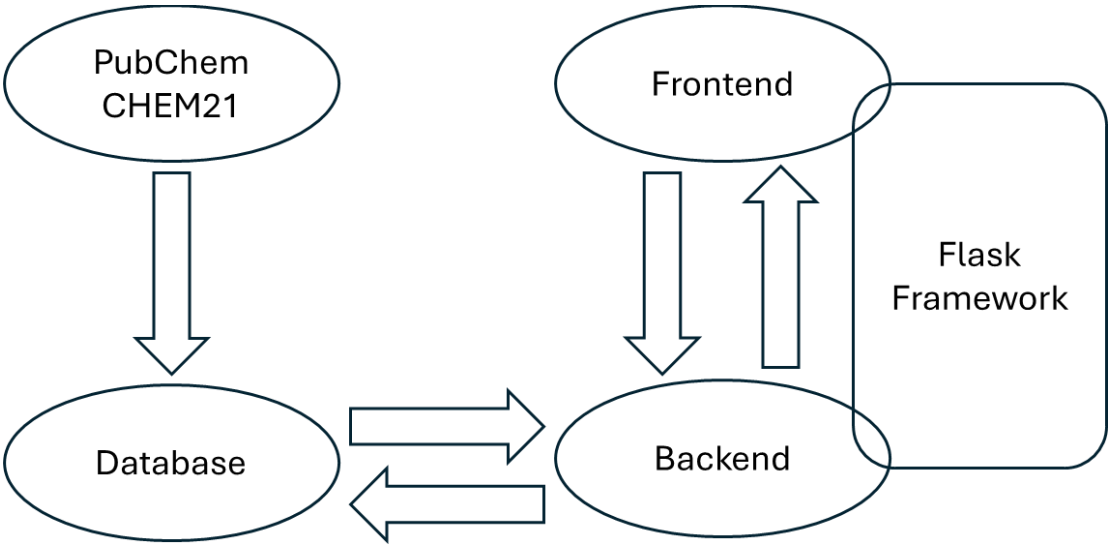


Figure 4.2: Simplified schematic of the AI4Green application architecture. External data sources, **PubChem** and **CHEM21** are used to populate the database**.** The Azure-hosted PostgreSQL **database** is connected to the backend through a connection string, and queries are made from the **backend** to the database using SQLAlchemy. The backend handles the application logic and is written in Python. The **Flask** web framework supports data exchange between the frontend and backend. The **frontend** is the interface the user views and interacts with and is made using HTML, CSS, and JavaScript.

4.2.1.3 Other Technologies

In addition to the application's frontend and backend, other supporting technologies are used within the stack. APIs are used to communicate with other software. Requests are sent to APIs, which then send a response in return. For example, if not found in the database, a compound identifier may be sent as part of a request to an external API. If this external service is successful, the response will contain information about that compound. APIs can also be used as an effective way to communicate with a machine-learning model hosted separately from the web application. An alternative would be to integrate all the functionality within the same application. However, this can lead to issues with maintainability, more requirements with greater potential for conflicts, and potentially computationally intensive functions that could reduce the performance of the rest of the application.

An integrated development environment (IDE) is used to develop the application. An IDE is software used to develop other software that is similar in concept to word processing software. It contains tools useful for software development, including error detection, debugger, and version control. The PyCharm IDE has been used in the development of this project.

Version control is a fundamental concept within software development. Git is the version control system used in this project. In the code base, there are multiple branches; each branch is a different version of the code base. There will be a main branch with special protection and rules designed for deployments as the working version of the application. Other branches are primarily used for the development of new features. This enables multiple developers to work on changes simultaneously

without interfering with one another's work. The IDE can display the uncommitted changes to the working branch. These changes to the project files can then be committed using Git. Once changes are committed, they are visible and part of the git history and can be restored or reversed. After changes to the code are committed, they can also be pushed. This pushes the committed changes to the online code repository. This means work is not saved on a single developer's machine, which may become lost or damaged. Pre-commit hooks can be used to ensure the code adheres to common styling practices, such as consistent indentation, are applied, and bad practices, such as unused variables, are not implemented. Commits which do not pass the pre-commit hooks are rejected with statements saying which lines fail and why, enabling the developer to change the failing lines of code. These pre-commit hooks can be included in the project repository and installed by the developer. Flake8 and Black are tools used to improve the quality of Python code in AI4Green. These tools make code comply with Pep8 guidelines to give Python code consistent conventions, prioritising readability as code is read more often than written.[136]

Once a new feature has been completed in a branch, a pull request to merge this into the main branch can be submitted. Policies in the code repository can be set, such as the number of reviewers required to approve changes and specific checks that the changes must pass, such as security bots that automatically detect changes that may introduce security risks. AI4Green uses an open-source GitHub repository. Once a pull request has been completed, the updated main branch can be deployed. This process can be partially automated using a continuous integration and deployment pipeline. Workflows using GitHub actions can be set up to take place when certain events occur,

such as a completed pull request into the main branch. In AI4Green, a workflow is used to deploy an updated main branch to a testing server. Once the test server is confirmed as working as expected, the updated main branch is then deployed to the production server. This step is valuable as there may be differences between a developer's local environment and the cloud server. However, there should be minimal differences between the testing and production servers. Workflows are advantageous as they automate the deployment process. This is faster and reduces the risk of human error while providing a record of all deployment actions.

The application requires a server to run on. In development, this can be the developer's machine. However, for testing and production servers that are designed to run on the internet and be accessible to others, an alternative is required. For this, external servers run by Microsoft Azure are used. Azure offers different server options for different use cases. The main application of AI4Green runs as an app service. This means that in the deployment process, the application is built, such as all dependencies being installed, and then run on this server. The PostgreSQL database also runs on a different specialised Azure-hosted server and has a connection string that the main application uses to connect to this database. Using a dedicated PostgreSQL server provides benefits, such as automated backups. The connection string and other secrets are saved in a Key Vault resource; they are not saved as regular text anywhere apart from the Key Vault. This is important since the database contains personal and potentially sensitive IP data, which must be secured. The application also accepts file uploads, which must be stored and saved somewhere. This is done by saving them to an Azure Binary Large Object (Blob) storage account.

When running the application in a development environment, the main application can be run by a Python script, either from the terminal or PyCharm. However, the application also relies on API microservices, the database, and a file storage system for user-uploaded file attachments. Since the files of these additional systems do not need to be edited whilst working on the main application, they can be run as Docker containers. Docker is a software that can run containers, which are packages of software that have been containerised. Containerised software is self-contained and includes everything needed to run, such as code and dependencies. While containers are not full virtual machines, as they use the host system's operating system kernel, which makes them more efficient, they behave in a similar way, allowing the software to run on any operating system, such as Mac, Windows, or Linux. In AI4Green, a PostgreSQL container is run to create a PostgreSQL server and Azurite, an Azure Blob storage account emulator. This means that the software can interface with these external applications in the same way in development and production environments.

4.2.1.4 Tests

PyTest is used for unit tests. These tests aim to isolate a unit of the code with a single functionality. These are quick to run and can rapidly identify the source of a potential error in the code. Typically, unit tests will use an input and check that the function under test returns the expected output. This provides a quicker and more consistent method than manually testing the code. Identifying edge cases involves considering non-obvious actions users may take. For example, it should be considered that users may include non-standard alphanumeric symbols in their inputs or trailing spaces. These things should be handled, and tests can be used to confirm expected behaviour

and appropriate function response. Because of the speed and structured nature of these tests, it is valuable to identify edge cases and include them in tests.

Integration or end-to-end tests simulate the user's interactions with the web application by opening a browser and clicking buttons, entering text, and navigating the application. Thus, integration tests can involve the entire application stack, including frontend, backend, and external services. This can identify what functionality is broken, and the error stack obtained can provide additional details about the involved functions. Integration tests take longer to run, requiring the web page to load after each step. Still, they can more accurately represent how a user would use the web application and test how functions interact and exchange data. Selenium with ChromeDriver was used to support integration tests with AI4Green, but these require updating after a significant restructuring of the application.

4.3 Functionality

AI4Green includes a wide range of functionalities that support chemists and take advantage of the benefits an ELN can provide.

4.3.1 Database

The database in AI4Green has two purposes: to provide relevant data on compounds, hazards, and sustainability and to save user data such as reactions, novel compounds, account data, and workgroup and book permissions.

4.3.1.1 Database Seeding

Compound data is seeded using the functionality in the compound_database directory within AI4Green. First, the most recent version of the PubChem laboratory

chemical safety sheets is downloaded as an Extensible Markup Language (XML) file. The XML file is parsed using etree from the lxml Python package, which converts the XML into a Python object, providing a Pythonic way to interact with the XML file. XML files are structured hierarchically and use tags to describe the data they contain, similar to HTML. Etree is used to obtain data inside the desired tags. The strings in these tags can then be manipulated to extract the desired value, which is validated. This validation takes several forms, depending on the property being extracted. Regular Expressions are used frequently in this string validation. This involves validating that the string matches a specified pattern. Examples include checking that CAS numbers fit the expected pattern, checking the units of temperature, and then converting them all to Celsius.

The properties selected to be seeded were chosen on two main criteria: relevance and feasibility. They are shown in Table 4.1. PubChem offers other download formats, but laboratory chemical safety sheets (LCSS) were selected as they provided the best balance between valuable data and all the relevant data. Other downloads included much larger amounts of unnecessary detail regarding 3D structural data. Hazard code data was the greatest priority, as retrieving these automatically can help chemists record safety data quicker and more reliably. Safety is also an integral part of the sustainability of a reaction or process. In the LCSS, hazard codes are provided by multiple chemical suppliers. Hazard codes are taken preferentially from the European Chemicals Agency (ECHA) and then Signal as a backup when data from the ECHA is not provided. Additional data related to sustainability were also captured where possible, including the boiling, melting, and flash points, and the autoignition temperature.

Chemical identifiers are essential when querying the database to retrieve the compound data. Name, CAS, PubChem compound identifier (CID), and InChI were extracted, and the InChI key and SMILES were also obtained from the InChI. Compounds without a unique CAS were not extracted, including duplicate CAS and compounds without a CAS number. While CAS numbers are proprietary, the ones used here are for common chemicals made publicly available by PubChem. Over 150,000 compounds were extracted into the database.

Table 4.1: Properties seeded into the compound database.

Property Name (Units)	Property Type
CAS	Identifier
PubChem CID	Identifier
Common Name	Identifier
InChI	Identifier
InChI Key	Identifier
SMILES	Identifier
Molecular Formula	Compound Attribute
Molecular Weight (g/mol)	Compound Attribute
Hazard Codes	Compound Attribute
Concentration (mol/dm^3)	Compound/Mixture Attribute
Density (g/cm^3)	Compound Attribute
Boiling Point (°C)	Compound Attribute
Melting Point (°C)	Compound Attribute
Flash Point (°C)	Compound Attribute
Autoignition Temperature (°C)	Compound Attribute

CSV files were made for solvent sustainability, element sustainability, and hazard code severity based on the CHEM21 sustainability framework. These CSV files were used to populate tables in the database and integrate this data into a single source. At this stage, the standard use and admin role types were seeded. Optionally, dummy user, workgroup, and workbook data are seeded to speed up development.

4.3.1.2 Storing User Data

The database manages user data, including login credentials stored in the User table. When a user registers, a corresponding entry in the Person table is created, establishing a one-to-one relationship between records in these tables. This structure enables role-based data management, with the User table handling personal and account data and the Person table recording their associated workgroups, workbooks and reactions.

This separation allows the web application to meet data privacy requirements, as users can request that their personal data be removed from the system. This allows the association of reactions with a specific person, even if they are anonymised as "Person 6". This can be valuable as reaction outcomes can vary between chemists.

The database also includes Workgroup, Workbook, Reaction, and Novel Compound tables. Workgroups and Workbooks are linked to the Person table, which manages access to only allow authorised users to view or edit specific data. The Reaction and Novel Compound tables store data used to reload a reaction or previously entered compounds. Access to these records is restricted to workbook members. These records are likely to contain proprietary data and unpublished data. This controlled access structure is essential for maintaining data security whilst enabling a collaborative research environment. The user is presented with this hierarchy of reactions belonging to workbooks which belong to workgroups. This is illustrated by the user interface for the "quick access" section on the home page, as shown in Figure 4.3.

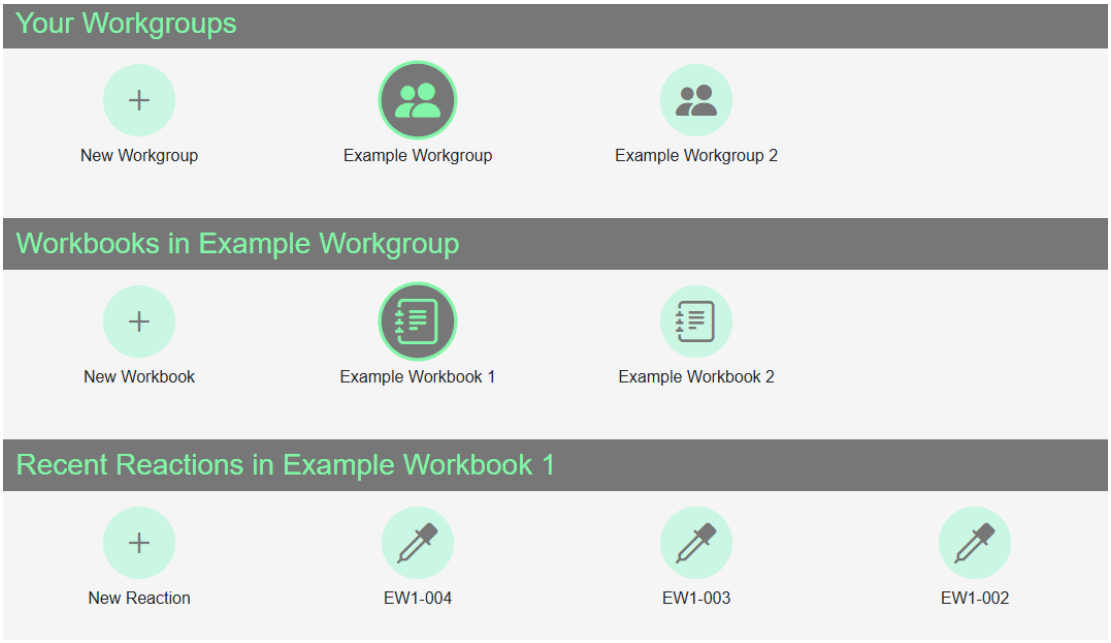


Figure 4.3: This is the user interface of the quick access section of the home page. The user selects a workgroup, which shows a list of workbooks that belong to it. The user then selects a workbook and is shown a list of reactions belonging to that workbook. This view provides a simple method for displaying the hierarchical nature of the data.

4.3.2 Workgroups & Workbooks

When registering on AI4Green, a user is encouraged to create or join a workgroup. A workgroup mimics a real-life research group. There are three roles within a workgroup that a user can be: principal investigator, senior researcher, and standard member, each with different levels of permissions. Multiple workbooks can be made within a workgroup. Workbooks are intended to be specific to a project or chemist, depending on user preference. A principal investigator has administrative control of the workgroup with features including adding new users by email or sharing a quick response (QR) code. Principal investigators and senior researchers can create workbooks and add or remove other users from a workbook, whereas standard members cannot. A use case of this is to assign students as standard members to stop them from being able to join other students' workbooks and copy work when they are

being assessed. One interesting option that can be incorporated into future versions of the software is deeper integration with university systems, where data on whether a user is a student or staff can be used to reduce the administrative load.

Reactions and Novel Compounds are stored at the workbook level. Novel compounds are added when a user wishes to input a compound to their reaction that is not already in the database. This can happen when they draw a chemical structure and the corresponding InChI is not in the Compound table of the database. Alternatively, if the user enters a CAS or a name without a match in the database, they can add a new compound as a Novel Compound. Novel Compounds are stored in a separate table, as these are not seeded from PubChem; therefore, less data are available. The user is prompted to enter details of the compound, such as density, name, CAS number, concentration, and if not drawn, SMILES and molecular weight, also. The novel compound is saved in the same workbook where the reaction was made. This means it can be retrieved by drawing it or entering one of the provided identifiers, either the provided CAS number or name and the chemist does not need to re-enter the data again. To aid retrieval, recently used compounds within a workbook are provided as a dropdown list when a user begins searching.

Reactions can be reloaded from their workbook page. A scrollable list of all the reactions in a workbook is provided, with an image of the reaction scheme. They can be sorted by date or alphabetically and can only be reloaded by a workbook member. If a user tries to load a reaction from a workbook they do not belong to, they will be redirected to the home page.

4.3.3 Reactions

Reactions are created from the workbook page. Upon creation, a name is required, and a unique identifier is generated based on a consistent three-letter code for the workbook followed by a zero-filled auto-incrementing number, WBK-001, for example. The reaction constructor is designed for a single-step chemical reaction and is best suited to organic chemistry. The reaction constructor can be divided into three parts: the chemical sketcher, the reaction table, and the summary table. The Ketcher chemical editor is integrated into the AI4Green code base to enable chemists to draw their reaction scheme with molecules drawn to the left of the arrow recognised as reactants and molecules to the right as products. When the chemist clicks the submit button to enter their reaction scheme, the molecules are sent to the backend as reaction SMILES with a '.' delimiter separating molecules' SMILES strings. These SMILES strings are then converted to an InChI, which is then used to query the database to find the compound. If a match is found, the name, molecular weight, density, and hazard codes are returned and used to generate the reaction table if a match is found. If no match is found, the user is prompted to provide data on the compound and add it to the database as a novel compound. This populates the reaction table with the reactants and products used in a reaction.

The user can add reagents to the reaction table by CAS number or name. A regular expression will determine whether the user has provided a CAS number or a name and then query the database for the provided identifier. If a match is found, similar to reactants and products, the name, molecular weight, density, and hazard codes are returned, and if no match is found, the user is prompted to add a novel compound to

the database. Solvents are added to the reaction table similarly, except there is also a dropdown list of common solvents colour-coded by their CHEM21 sustainability assessment. The solvent CHEM21 sustainability assessment has four possible levels, providing a simple but meaningful way to distinguish solvent sustainability. InChI and CAS numbers are the preferred identifiers, as compounds only have a single CAS number or InChI string but can have multiple names or SMILES.

Cells where the user needs to input values to the reaction table are highlighted with a red border. These include the mass of the limiting reactant, which the user specifies, and the equivalents of all other reactants and reagents. The mass, molar, and volume units can all be changed to reflect the scale of the reaction. The remaining mass, molar, and volume values for reactants, reagents, and products are automatically calculated. The volume of solvent used must be specified, and is used in conjunction with the amount of the limiting reactant to calculate the reaction concentration. The physical forms of the compounds must be selected from a dropdown; this includes an 'Unknown' option for compounds not previously synthesised. The physical form provides a compound's exposure potential, contributing to the safety assessment in the summary table. The user can generate a summary table once all the desired compounds have been added and highlighted cells filled in.

The summary table is similar to the reaction table, containing all the masses and compounds required for a reaction. It also has a safety assessment, sustainability assessment, file attachments, and an input field for the yield. The safety assessment includes the risk matrix, which combines the severity of a compound's hazard codes and its exposure potential to generate a risk rating for each compound and an overall

risk rating for the reaction. The hazard code severity potential has three possible ratings; these are assigned using labels created in the CHEM21 project. The exposure potential also has three possible ratings, with dense solids having the lowest exposure potential and gases and highly volatile liquids having the greatest. In addition to the risk matrix, there is also space for chemists to indicate any standard protocols they need to follow, such as the use of a sealed tube, how they intend to dispose of waste materials, spillage procedure, and space to write any other risks and actions to mitigate risk, such as the use of personal protective equipment (PPE). Finally, there is space for a final assessment by the chemist to indicate the hazard, likelihood of risk, and scale of consequences of the reaction to generate a numerical risk score, which is assigned a category between D and A, where D is the lowest risk category and A the highest. Below is a space for them and their supervisor to sign the reaction. The safety assessment uses the industry-standard practice of a risk matrix but includes flexibility to enable chemists to meet their institutional requirements.

The sustainability assessment (Figure 4.4) is automated as much as possible and is based on the CHEM21 holistic sustainability framework. Element sustainability refers to the remaining years until known reserves are depleted based on the current extraction rate. It is computed by looking at all elements used in the compounds in the reaction and their remaining supply. Atom economy is also automatically calculated from the reaction table based on the compounds' equivalents and molecular weights. Percentage yield and mass efficiency are calculated from the mass yield, which a chemist would enter in a reaction as standard practice. Conversion and selectivity of the reaction are calculated once the chemist enters the mass of

unreacted starting material, which can most easily be estimated by methods used to measure the reaction progress, such as HPLC or NMR. Temperature, whether the reaction is completed in a batch or flow system, how the product is isolated, what stoichiometry of reagents is used, and whether a catalyst, if used, is recovered must all be inputted manually by the user but are important experimental details that should be captured and are then colour-coded by their sustainability automatically.

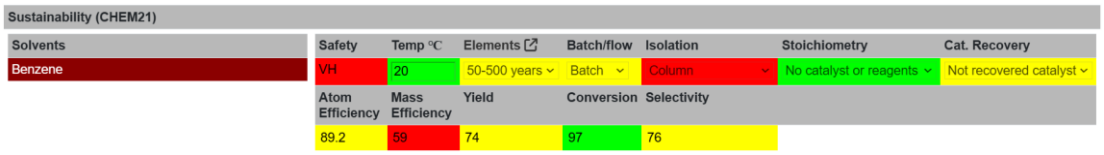


Figure 4.4: Example sustainability assessment from the AI4Green interface. Metrics are colour-coded by their sustainability.

Once a mass for yield and unreacted starting material has been inputted, the reaction can be locked, preventing further editing and providing a timestamp. Once a reaction has been locked, the author can add comments to provide any additional information that may be needed. Chemists can upload file attachments to reactions to include spectral data demonstrating the isolation of the desired product. Files can be uploaded before and after a reaction has been locked. Users can search for reactions by drawing a compound, and the system will return all reactions involving that compound. The search results can also be filtered by workbook and workgroup.

4.3.4 Data Export

Reaction data can be exported from AI4Green in machine-readable formats such as JSON, RXN, RDF, and ELN, as well as human and machine-readable formats such as comma-separated values (CSV) and simple, user-friendly reaction format (SURF). Users can export for all or selected reactions in a workbook if a recorded yield is

available. A data export request system is used since the data may be sensitive, such as pre-publication or patent. An email with details of the request is sent to all principal investigators in the workgroup, and all must approve the request to proceed. Once all approvals have been granted, the requestor receives an email with a download link. Using email here is a type of two-factor authentication, as it requires the approvers and requestors to all have access to their AI4Green and email accounts, making the process more secure.

If multiple files are required for the export, then these are compressed and zipped into a single file. During the creation of the export, temporary files are often required, which are deleted once the export has been created. These are saved and managed using the Azure blob service. The export file is then stored for seven days before it is deleted, and this is controlled by using an export container and creating a rule on the Azure blob service to delete files in this container seven days after creation.

4.3.4.1 Human and Machine-readable Formats

CSV and SURF are tabular exports with a series of columns of data properties and one row per reaction. The columns in the CSV export include all data that AI4green stores for a reaction, whereas SURF is an open-source format with a few differences.[137] SURF aims to be an interoperable machine and human-readable format. The authors of SURF provide code that enables it to be converted to and from the unified data model (UDM) and ORD formats. The SURF format describes the reactants, reagents, catalysts, solvents, and products, their equivalents within a reaction, the scale of the reaction, and solvent concentration. A structural identifier, InChI or SMILES, is used in addition to the CAS number for each compound. Catalysts are classified as reagents with lower

than 0.25 equivalents or a 25% catalyst loading. Conditions including the temperature, time, atmosphere and whether it was stirred or shaken are all included. Additionally, there are fields for yield, yield type, product characterisation, experimental procedure, and provenance. This provides detailed information suitable for reproducing the reaction in most cases. The CSV export aims to export all the reaction data stored in AI4Green, including all columns found in the SURF export, sustainability and safety data and details of who performed the reaction.

4.3.4.2 Machine-Readable Formats

The JSON export includes a JSON file per reaction and is downloaded as a zip file. The JSON export includes the same information as the CSV export. RXN and RDF formats are both examples of Chemical Table file formats. These are exported as one RXN or RDF file per export and are downloaded as a zip file. These provide a way to describe the structure of the reactants, products, and agents in a reaction. Agents include catalysts, solvents, and reagents. Making an informed guess at the agent's role is possible, but it can be challenging, especially as these roles can sometimes overlap. The RDF format includes additional data at the end of the file. Meanwhile, the RXN file contains only the structural information of the molecules involved.

The ELN file format was developed to enable interoperability between different ELN systems.[138] It is based on the research object crate (RO-Crate) specification.[139] ELN files are ZIP archives that must contain a JSON for linked data (JSON-LD) file called ro-crate-metadata.json and then a directory for every experiment recorded. The ro-crate-metadata.json file describes the contents of the ELN export using a standardised vocabulary developed from the RO-Crate specification and Schema.org.[140] The

experiment directories contain a RXN and JSON export file as well as any file attachments from the user. The ELN file export is not specific to chemistry. Therefore, the JSON-LD includes properties common to all experiments and, for each experiment, describes the name, author, comments and associated files for a reaction. There are also flexible description and text sections. The experimental procedure is entered as the description, and the text encoding format is specified as HTML. It includes the rendered summary table template and reaction SMILES in place of the scheme, as images are not supported, and other ELNs may not provide a chemical editor. Authors, comments, and files are specified as belonging to a reaction by their ID and are defined fully later in the JSON-LD.

4.4 Discussion

4.4.1 Overview

AI4Green is a long-term project and is nearly halfway through a ten-year funding grant, and the ELN is a crucial part of it. There are a few key methods by which the ELN can be evaluated. The project aims to enhance sustainability, develop tools that support this goal, improve chemists' workflows, and make chemistry data FAIRer and more useful. These aims require a quality product with intelligent functionality, solid development foundations that can be built upon to continue the sustainable delivery of new features, and a growing user base. These requirements are interlinked as a solid foundation that enables the continuous integration/continuous deployment (CI/CD) pipeline to deploy new features quickly, which is especially important for bug fixes. Advertising these features and providing a quality experience will help attract

and retain new users. The extent to which these requirements have been met and what future work is required will be evaluated and discussed.

4.4.2 Digital Research Environment Assessment

An analysis of the requirements of a digital research environment (DRE) was carried out by Kanza and co-workers and will form the basis for assessing to what extent AI4Green has the required features for a DRE.[132] Their publication includes a series of feature lists with multiple examples of what should be implemented for each feature. The functionality in AI4Green will be compared to the list of requirements for each feature, and it will be determined if this has been wholly implemented or if more work is required. It is not expected that AI4Green will implement all of this functionality itself, but instead, it will implement enough of the required functionality that integration with other specialised software will form a complete DRE. Avoiding "reinventing the wheel" is a common aim within software projects where developer resources and time are limited, and it is an important ethos for building specialised software.

4.4.2.1 Generic Features

Generic features are ones which are expected from modern software (Table 4.2). They make the software more practical to use and improve the user experience. API access requires an implementation to allow external services to interface with AI4Green programmatically. This can be done to aid data import and other tasks that may be more suited to doing programmatically. Additionally, it can enable external services to use select features within AI4Green. A comparison can be made with other tools available through API access, such as the Chemical Identifier Resolver (CIR) provided

by the US National Cancer Institute. API access could be integrated into AI4Green using the existing Flask Blueprint structure and adding endpoints designed for API access. Existing features like sustainability assessments would make logical API endpoints for other applications.

Autosaving, sustainability assessments, molar calculations, data backup and updates to the application are examples of automated features in AI4Green. These are supported by different aspects of the technology stack, with autosave, sustainability, and molar calculations handled by the frontend and backend as required, and the data backup and updates to the application are supported by Azure resources and the CI/CD pipeline.

Table 4.2: Generic features required for a digital research environment.[132]

Feature	Status in AI4Green
API Access	Requires implementation
Automation	Multiple examples implemented
Graphical User Interface	Implemented
Localisation	Requires implementation
Remote Access	N/A - Browser-based
Scalable	Implemented
Synchronisation	N/A - Browser-based

Many of the listed features are implemented as default or are not applicable because AI4Green is a browser-based application. The graphical user interface (GUI) uses HTML to create elements on the page, JavaScript to make this dynamic, and CSS to style and make the application visually appealing and modern. Remote access and synchronisation are not concerns for a web-based application but may be needed for a desktop application. AI4Green is accessible via the internet, and the database does not require synchronisation with any other storage location. The application is also

scalable by default because the Azure resource on which they are hosted can be automatically scaled. There is no localisation support for different languages, but it is not a priority at this relatively early stage. Current efforts have most strongly focussed on growing the user base in the UK higher education sector. The generic features are mostly implemented, and those that have not yet been implemented are not to the detriment of the application at this stage.

4.4.2.2 Notebooking features

AI4Green has Notebooking features integrated throughout (Table 4.3). However, most of them have room for further development. The integrated chemical sketcher demonstrates that content support is present but mostly aimed at organic chemists. This could be expanded to include more general features, such as supporting custom table creation and support for barcodes and lists. Multiple free text boxes enable the user to flexibly add links and references. The file upload system also supports various attachments of common and scientific file types. The word processing support offered by the free-text boxes could be improved by building in features found in Microsoft Word that support scientific writing, such as super and subscript text. The proposed feature of dynamically linking compounds in the reaction table to text in the experimental write-up relates to this. This would involve automatically entering the compound name, mass, amount, and volume as text in the write-up, updating as the reaction table updates. Paper integration, including features such as digitising old data on paper and scanning documents, is not integrated within AI4Green. It is foreseeable that this specialist functionality could be used in multiple software applications, and an existing, purpose-built solution would be preferred.

Table 4.3: Notebooking features required for a digital research environment.[132]

Feature	Status in AI4Green
Content Support	Partially Implemented
Interaction/Access	Partially Implemented
File Links	Implemented
Organisation/Reconfiguration	Mostly Implemented
Paper Integration	Not Implemented
Referencing/Literature	Partially Implemented
Word Processing	Partially Implemented

4.4.2.3 Data Features

Most data features are partially implemented, reflecting the planned future work in the area (Table 4.4). Additionally, there is significant overlap between data-related features, with some solutions meeting multiple requirements. This demonstrates the importance of judicious design choices for an application. In AI4Green, data are either from AI4Green or provided by users. The former refers to sustainability and compound data, with the latter referring to reaction data. The user login, workgroup, and book hierarchy systems control data access. However, users cannot access the data via API or see who has accessed it. These are both requirements that could be incorporated to improve the application. Data cannot be converted between different formats or exchanged across platforms in AI4Green. Conversion could be implemented with predominantly existing functionality or external purpose-built tools. AI4Green is not a data exchange platform, but working with one, such as the ORD, is part of the project goal of identifying a suitable home to publish data in a location with FAIR data principles.

Table 4.4: Data features required for a digital research environment.[132]

Feature	Status in AI4Green
Access	Partially Implemented
Conversion	Not Implemented
Exchange	Not Implemented
Integration	Partially Implemented
Management	Partially Implemented
Quality	Partially Implemented
Retention	Partially Implemented
Security	Implemented
Standards	Partially Implemented
Storage	Partially Implemented
Support	Partially Implemented
FAIR	Partially Implemented
Identifiers	Partially Implemented
Provenance	Partially Implemented

Data integration is the ability to combine different data. This includes from different sources, teams, and types. Workgroups and workbooks aim to integrate data across teams. Currently, AI4Green is tailored exclusively to organic chemistry, meaning that data outside the scope of organic synthesis is only supported by file attachments and free text, which limits its integration. Planned work to revamp the reaction constructor and build a more flexible modular system will enable this improvement and others. Data management is partly implemented, and the next steps of providing greater user-management options and a pipeline for processing and export to an external repository will meet this requirement. Current data management functionality centres around the database, which provides a robust solution. The database also provides a strong level of security and has the option for appropriate retention policies. Currently, all data are retained indefinitely.

Data provenance, quality and standards enable the data to be trusted, valuable and universal. The provenance of data used in AI4Green, such as hazards and sustainability

scores, is explained. The provenance of reaction data from the user is stored by default as the author, workbook, and workgroup of a reaction are all captured. Any literature provenance for the reaction can be saved as free text, but a specific field to capture this would encourage users to add a reference and aid reproducibility. Reaction data are not quality-checked and relies on the user to input accurate data. This is a challenging problem as the problem is more complex than a spell-check, with a clear solution, for example. The problem area of synthesis is more complicated, with results highly variable depending on numerous factors, and much work is novel; therefore, there is no precedent to compare. However, introducing basic quality assurances, such as challenging percentage yields over 100, would be beneficial. To control the quality of reaction data and have a level of trust, workgroups must be approved by the AI4Green administration team.

Data storage requirements are met by having automatic saves, backups, and archival of deleted reactions and orphaned workgroups, defined as those without a principal investigator. The data support functions in AI4Green enable secure downloading of reaction data through an export system. Additionally, file attachments allow users to integrate data from external services. Combined with the reaction constructor in AI4Green, this functionality supports data management throughout the reaction process. However, the current system only supports importing data via file attachments, with no other import methods available currently.

Unique identifiers are generated in AI4Green in a few ways. The database can enforce a particular property as unique by forbidding two rows from sharing a value within a table. Users can provide values checked for uniqueness and rejected if this fails.

Alternatively, a unique identifier called a universally unique identifier (UUID) can be generated in Python using the built-in UUID library. Version four UUIDs are randomly generated, compared to non-random UUIDs generated using a variable input string and application-specification namespace. This is because 122 bits of the generated string are random, giving 2^{122} possible combinations. This low likelihood of collision and relatively low risk of the application (compared to software used in critical systems, such as aviation safety) means these are assumed to be unique. However, to meet the requirements of a DRE, unique identifiers could be extended to more data in AI4Green, such as the specific batch of a compound.

Many areas requiring more work to fully meet the requirements of a digital research environment will become vital as the application matures and demand for features such as API access grows. Currently, this demand is not there as the end-users are chemists. Still, once a significant data repository is built up by prolonged use within a group, it can be anticipated that demand for support and data-focused users will grow.

4.4.2.4 Publishing & Sharing Features

Publishing and sharing features help users more easily publish and share their work this includes stronger methods for publishing data in a superior way to a standard supplementary information (SI) (Table 4.5). User documentation is provided, but more support for creating documentation that can be shared and reused for techniques and standard procedures still requires planning and implementation. The export feature in the application also has six formats for exporting the data. A pipeline to publish data still requires implementation. A pipeline to publish data in the ORD could meet most of these requirements. It would have to generate Digital Object Identifiers (DOIs), the

data would be licensed under the ORD's terms in that scenario, it would be open-access, and the dataset would be attributed to the correct researchers. This makes sense within the ethos of a DRE, where AI4Green, as the ELN, interacts with another specialist software, such as the ORD or Chemotion repository.[80,81]

Table 4.5: Publishing and sharing features required for a digital research environment.[132]

Feature	Status in AI4Green
Documentation and Instructions	Partially Implemented
DOIs	Not Implemented
Export	Implemented
Licensing	Not Implemented
Open Access	Partially Implemented
Publishing	Partially Implemented
Sharing	Partially Implemented
Social Media	Not Implemented
Researcher Attribution	Partially Implemented
Repositories	Not Implemented

The workgroup functionality enables data sharing, and there are plans to add view-only workgroup roles and the sharing of specific reactions to improve this functionality. Further export options are planned for development, notably a human-readable export of the experimental write-up in the style required for a thesis or publication. This feature is mentioned in the analysis of the requirements of a DRE work published by Kanza and co-workers and has also been requested in user feedback.[132]

4.4.2.5 Collaboration & Management Features

Collaboration and management features are important as they provide functionality that can increase the effectiveness of the ELN (Table 4.6). Auditing is a feature currently lacking and requires implementation. This can be done as part of the planned

changes to saving, where versions of a reaction are saved as checkpoints, similar to version control in software development. This would provide a series of timestamps and the corresponding activity, which could form an audit trail. Audit trails are essential for companies and organisations to ensure policy compliance and can contribute to patent claims. Additionally, saving reactions using version control provides a "soft lock," which can be beneficial as users can reload if they make any erroneous changes and see when changes were made. Also, it is less intimidating than a "hard lock," as is currently implemented.

Table 4.6: Collaboration and management features required for a digital research environment.[132]

Feature	Status in AI4Green
Auditing	Not Implemented
Comments	Partially Implemented
Notifications	Implemented
Subscribe	Not Implemented
Team management	Implemented

Comments are only enabled on locked reactions to add any omitted details without changing the existing work, but they could be extended to all reactions to increase communication. Notifications are implemented and alert users to requests and invitations, such as to join a workgroup or for a data export. They are combined with an email alert system, as not all users, especially principal investigators who receive the most requests, will necessarily log in to the application frequently.

A basic team management system is implemented using the workgroup and workbook system, where principal investigators and senior researchers can manage users. However, this does not meet all team management requirements, such as work allocation or team statistics. Full team management requirements could be integrated

within future versions of AI4Green. Specialist software could be used alongside AI4Green. However, that software must be free or open-source with a suitable API and extensible to organic chemistry. Subscriptions are not part of AI4Green and would have to be implemented in a way that ensures sensitive data is still protected.

4.4.2.6 Domain-based Features

AI4Green has strong domain support for organic chemistry (Table 4.7). Domain-specific data are extracted from PubChem and CHEM21 and stored in the database. This enables chemical support through identifiers, default lists of solvents, and extensive health and safety data. Experiment planning is implemented in AI4Green, including features such as experiment locking, copying, and protocols. However, additional flexibility is required to support HTE and optimisation work more intuitively. Ketcher is integrated into AI4Green as the chemical editor, but links to other domain-based software, particularly a synthetic chemistry database such as Reaxys or SciFinder, would be valuable. Other planned work includes integration with a chemical inventory application. One such application is ChemInventory, and integration with this has been explored, but further work is required. Each institution would have its own ChemInventory API key, and integration of this would enable workgroup members to see the chemicals in stock within their parent institution. Similarly, the equipment interface would require some level of institution-specific implementation and requires implementation in a future version of AI4Green. It could significantly speed up chemists' workflows if data were automatically uploaded from the instrument to the correct experiment.

Table 4.7: Domain-based features required for a digital research environment.[132]

Feature	Status in AI4Green
Chemical Support	Implemented
Default Lists	Implemented
Equipment Interface	Not Implemented
Experiment Planning	Partially Implemented
Health and Safety	Implemented
Link to Inventory Software	Not Implemented
Link to domain-based databases and software	Partially Implemented

4.4.2.7 Coding Support

Coding and versioning within AI4Green are supported through the open-source code repository which anyone can contribute towards (Table 4.8). The current version of AI4Green is specified in the TOML file and is findable at the /version endpoint. The codebase is hosted on GitHub and includes instructions on setting up the application in the README. Tests are automatically run as part of the pull request pipeline, the code is documented, and Git is used for version control. These contribute to a professional software ecosystem, providing a solid foundation for building a quality application. Without these systems, more time is wasted on manual code management, bug fixing, and an over-reliance on manual testing.

Table 4.8: Coding support features required for a digital research environment.[132]

Feature	Status in AI4Green
Coding	Implemented
Versioning	Implemented

4.4.2.8 Metadata, semantics, & AI

Metadata, semantics, and AI can provide value through interoperability, better data recording, and intelligent suggestions from data analysis (Table 4.9). AI4Green has integrated machine learning models to help chemists and encourage sustainability.

These models rely on external data; solvent data come from CHEM21 and other sources, and reaction data are extracted from the USPTO and Reaxys.[38,43,90] This could be built upon by enabling users to use their data for modelling purposes; these data would be more relevant and specific to their problem area. Transfer learning could be used to update models and tailor them to be more project-specific. Metadata is recorded for experiments, but there is potential for more details to be captured. More advanced methods, such as knowledge graphs, are not used. Plans to enable users to add keyword tags to experiments could help with this and semantics. Full metadata would rely on other features being implemented, such as interfacing with instruments to capture metadata of analyses. The ELN file export relies strongly on an ontology, which could be adapted elsewhere in the application to enhance the interoperability of the recorded data.

Table 4.9: Metadata, semantics, and AI support features required for a digital research environment.[132]

Feature	Status in AI4Green
AI tools/integration	Partially Implemented
Metadata	Partially Implemented
Semantics	Partially Implemented

4.4.2.9 Searching

ELNs can aid in more effective searching. Searching is essential in rediscovering existing data and can prevent unnecessary experimentation duplication (Table 4.10). Domain and notebook searching are used in AI4Green to help users find old experiments they can access. Users can search for reactions containing specific structures, which can be filtered by workbook and workgroup. Planned features to enable more intelligent domain-based searching require implementation and would

include functionality such as substructure searching. Additionally, searching for keywords and the contents of experiment protocols would be beneficial. Literature searching is not part of AI4Green, but it may be helpful to integrate through other software such as Sci-Finder or Reaxys, or alternatively, the ORD for just the reaction precedent. Datasets cannot be found through AI4Green, as only authorised users can view the data. This would be an important consideration when selecting downstream data repositories to integrate with.

Table 4.10: Search features required for a digital research environment.[132]

Searchable by	Status in AI4Green
Domain	Partially Implemented
Characteristics search	Not Implemented
Keyword or content types	Not Implemented
Literature	Not Implemented
Notebook	Implemented
Indexing	Not Implemented

4.4.2.10 Customisation & Extension

Customisation can help users work by enabling them to tailor the software to their individual needs, which may vary compared to other users (Table 4.11). AI4Green enables the personalisation of the colour schemes used to indicate sustainability. These can be saved by a user and changed throughout the website for that user. Templates are not yet implemented in AI4Green but are part of planned work on overhauling the reaction constructor to increase flexibility, providing an environment more suitable for templates.

Table 4.11: Customisation features required for a digital research environment.[132]

Feature	Status in AI4Green
Personalisable	Partially Implemented
Templates	Not Implemented

4.4.2.11 Training & User Support

Training and user support are important for giving new users the tools to make the most of the software (Table 4.12). It also helps existing users to adopt new features quickly. A user manual, quick-start guide, and video demonstrations have been developed and are updated for each new version of AI4Green. In-person or virtual training is provided when onboarding new research groups. When hosted virtually, these meetings can be recorded so they can rewatch the video if needed. There is also an interactive tutorial where users can create reactions. An email address is provided on the website that users are encouraged to email if they need help or have any queries or suggestions.

Table 4.12: Training and user support features required for a digital research environment.[132]

Feature	Status in AI4Green
Training	Implemented
User Documentation	Implemented

4.4.2.12 Overview

The DRE analysis of requirements produced by Kanza and coworkers provides insight into how AI4Green can contribute to a DRE where features require implementation into AI4Green and where integration with external software is required.[132] This analysis demonstrates that AI4Green contains all the core and basic functionality required to be used as an organic chemistry ELN, and its growing user base and

improved user retention reflect this. AI4Green's generic features are robust, which is vital as these can form the first impressions of a new user. The note booking and domain features are essential for daily use, and overall, these features are implemented. Data export is a strength of AI4Green, as many ELNs do not support exporting machine-readable formats, which are essential for FAIR data.

However, there are also demonstrated areas where necessary functionality is lacking. Whilst this functionality does not prevent the use of AI4Green, incorporating this functionality through new features or the required integrations will enable users to improve their workflows and data management. Some aspects were lacking, such as the equipment interface, which, whilst requiring significant work, would be very powerful when using the application on an institutional level. Additional integration with equipment interfaces would support consistent data collection by automatically adding attachments to an experiment, enhancing user efficiency and data integrity. API access is a crucial aspect currently lacking and prevents the partial integration of AI4Green into other systems. The codebase is built with an open-source ethos with the idea that others can use it, advance it, and tailor it to their needs; incorporating API access aligns with this ethos and increases the project's impact. API access would be a prerequisite for integrating some functionalities into a DRE. For example, facilitating data exchange with other systems. Data import is another feature which is critically lacking and requires implementation. A flexible data import function would enable researchers to transfer data from previous ELNs or other software. A lossless and seamless data import can be incredibly challenging as data stored in AI4Green would likely differ in structure from the imported data. For example, AI4Green

emphasises sustainability and collects related data, which many other systems do not. Although challenging, a basic data import would capture the basic facts of previous experiments being imported, such as the compounds involved, the product, and the outcome data. Where a complete data transformation is not possible, file attachments can offer a sub-optimal but versatile means of importing data which ensures important data are not lost. Additionally, these challenges demonstrate the need for controlled vocabularies and common standards to enable interoperability. This is part of the future vision of the project and the ELN consortium, which aims to bring about interoperability and FAIR data within the ELN community.

Two identified and planned recommendations for future work are revamping the experiment planning functionality and establishing a data pipeline to a downstream repository. Updating experiment planning will provide significant quality-of-life improvements to chemists and add more flexibility to the domain features. It will allow fields for additional information, enabling better planning and data collection for methods such as optimisation reactions and HTE. Related to this is the potential for users to create and share templates of different experiment types. This enhancement improves the user experience by allowing more experiment planning and documentation variation and expands the software's ability to support complex workflows. In the backend and database layer of the application, additional fields would increase the structure of the data, as less data would be in unstructured free text. This could strengthen workflows, including searching, import/export and publication functionalities. Searching will be improved by enabling more specific search filters. Imports from non-specialist organic chemistry ELNs can be better

represented, and exports can include additional machine-readable data due to the increased labelled data collection. These upgrades also enable the possibility to generate human-readable exports aimed at publication. It is sensible to do this upgrade before creating a pipeline to a data repository to maximise the quality of the data exported down the pipeline. The ORD has been identified as a suitable repository, and most work here relies on saving data in a structure that can integrate with their existing pipeline for data upload. A proposed format for this pipeline is shown in Figure 4.5.

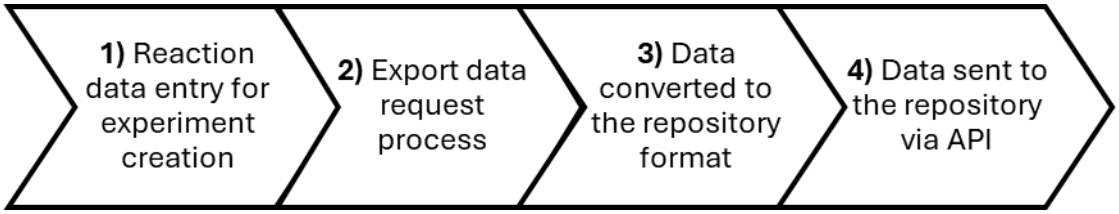


Figure 4.5: The proposed pipeline for exporting data to an external repository, such as the open reaction database. **1)** The user must enter reaction data during experiment creation, and this step is already present within the ELN. **2)** The export data request process also already exists within the ELN for the user to receive data in a format of their choice, and approval is required from all principal investigators within a workgroup. **3)** The data must then be converted to the repository format. No such mechanism yet exists within the ELN. **4)** The final step is sending this data to the repository using an API.

Another aim is to make the software entirely suitable for a professional environment. Features such as audit trails may not seem exciting, but they are essential for the professional uptake of the application. Fortunately, many aspects required for professional software align with the requirements of a DRE, paving a clear path forward for future development of the application.

Many of the data features are partially implemented and require further work, some of which are planned. Having features related to data storage before submission to a repository is essential, as promoting FAIR data is one of the major project aims.

4.4.3 User Base

The number of users is an important metric and can be measured in a few different ways. The total number of users is shown in Figure 4.6. However, the number of total registered users is significantly greater than the number of active users. Possible reasons are people testing the application, downloading the source code, and running an instance themselves. This can be due to user concerns and policies over data transfer. Additionally, users may find the application unsuitable for their needs if they primarily work in an area outside of organic chemistry. Active users are currently recorded as users who have made a reaction within the last month. However, users who have logged in within the last month may provide a better alternative to capture users, such as PIs who may not make any reactions but still view others' reactions.

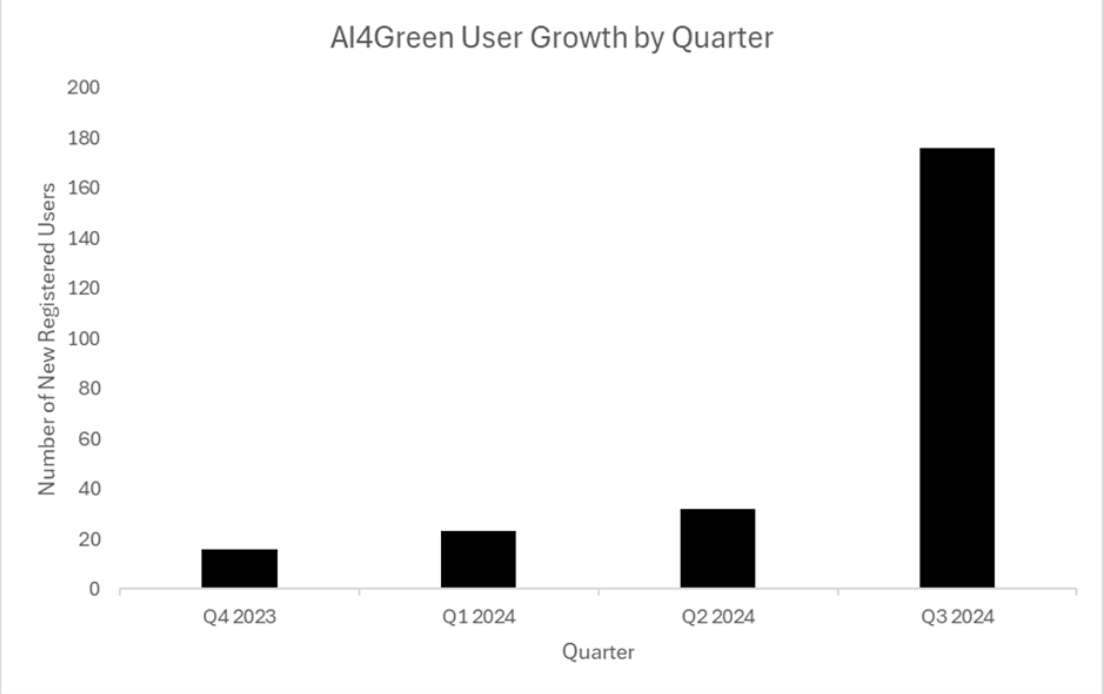


Figure 4.6: Number of new users registered in each quarter.

Another method is to measure the number of reactions. The rate of reaction creation has increased since the initial deployment of the application. At the time of writing, 70 reactions were created in the last month to give 507 reactions in the database.[141] If we assume this rate of reaction creation to remain constant, then 840 reactions should be added within a year. However, it is not overly optimistic to expect this rate to continue growing. The number of reactions created reflects user retention and engagement. It also better measures the aims of the application, which are to help chemists perform reactions more sustainably and improve their data management.

Increasing awareness of AI4Green is important to grow the user base. A software note has been published, and the codebase is on GitHub.[142] There is also an annual stakeholder forum and promotion to select research groups. AI4Green is listed on the ELN-Finder website. QR codes are also generated for presentations and posters at conferences and events. It is possible to use Google Analytics to find the source of where people click from.

Many users are undergraduate users whose teaching leader was interested in the application. There are many potential benefits to introducing an AI4Green-like tool into undergraduate curricula, such as promoting and educating about sustainability and teaching students how to use an ELN as part of modern-industry requirements. A related AI4Green4Students web application is currently being developed within the research group.

4.5 Conclusion

In this chapter, the ELN landscape has been analysed, identifying key barriers to adoption, particularly cost, time, and concerns about product reliability. AI4Green has been designed to minimise these barriers by being open-source, free, and accessible through a web browser. Collectively, these attributes reduce the time and cost of adoption for academic research groups.

A comprehensive feature review from Kanza and co-workers was used to evaluate AI4Green's integration into a DRE, finding strengths in domain-specific support and data export capabilities while also identifying areas for further development.[132] Planned future work includes making a more flexible reaction constructor to improve user experience and data recording. This would also provide a foundation for improving interoperability and establishing a robust data pipeline to a reaction repository. These improvements aim to increase product quality to better attract new and retain existing users and facilitate data-sharing in line with the FAIR principles.

Growing AI4Green's user base is a core objective, with early data showing promising trends in user adoption and retention, particularly when entire research groups have been onboarded and engaged with the ELN. Targeted outreach efforts, such as training materials and demo sessions, have proven effective for onboarding new users.

With a large number of academic chemists not yet using an ELN, AI4Green has scope to make a significant impact in chemistry by promoting sustainable practices, improving data management, and accelerating workflows. As five years remain on the

research grant, securing either longer-term funding or becoming self-sufficient is essential for the costs of servers, administration, and the continued development of AI4Green. If this challenge can be overcome, there is potential to establish an ELN that can be used across universities to promote sustainability and good data practices within the chemical sciences, addressing two of the most pressing challenges in chemistry today.

Chapter 5: Integration of a Retrosynthesis Tool with Sustainability Metrics into the AI4Green Electronic Laboratory Notebook

Abstract

This work presents a software tool integrated into the AI4Green ELN that predicts a series of routes and conditions to synthesise a target molecule and then assesses the sustainability of the reactions and routes. The sustainability assessment is adapted from CHEM21, and a weighted median is introduced to support sustainability comparisons of reactions and routes. Alternatively, routes can be entered by file upload by adapting a CSV template. These are supported by predictive machine learning models that can predict a retrosynthetic route and forward reaction conditions for a target molecule. A cloud-hosted instance of the retrosynthetic planning software AiZynthFinder is used to predict routes to the target molecule using a stock file and policy trained on USPTO data. A cloud-hosted instance of the ASKCOS context recommender tool is then used to predict conditions for each forward reaction in these routes. The routes and conditions are combined and assessed using suitable CHEM21 metrics to provide colour-coded feedback at the route and reaction level. Additionally, users can adjust how much these metrics contribute to the weighted median by moving a slider for each metric to help achieve their specific goals. The integration within the AI4Green electronic laboratory notebook provides

beneficial features such as import and export from reactions, use of the AI4Green compound database, and saving with the ability to share with other project members.

5.1 Introduction

5.1.1 Retrosynthesis

Chapter 1 introduces the concepts of CASP and the computational approaches associated with retrosynthesis and condition prediction. Applying AI to retrosynthesis has been a research focus for many decades, with work first published in the 1960s on retrosynthetic software.[143] There are two main approaches to retrosynthesis software. One is knowledge-based or symbolic AI, where software is trained on hand-encoded rules; Synthia (formerly Chematica) is a notable example.[144] The other is machine learning, with methods developing mostly from 2018 onwards following Segler's seminal publication.[40] The development of datasets such as the open-source USPTO reaction dataset was a significant milestone, as machine learning approaches rely on data quality and volume.[41] Numerous machine learning-based retrosynthesis software tools have been developed and refined in recent years.[145] Chemical space is incredibly vast; hence, creating a knowledge base capable of navigating it is a significant challenge. High-quality data, appropriate representations, and algorithms are all required and are areas of development for retrosynthetic software.

5.1.1.1 Symbolic AI Approaches

The software Synthia has been in development for nearly 20 years and is the most eminent example of retrosynthetic software, which uses hand-encoded rules. It has over 100,000 rules and has been successfully applied to finding routes for medicinally relevant targets and complex natural products.[146–148] This is a significant challenge and validates its approach. Limitations include the naturally slow development of

increasingly complex hand-encoded rules and the software's inability to operate outside these rules. This approach gives no mechanism by which generalities can be learnt and begin to understand how chemistry can be applied. This limitation prevents the possibility of the system proposing novel chemistry.

5.1.1.2 Machine Learning Approaches

Machine learning or data-driven approaches can be split into two categories: template-based or template-free approaches.

5.1.1.2.1 Template-free Approaches

Template-free approaches are trained on reaction data and use generative AI methods to predict routes using molecular representations such as strings (often referred to as sequences in this context) or graphs. These methods typically use sequence models such as RNNs or transformers, similar to those used in large language models. The architecture can be tailored to the molecular representations being inputted and outputted. Different representations have their advantages and disadvantages. Graphs, for example, capture connectivity as molecular graphs represent atoms as nodes and bonds as edges. However, this requires additional data to describe features such as stereochemistry. This template-free retrosynthesis method proposed a modified version of a graph attention network and used this in a graph-to-sequence model.[149] Sequence-to-sequence, graph-to-graph, and other similar architectures have also been used.[150–152] SMILES are the most widely used string representation in cheminformatics. However, SELF-referencing embedded strings (SELFIES) were proposed in 2020 as they guarantee a valid molecule and may be used more in future applications that output a string.[153] These rules-free approaches can output

transformations from outside of the training data. Theoretically, this means that if the underlying principles behind possible transformations are learnt, novel transformations could be proposed. Whilst this is an area of research, this is an extremely ambitious goal for the current generation of models.[154]

5.1.1.2.2 Template-based Approaches

Template-based approaches involve the model learning transformations from the training data. The model then seeks to apply these templates and selects the most likely one for a given molecule.[155] This method can also be described as a rules-based approach similar to hand-encoding rules, except training relies on a machine extracting data to form rules rather than a human expressing their expertise as rules and logic. This rules-based approach means that the model cannot predict transformations not seen in the training set and, therefore, cannot predict novel chemistry. Methods have also been proposed using sequence-to-sequence models to generate templates rather than reactants or synthons directly.[156]

5.1.1.3 Data

It is widely acknowledged that models are only as good as the data they are trained on. Table 5.1 shows some of the current databases used to train retrosynthetic models. Of these large reaction datasets, only the USPTO dataset is open access.[41] The USPTO dataset includes US patent data from 1976 to 2017. The lack of open reaction data is a challenge in the field, and the ORD is one initiative to improve this.[80] The ORD was set up to provide an open-access single-step reaction database for computational researchers to use for training machine learning models and for lab chemists to explore reactions with precedent. The ORD relies on industrial and academic chemists

to deposit their reaction data and provide a schema and workflow to aid with this. Upon release, the largest submission contained 400,000 reactions.[80] One advantage of the ORD compared to some databases formed from literature extraction is the presence of negative data. There is a culture of not reporting negative data (failed reactions) within organic chemistry publications. This is a problem when training machine learning models and for lab chemists who may repeat a known failed reaction due to its absence in publication. Segler et al. addressed this by generating a negative dataset and using this to supplement the original data when training a retrosynthetic model.[40]

Table 5.1: Selection of large databases recording chemical reactions. The table is taken from Weber et al.[12]

Database	Size	Accessibility
CASREACT	>128 million reactions	Proprietary
Reaxys	>46 million reactions	Proprietary
Pistachio	>9 million, patent literature based	Proprietary
SPRESI	>4.6 million reactions	Proprietary
USPTO	>3.3 million, patent literature based	Open

As the largest freely available reaction dataset, the USPTO dataset is frequently used in academically developed retrosynthesis tools. An evaluation of datasets and their influence on CASP tools was undertaken by Thakkar *et al.* They found that, unsurprisingly, the larger Reaxys dataset has a greater chemical space coverage than the USPTO dataset. However, this only translated to minor improvements in retrosynthetic solutions for the molecules used in this experiment. An additional comparison was made between how the different datasets fared in solving the compounds in the AstraZeneca virtual library and the top 125 pharmaceutical

compounds. Each dataset found more solutions for the compounds in the AstraZeneca virtual library than the top 125 pharmaceutical compounds. The authors suggested this was due to the greater natural product-like character and more complicated ring systems in the top 125 pharmaceutically relevant molecules.[157]

Current CASP tools trained on reaction data tend to perform better on pharmaceutical molecules than natural product molecules. This is not surprising as natural products are often of greater synthetic complexity than pharmaceuticals. Additionally, training data originating from ELNs, patents, and the chemical literature tends to be biased towards pharmaceuticals. The biosynthetic pathways that produce natural products use more C-C bond-forming reactions, which produce OH hydroxyl groups, meaning more asymmetric carbon centres are formed.[158] However, in medicinal chemistry, reactions that are reliably amenable to high-throughput chemistry, such as amide formation and Suzuki-Miyuara reactions, yield aromatic and achiral products.[159,160] Current retrosynthetic planner tools recognise these limitations and often apply maximum step counts. There is an acceptance that these molecules are beyond the scope of current data-driven tools, which often describe themselves as being for use in the pharmaceutical domain. Hyperparameter optimisation is a consideration for these tools and must balance the time and computational resources for a search against the quality of their output. Hyperparameters include maximum search time, depth of the tree search, and maximum step count. Research on their optimisation in AiZynthFinder has been published.[161] This reinforces the idea that current data-driven retrosynthesis tools can propose useful routes to pharmaceutically relevant molecules

but cannot design routes to more complex molecules, and these tools are below the level of experts in the field.

Multiple research groups have analysed the USPTO dataset. The data are imbalanced, with some reaction classes overrepresented and others underrepresented. This underrepresentation of many reaction templates is demonstrated by 48.3% of reactions being unique.[162] Another analysis looked at the specific reaction classes of the 1.1 million reactions included in patents from 1976 to 2013 (Figure 5.1).[163] This showed that the top three reaction classes make up 64.5% of reactions: substitution, acylation, and deprotections. Another challenge in using the USPTO dataset for CASP is changing working practices and aims over time. For example, there is now a much greater focus on sustainable chemistry and solvents like benzene and carbon tetrachloride are now typically only used as a last resort.

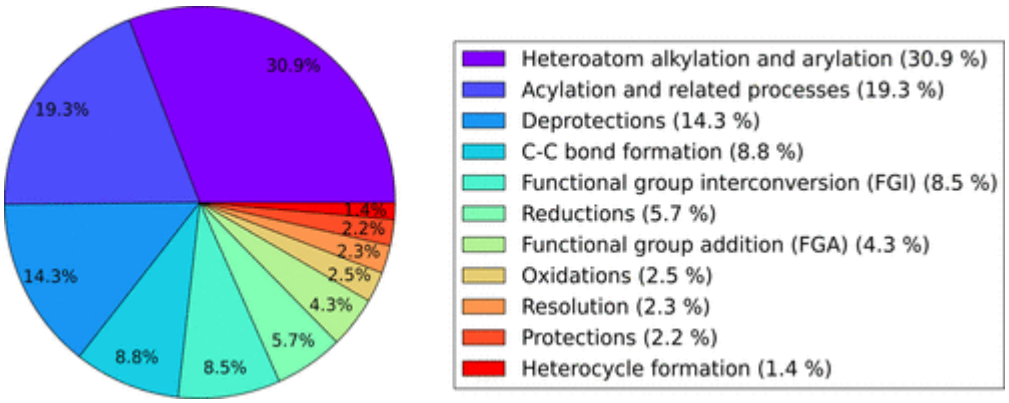


Figure 5.1: The distribution of reaction classes of the USPTO dataset, omitting the 46% of reactions that were not classified. Taken from Schneider et al.[163]

5.1.1.4 Single-Step and Route Planners

Most machine learning models solve for a single step. Planning the whole route is a more complex problem but also a more useful process. A single step might be easier but could result in longer or more challenging routes overall. Single-step models can

be integrated into a search algorithm with a stock list of valid building blocks to obtain a retrosynthetic route planner that outputs complete routes.

5.1.1.5 Algorithm

Segler *et al.* used a Monte Carlo Tree Search (MCTS) with their template-based method.[40] MCTS is a heuristic search algorithm consisting of four steps. It has been used in similar problems where numerous possibilities can be described by creating tree-like structures and where the success of each move can be evaluated. Examples include games such as Chess and Go.[164] In retrosynthesis, the nodes in the tree represent molecules. The four steps (Figure 5.2) are selection, expansion, simulation, and backpropagation. Selection is where the exploitation of successful nodes and the exploration of underexplored nodes are balanced. This is commonly done by using the Upper Confidence Bounds Applied to Trees (UCT) algorithm. This equation uses the number of visits and the score of a node to give it a score. The score of the node, as determined by successful simulations, is also affected by the number of visits. To balance exploration and exploitation, a node with more visits is penalised relative to a node with fewer visits. Visits refer to times the node is used during the traversal to an expandable node during selection. A different UCT equation can be used to adjust how the search performs, such as increasing the priority given to exploitation over exploration. The most common form of the UCT equation is shown in Equation 5.1, where i is a specific node, w_i is the node value from successful simulations, n_i is the number of the times the node has been visited, N is the number of visits to the parent node, and c is the exploration constant. The value of c determines the balance of

exploitation and exploration. A larger *c* value biases the algorithm towards more exploration and a smaller *c* value biases the algorithm towards more exploitation.

$$UCT = \frac{w_i}{n_i} + c\sqrt{\frac{lnN_i}{n_i}} \tag{5.1}$$

In the expansion step, the selected node is then expanded. This is where the trained machine-learning model is used to select the chemical transformation or move that is predicted to be most successful. The simulation step involves continuing this process for each subsequent node made during the expansion step until the search criteria are met or time or iteration limits are reached. In retrosynthesis, the search criterion is whether or not the simulation results in a building block, defined as molecules within the stock list. Backpropagation then involves updating the scores of the newly expanded nodes and all nodes that preceded it on the way to the origin node. The score for a node is improved if a successful route to building blocks is found during the simulation step or lowered if no successful route is found. Selection then occurs again with the updated scores of the nodes. This balance of exploitation and exploration guides the search more efficiently than brute force or random searching, for example. This was a significant advancement in multi-step retrosynthesis planning and is an approach that has been applied in other work since then, including open-source software packages, ASKCOS, and AiZynthFinder.[40,90,165] The stock list comprises chemicals defined as building blocks, meaning commercially available starting materials reasonably priced for synthesis. This is subjective and project-dependent, with synthesis scale and budget strongly influencing a compound's acceptable price per gram. Version 4.0 of AiZynthFinder supports filtering by price, but these data

typically need to be purchased and are not freely available in a machine-accessible format.[166]

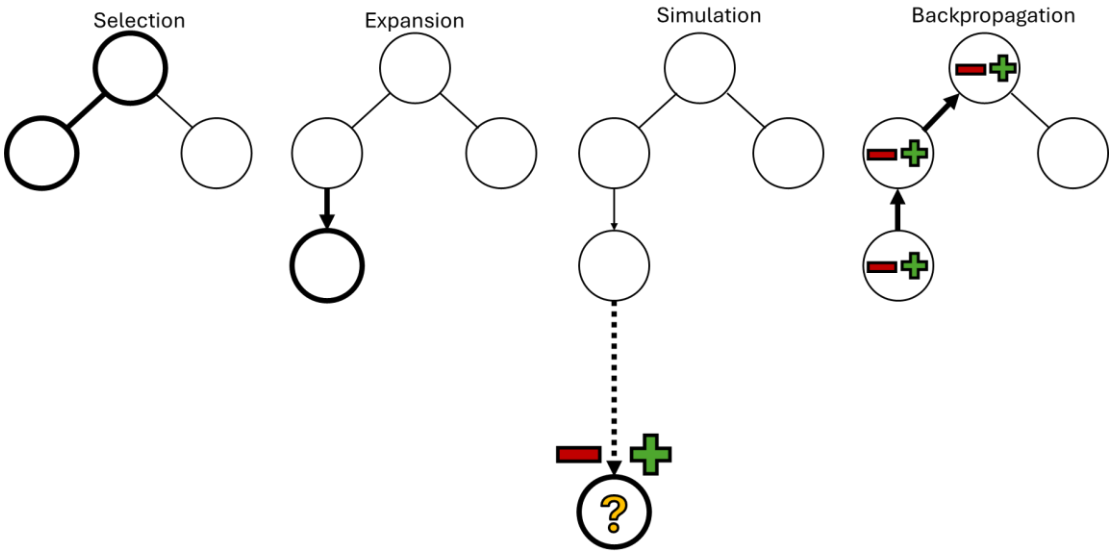


Figure 5.2: The four steps of a Monte Carlo Tree Search. **Selection**: a node is selected using the UCT equation to balance exploration and exploitation. **Expansion**: the selected node is expanded, adding to the tree search. **Simulation**: the expanded node is simulated until a terminal position is reached, and this is assessed. For retrosynthesis, a successful simulation results in molecules belonging to the stock file. **Backpropagation**: the scores of the preceding nodes are updated positively or negatively, depending on the outcome of the simulation. Selection occurs again with the now updated node scores.

5.1.1.6 Evaluation

As more retrosynthetic tools are developed, the need for frameworks to evaluate and compare these tools has grown. The evaluation of a single-step retrosynthesis model differs from that of multi-step retrosynthetic planners, with each having unique and shared challenges. One common challenge is the use of training, testing, and validation data splits, which is typical in many machine learning applications. The validation split is reserved for testing the model after training. In retrosynthesis this means either reference reactions for a single step model or reference routes for a multi-step planner. However, molecules can be synthesised in multiple ways. This is a

significant limitation when trying to assess models. To partially address this, top-N accuracy metrics are commonly used. These measure whether the expected reference reaction or route appear within the top N results proposed by the model, with typical values of N being 1, 5, or 10. Stronger validation methods include assessments on the predicted routes and reactions by expert chemists or experimental validation in the lab to determine if they are synthetically viable. However, these methods have their own challenges, as they are more costly in terms of time and resource.

PaRoutes is an evaluation framework published by the authors of AiZynthFinder.[167] This framework only supports route evaluation but allows any template-based single-step model and search algorithm to be part of the multi-step route planner. Routes were extracted from the USPTO dataset, and PaRoutes provides two benchmark datasets with a stock file for each. Each benchmark dataset has a series of routes for training and benchmarking, ensuring that the reactions used in the benchmarking routes were not in the training set. A stock file is also provided for each of the benchmarking datasets. 10,000 routes are provided for benchmarking, and the model is assessed based on its search time, number of solved targets, route diversity, and ability to find the reference route used in the USPTO data.[167] More recently, SYNTHESEUS, a retrosynthesis package with a benchmarking framework included was released by Segler and co-workers.[168]

5.1.2 Condition Prediction

Retrosynthesis applications have received the most attention within the CASP field but are only one piece of the puzzle. Retrosynthesis is used to elucidate the compounds that contribute atoms to the desired target molecule. However, this ignores the

necessity of introducing controlled reactivity to make the intended product, the energy required to break and make bonds, and to dissolve all the participating species in a reaction within the same solvent medium. The conditions necessary to achieve the desired transformation depend on the specific reactants and product being made. Coley and co-workers have developed and published condition prediction models as part of their many CASP software packages within the open-source ASKCOS platform.[43,104,155] These models are trained on approximately 10 million examples from the Reaxys database, and accept reactants and products as inputs and output predicted conditions, including the temperature, solvent, reagents, and catalyst.[43] These models have also been used as part of autonomous robotic synthesis platforms.[169] Another example of a CASP platform is IBM's closed-source RXN4Chemistry, which has a publicly available Python wrapper that costs $88,500 a year for ten users with unlimited API access.[170] These cloud-based AI models are also part of IBM's RoboRXN project for automated synthesis.[171]

5.1.3 Sustainability in CASP

Sustainability has so far been underrepresented in CASP research and tools. The majority of tools do not integrate sustainability or optimise for it beyond shorter route lengths, though there are a few examples. Work has been done on altering the MCTS algorithm and using a neural network for solvent prediction to predict shorter routes with greener solvents.[42] Additionally, Merck has integrated DOZN, a quantitative green chemistry evaluator, into Synthia and published case studies in which it found improved routes to known compounds.[172]

The case can be made that these models enhance sustainability through intelligent suggestions, which can help accelerate the process of finding suitable methods for synthesis and reduce trial-and-error approaches, either in retrosynthetic routes or predicted conditions. However, this assumes the model's predictions are superior to the chemists', which is unlikely, and ignores the energy costs associated with training a model, which can vary wildly. Computational approaches to reaction optimisation have a much stronger case to make on these grounds; multitask Bayesian optimisation has been demonstrated to accelerate condition optimisation.[173] However, there is a gap where more can be done to integrate sustainability into CASP tools, particularly in open-source applications.

5.1.4 This Work

One of the advancements this work makes is the integration of CASP tools within the free-to-use and open-source AI4Green ELN. This tool is designed to complement the existing suite of interactive digital tools which focus on supporting chemists and encouraging sustainable chemistry.[142] Sustainable chemistry requires holistic thinking, and currently, within the AI4Green platform, metrics are only visible at the reaction level. It is more holistic to think about the sustainability of the entire route than a single reaction. CASP offers a robust foundation by presenting multiple routes and conditions, enabling chemists to apply their intuition in selecting options while also considering the sustainability of each choice. Chemists frequently use services such as Reaxys and Sci-Finder to find reaction pathways, conditions, and literature precedent for their synthetic targets. Both now also offer retrosynthetic analysis of targets in addition to their search for reaction schemes in the literature. Integration within

AI4Green enhances a chemist's workflow by minimising duplicative data entry, aligning with the goal of creating a one-stop solution for chemists. ASKCOS is an example of a website with multiple CASP tools, including retrosynthesis. Integrating a retrosynthesis tool within the ELN reduces the chemist's workload by removing the need to enter data multiple times. The methods and implementation of the integration of CASP tools for route planning with a sustainability assessment are described.

5.2 Methods and Implementation

The core components of this work are the AI4Green ELN, a cloud-hosted instance of AiZynthFinder and ASKCOS's condition prediction model, and the development of an interface in AI4Green to show the adapted CHEM21 sustainability assessment.[43,142,166] The AI4Green ELN was developed prior to this work and is discussed in the previous chapter.

Version 4.0 of AiZynthFinder has been integrated into a Flask application. An endpoint accepts a SMILES string as a target and then uses AiZynthFinder to solve the retrosynthesis and return the solved routes as JSON. This method enables the addition of future arguments from AI4Green to the endpoint, such as user-specified stock lists. The code for this application is stored in a GitHub repository. ASKCOS's condition prediction tool is a FastAPI application provided by the Machine Learning for Pharmaceutical Discovery and Synthesis Consortium (MLPDS) at MIT on GitLab. Both of the applications are containerised using Docker. The containers are hosted on Docker Hub and deployed as Azure Container apps. A schematic of the data flow between these applications is shown in Figure 5.3. These container apps are serverless, meaning they only run when a call is made to them, reducing costs. This uses a type of microservice architecture as each application, AI4Green, retrosynthesis model, and conditions predict model, all run on separate servers. This architecture type can be useful to isolate functionality to simplify environment requirements and prevent an issue with one functionality interfering with another. The retrosynthesis model, in particular, is relatively computationally demanding when using a large stock

file, and this hosting method prevents this, stopping the AI4Green ELN from crashing if an error occurs.

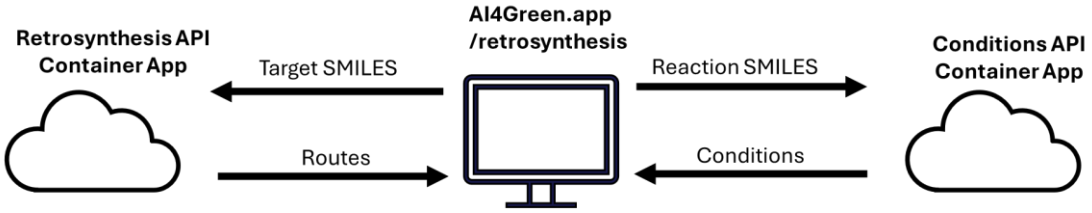


Figure 5.3: Schematic for the architecture showing the data exchange between the AI4Green web application and the APIs of the conditions and retrosynthesis microservices.

The Python package Plotly Dash Cytoscape was used to display the route data. This displays a collection of connected nodes where each node is a molecule, and the edges linking the nodes denote a retrosynthetic relationship. The Dagre layout arranged the nodes as a treelike structure well suited for retrosynthesis with the target molecule at the top. A non-terminal node will be connected to one or more nodes below, and this iteratively continues until terminal nodes are reached. In chemical terms, a product is connected to one or more reactants below until molecules designated as building blocks by the stock file are reached.

The metrics used are the same as those in the CHEM21 reaction assessment, excluding metrics that cannot be applied before the experiment is carried out. These include the purification method, catalyst recovery, and all yield-based metrics. The temperature, solvent, stoichiometry, element sustainability, atom economy, and safety are all assessed based on the predicted route and conditions. Each reaction is individually assessed on these metrics, and each value is colour-coded. Every reaction in a route can also be colour-coded by taking a weighted median of the metrics, where sustainable is equal to 1, problematic 2, hazardous 3, and highly hazardous 4. Each

metric has a corresponding slider that the user can assign between zero and ten; if the weights are all equal, it operates as an ordinary median. The total sum of weights is calculated, and the median or 50th percentile weight is found. The cumulative weights are calculated and the weighted median is the first value where the cumulative weight is greater than or equal to the median obtained in the previous step.

This is extended to colour-coding routes in the selection dropdown by calculating the median of all of the weighted medians of reactions belonging to the route. Whilst a "final score" was intentionally excluded from the original CHEM21 publication to encourage a holistic perspective, this work maintains this spirit.[17] The route's sustainability can be seen in a grid of reaction steps and metrics to provide a holistic perspective. Extending it to colour-coding the route in the dropdown helps users process the significant volume of data generated. Using a weighted median also enables a focus on the metrics they consider the most important. An example is solvent selection, as solvents typically contribute a greater mass of reaction waste than any other component.[174]

The integration within the AI4Green ELN enhances chemists' digital research environment and provides an improved user experience. Results can be saved to the database and accessed by other workbook members, negating the need for an additional account and enabling secure sharing. The database also retrieves compound data for the compounds involved in the predicted route. This provides hazard codes for molecules, which are necessary for assessing safety as part of the wider sustainability assessment. Importing and exporting between reaction planning and retrosynthesis is possible, reducing duplication of data entry.

Retrosynthesis and predicted conditions are exciting scientific areas at the frontier of CASP research, and their inclusion in this software allows chemists to interact with these models as part of their workflow. However, their limitations are known, and a chemist may want to use the route sustainability assessment for their chosen route. To this end, chemists can adapt a template CSV file and add their solvents, temperatures, reagents, reactants, and products for a route and upload it. This gives a sustainability assessment using the same method.

5.3 Results & Discussion

A series of case studies were conducted by applying the retrosynthesis and sustainability assessment to drug molecules. The predicted synthetic routes for these compounds were compared to the unoptimised and optimised synthetic routes reported in the literature.[175] The routes suggested come from AiZynthFinder. The primary focus is not to evaluate AiZynthFinder's routes but to evaluate the integrated synthesis planning and sustainability assessment module. CASP software will likely continue to improve, and the microservice architecture supports the ability to update and use new, better-performing retrosynthesis or condition prediction models as these are released. For these predicted routes and conditions, AiZynthfinder used models trained on USPTO data and searched to a maximum of up to ten transformations with a time limit of one hour but always less due to an iteration limit of 500. The Zinc stock file was chosen. eMolecules only and eMolecules plus Zinc were also trialled. However, eMolecules stock was not used as it contained complex drug-like intermediates, resulting in single-step compound transformations that would exceed the expected cost of synthetic building blocks.

5.3.1 Case Study 1: Sildenafil Citrate

Sildenafil Citrate, commonly known as Viagra, is a well-known drug patented by Pfizer in 1996. It is used as a case study as a pharmaceutically relevant compound. The predicted route in Figure 5.4 shows a three-step synthesis to make Sildenafil.

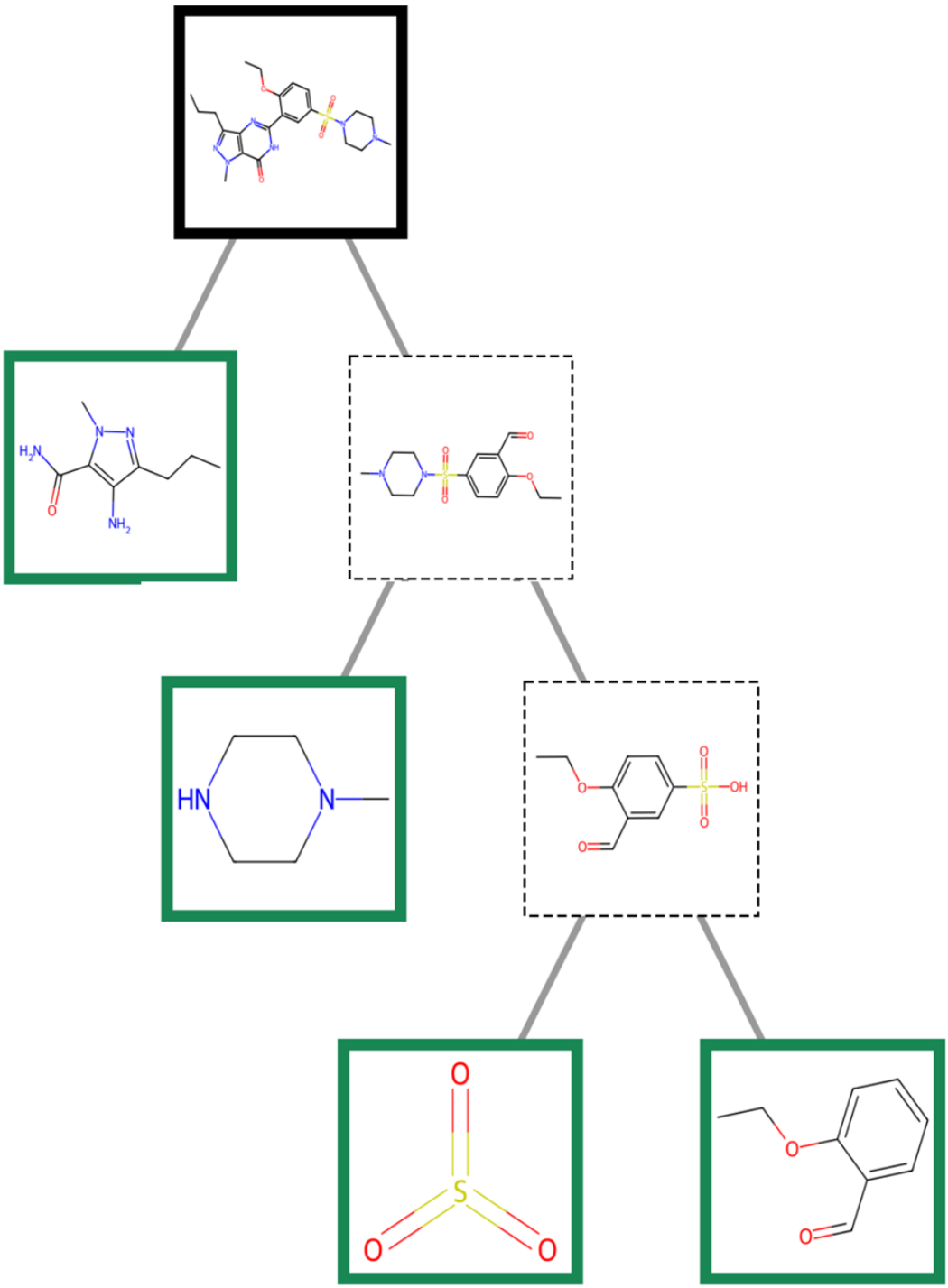
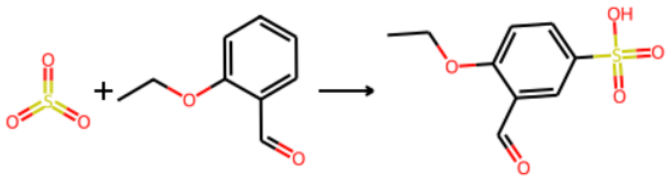


Figure 5.4: Predicted retrosynthetic route for sildenafil citrate.

The first step is a sulfonylation reaction using sulfuric acid (as a reagent shown in the predicted conditions in Figure 5.5) and sulfur trioxide. The predicted conditions for this transformation are shown in Figure 5.5. The second step is forming the sulfonamide, and the third step is a cyclisation reaction.



Condition Set 1

	Conditions & Sustainability
Temperature (°C)	55
Solvent	No solvent
Reagents	Sulfuric Acid
Stoichiometry/Catalyst	No catalyst
Element Sustainability	50-500 years
Atom Economy	70.1%
Safety	H314, H318, H330, H335, H351, H411, H315, H319, H335, H314

Figure 5.5: Predicted sulfonylation conditions in the synthesis of sildenafil citrate.

These are all known reactions, which makes sense as the retrosynthesis method used is template-based. Therefore, these are all recognised templates from the retrosynthesis policy model trained on USPTO data. However, whether these theorised disconnections could occur in reality under the predicted conditions is unknown. A chemist using this software would be required to use a mix of intuition

and cross-referencing to see whether similar transformations have precedent in the literature. An indication of the sustainability of the suggested route and each reaction is also provided.

Figure 5.6 compares the sustainability of the reported unoptimised and optimised routes for Sildenafil Citrate. This was done by finding these routes reported in the literature, converting them to a CSV, and uploading them into the route planning module of AI4Green. This analysis shows that the optimised route is more sustainable than the unoptimised route in most aspects; this can be most readily seen in the weighted median sustainability of each step. This was the expected outcome, and the software provides a visualisation that can help compare the sustainability of the routes in a way that someone without specialist chemistry knowledge could also understand.

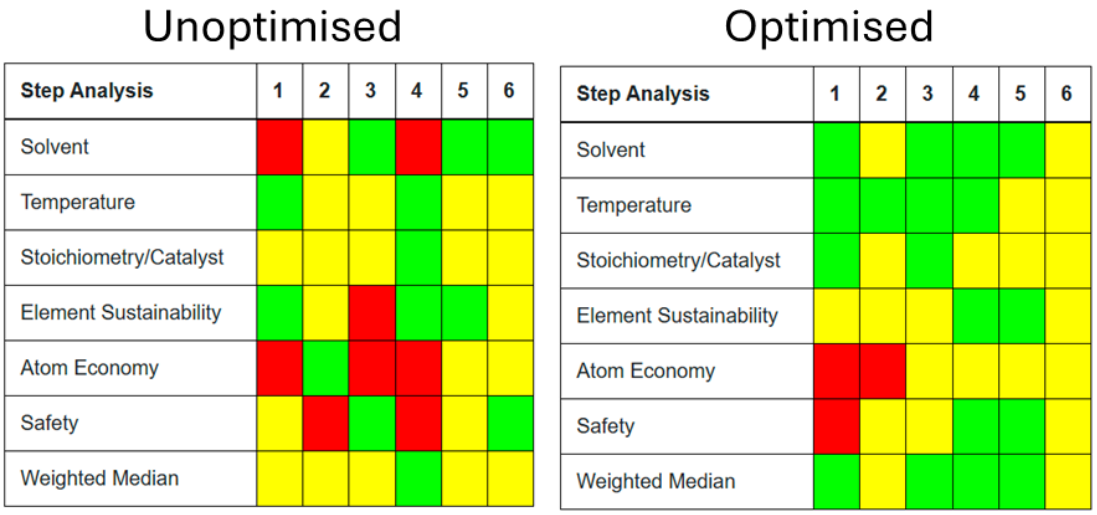


Figure 5.6: The unoptimised and optimised medicinal chemistry routes used in the synthesis of Sildenafil Citrate as reported in the literature.[175]

5.3.2 Case Study 2: Ibuprofen

Ibuprofen is a famous drug discovered by chemists at Boots in the 1960s.[176] The initial synthesis involved a six-step route. However, a 1990 patent by Boots-Hoechst-Celanese (BHC) published an optimised three-step route to ibuprofen.[177] Figure 5.7 compares these routes. Temperature and solvent data were not readily available for all steps, so these were omitted from the calculated weighted median by adjusting their sliders to zero. Both routes show similar colour assessments, but the optimised route is half the length whilst using the same starting material and, therefore, can be considered more sustainable. Individually highlighting each metric can help the chemists proceed in the best way possible. For example, the improved route still has safety concerns due to the use of harmful chemicals such as hydrofluoric acid. This means that safety considerations must be at the forefront when implementing this route.

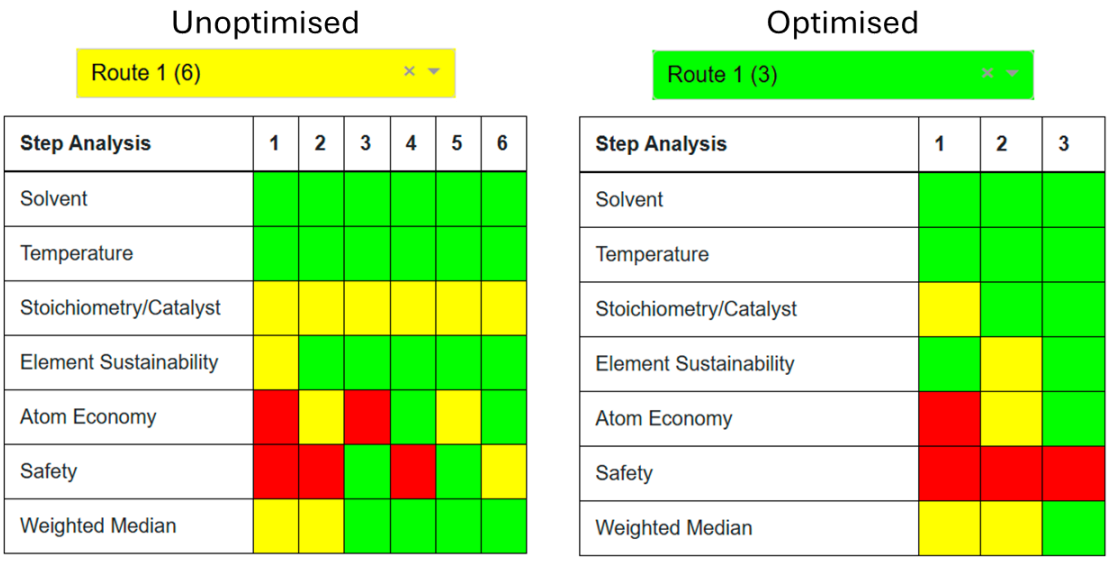


Figure 5.7: Comparison of initial Boots and BHC routes for the synthesis of ibuprofen.

5.4 Conclusions

The work presented in this chapter shows the integration of CASP tools, namely retrosynthesis and condition prediction, into the AI4Green ELN with a colour-coded sustainability assessment. The tool is made free and easy to use as part of AI4Green and is part of the ethos of breaking down the barriers of getting data-driven tools into the hands of lab chemists to help them improve their sustainability. The case study of Sildenafil Citrate demonstrates the utility of the tool in predicting routes to pharmaceutically relevant molecules and comparing the sustainability of different routes using either predicted or uploaded routes.

This work can be built upon significantly. Future iterations of this tool can include customised stock files as part of a planned integration with ChemInventory. This means that the user could choose for the retrosynthesis tool only to use chemicals the user has available in their laboratory as stock compounds. Additional work can also be done on developing more relevant stock files that better reflect what a chemist would include in their building blocks. Integrating sustainability directly into the search would be an interesting and beneficial development. The key to the further development of the CASP tools used is more high-quality reaction data. The higher quality and more modern the data, the better and, likely, more sustainable the outputs of these tools will be. Further integrations with AI4Green can be done. For example, a series of reactions within AI4Green that belong to the same route can be combined, and then the route's sustainability can be assessed. This would provide a route sustainability assessment that could include mass-based metrics dependent on the yield. In the

future, accurate generalised yield prediction models could be developed, and these would significantly aid the route sustainability assessment.

Chapter 6: Concluding Remarks

This work addresses two major challenges within synthetic chemistry: the need to improve sustainability and better data practices. Also highlighted are the importance of developing a complete digital research environment and the value of reaction data. The preceding chapters explore the current landscape of digital tools for sustainable chemistry and then describe work on applying digital methods, including machine learning and software development, to support chemistry with an emphasis on sustainability. Chapter One presents and evaluates the digital tools for green and sustainability chemistry. Chapter Two introduces machine learning and cheminformatics and describes the methods referenced in this work. Chapter Three presents research on applying machine learning to real-time reaction data to reaction yield. Chapter Four describes the role of ELNs, their current uptake and the development of our own ELN, AI4Green. Chapter Five presents the integration of various CASP tools into the AI4Green ELN and the development of an accompanying sustainability assessment.

The landscape evaluation reveals a range of tools available for chemists. However, getting these tools into the hands of chemists working outside the specific green chemistry field is a challenge. Synthetic chemists, whilst increasingly conscious of sustainability, typically have priorities that prevent investing a significant amount of time into studying the sustainability of their reactions. More work has been done within process chemistry, where sustainability is a greater priority. Top journals in the field, such as *Organic Process and Research Development* (OPRD), have demonstrated their commitment to this by requiring papers using “strongly undesirable solvents” to

have a convincing justification for their use.[178] However, there is a recognition that improving processes earlier is beneficial from an initial waste and cost reduction perspective, as well as reducing the workload of process chemists by providing a more optimised starting route. Therefore, the development of tools to encourage sustainability early on, such as the AI4Green ELN, is needed. The integration of sustainability into the core of the ELN means that there is minimal increase in workload for the chemist to measure the sustainability of their reactions. The integration of data-driven CASP tools with sustainability into AI4Green in Chapter Five follows previous integrations into AI4Green and contributes to an enhanced digital research environment. Further requirements for a digital research environment and methods to meet them have also been identified.[132]

The potential of data-driven tools is demonstrated in Chapters Three and Five. Reiterated throughout this work is the need for better reaction data management policies that result in machine-readable FAIR data. Chapter Four makes explicit the role of ELNs as data entry and storage tools. The reliance on paper lab notebooks demonstrates a need for an ELN suitable for the barriers that academia faces. AI4Green offers a solution for this and aims to provide a superior experience to paper lab notebooks and competitor ELNs. The key is that the ELN is a free-to-use browser-based tool, removing cost and elaborate setup from the equation.

This thesis presents digital methods for sustainable chemistry with the intent to modernise workflows to prioritise sustainability and data management. The foundations this work builds are to capture reaction data at the source, and it is

anticipated this will contribute to the further development of intelligent tools that can aid chemists' sustainability.

Bibliography

1. P. T. Anastas and J. C. Warner, *Green chemistry: Theory and Practice*, Oxford University Press, Oxford [England]; New York, 1998.
2. UNESCO, *UNESCO science report: towards 2030*, Paris, France, 2015.
3. M.-L. Tseng, C.-H. Chang, C.-W. R. Lin, K.-J. Wu, Q. Chen, L. Xia and B. Xue, *Environ. Sci. Pollut. Res.*, 2020, **27**, 33543–33567.
4. Department for Energy Security and Net Zero and Department for Business, Energy & Industrial Strategy, *Net Zero Strategy: Build Back Greener*, 2022.
5. H. F. Sneddon, in *Green Chemistry Strategies for Drug Discovery*, The Royal Society of Chemistry, 2015, pp. 13–38.
6. E. J. Corey, H. W. Orf and D. A. Pensak, *J. Am. Chem. Soc.*, 1976, **98**, 210–221.
7. D. A. Evans, *Angew. Chem. Int. Ed.*, 2014, **53**, 11140–11145.
8. I. N. Derbenev, J. Dowden, J. Twycross and J. D. Hirst, *Curr. Opin. Green Sustain. Chem.*, 2022, **35**, 100623.
9. R. A. Sheldon, *ACS Sustain. Chem. Eng.*, 2018, **6**, 32–48.
10. C. Jimenez-Gonzalez, C. S. Ponder, Q. B. Broxterman and J. B. Manley, *Org. Process Res. Dev.*, 2011, **15**, 912–917.
11. A. Zamagni, L. Zanchi, S. Di Cesare, F. Silveri and L. Petti, in *Life Cycle Engineering and Management of Products: Theory and Practice*, eds. J. A. de Oliveira, D. A. Lopes Silva, F. N. Puglieri and Y. M. B. Saavedra, Springer International Publishing, Cham, 2021, pp. 143–168.
12. J. M. Weber, Z. Guo, C. Zhang, A. M. Schweidtmann and A. A. Lapkin, *Chem. Soc. Rev.*, 2021, **50**, 12013–12036.
13. S. Herres-Pawlis, F. Bach, I. J. Bruno, S. J. Chalk, N. Jung, J. C. Liermann, L. R. McEwen, S. Neumann, C. Steinbeck, M. Razum and O. Koepler, *Angew. Chem. Int. Ed.*, 2022, **61**, e202203038.
14. S. Kanza, C. Willoughby, N. Gibbins, R. Whitby, J. G. Frey, J. Erjavec, K. Zupančič, M. Hren and K. Kovač, *J. Cheminform.*, 2017, **9**, 31.
15. E. Alladio, M. Baricco, V. Leogrande, R. Pagliari, F. Pozzi, P. Foglio and M. Vincenti, *Front. Chem.*, 2021, **9**, 734132.
16. D. Prat, A. Wells, J. Hayler, H. Sneddon, C. R. Mcelroy, S. Abou-Shehada and P. J. Dunn, *Green Chem.*, 2016, **18**, 288–296.
17. C. R. McElroy, A. Constantinou, L. C. Jones, L. Summerton and J. H. Clark, *Green Chem.*, 2015, **17**, 3111–3121.
18. T. V. T. Phan, C. Gallardo and J. Mane, *Green Chem.*, 2015, **17**, 2846–2852.
19. J. Płotka-Wasylka and W. Wojnowski, *Green Chem.*, 2021, **23**, 8657–8665.

20. D. A. Lopes Silva, A. O. Nunes, C. M. Piekarski, V. A. da Silva Moris, L. S. M. de Souza and T. O. Rodrigues, *Sustain. Prod. Consum.*, 2019, **20**, 304–315.

21. M. E. Kopach and E. A. Reiff, *Future Med. Chem.*, 2012, **4**, 1395–1398.

22. P. Dutta, A. McGranaghan, I. Keller, Y. Patil, N. Mulholland, V. Murudi, H. Prescher, A. Smith, N. Carson, C. Martin, P. Cox, D. Stierli, M. Boussemghoune, F. Barreteau, J. Cassayre and E. Godineau, *Green Chem.*, 2022, **24**, 3943–3956.

23. A. D. Curzons, D. C. Constable and V. L. Cunningham, *J. Clean. Prod.*, 1999, **1**, 82–90.

24. D. Prat, J. Hayler and A. Wells, *Green Chem.*, 2014, **16**, 4546–4551.

25. H. Sels, H. De Smet and J. Geuens, *Molecules*, 2020, **25**, 3037.

26. M. González-Miquel and I. Díaz, *Curr. Opin. Green Sustain. Chem.*, 2021, **29**, 100469.

27. M. J. Kamlet, J. L. Abboud and R. W. Taft, *J. Am. Chem. Soc.*, 1977, **99**, 6027–6038.

28. P. G. Jessop, D. A. Jessop, D. Fu and L. Phan, *Green Chem.*, 2012, **14**, 1245–1259.

29. J. Sherwood, J. Granelli, C. R. McElroy and J. H. Clark, *Molecules*, 2019, **24**, 2209.

30. C. M. Hansen, *Hansen Solubility Parameters: A User's Handbook, Second Edition*, CRC Press, Boca Raton, FL, 2007.

31. J. Clark, T. Farmer, A. Hunt and J. Sherwood, *Int. J. Mol. Sci.*, 2015, **16**, 17101–17159.

32. S. Boobier, D. R. J. Hose, A. J. Blacker and B. N. Nguyen, *Nat. Commun.*, 2020, **11**, 5753.

33. J. Kern, S. Venkatram, M. Banerjee, B. Brettmann and R. Ramprasad, *Phys. Chem. Chem. Phys.*, 2022, **24**, 26547–26555.

34. L. Fleitmann, J. Kleinekorte, K. Leonhard and A. Bardow, *Chem. Eng. Sci.*, 2021, **245**, 116863.

35. P. Harten, T. Martin, M. Gonzalez and D. Young, *Environ. Prog. Sustain. Energy*, 2020, **39**, 13331–13331.

36. L. J. Diorazio, D. R. J. Hose and N. K. Adlington, *Org. Process Res. Dev.*, 2016, **20**, 760–773.

37. D. Paurat and T. Gärtner, in *Machine Learning and Knowledge Discovery in Databases*, Springer Berlin Heidelberg, 2013, pp. 672–676.

38. S. Boobier, J. Heeley, T. Gaertner and J. Hirst, *ChemRxiv*, 2024, DOI: 10.26434/chemrxiv-2024-nv0r4.

39. B. Mikulak-Klucznik, P. Gołębiowska, A. A. Bayly, O. Popik, T. Klucznik, S. Szymkuć, E. P. Gajewska, P. Dittwald, O. Staszewska-Krajewska, W. Beker, T.

Badowski, K. A. Scheidt, K. Molga, J. Mlynarski, M. Mrksich and B. A. Grzybowski, *Nature*, 2020, **588**, 83–88.

40. M. H. S. Segler, M. Preuss and M. P. Waller, *Nature*, 2018, **555**, 604–610.

41. D. M. Lowe, Thesis, University of Cambridge, 2012.

42. X. Wang, Y. Qian, H. Gao, C. W. Coley, Y. Mo, R. Barzilay and K. F. Jensen, *Chem. Sci.*, 2020, **11**, 10959–10972.

43. H. Gao, T. J. Struble, C. W. Coley, Y. Wang, W. H. Green and K. F. Jensen, *ACS Cent. Sci.*, 2018, **4**, 1465–1476.

44. D. Probst, M. Manica, Y. G. Nana Teukam, A. Castrogiovanni, F. Paratore and T. Laino, *Nat. Commun.*, 2022, **13**, 964.

45. J. C. Kromann, J. H. Jensen, M. Kruszyk, M. Jessing and M. Jørgensen, *Chem. Sci.*, 2018, **9**, 660–665.

46. N. Ree, A. H. Göller and J. H. Jensen, *Digit. Discov.*, 2022, **1**, 108–114.

47. T. A. Soares, A. Nunes-Alves, A. Mazzolari, F. Ruggiu, G.-W. Wei and K. Merz, *J. Chem. Inf. Model.*, 2022, **62**, 5317–5320.

48. OECD, *Guidance Document on the Recognition, Assessment and Use of Clinical Signs as Human Endpoints for Experimental Animals Used in Safety Evaluation*, OECD Publishing, Paris, 2002.

49. D. J. Ponting, M. J. Burns, R. S. Foster, R. Hemingway, G. Kocks, D. S. MacMillan, A. L. Shannon-Little, R. E. Tennant, J. R. Tidmarsh and D. J. Yeo, in *In Silico Methods for Predicting Drug Toxicity*, ed. E. Benfenati, Springer US, New York, NY, 2022, pp. 435–478.

50. A. M. Bran, S. Cox, O. Schilter, C. Baldassari, A. D. White and P. Schwaller, 2023, arXiv:2304.05376.

51. D. Probst, 2022, arXiv:2210.00356.

52. Y. Shi, P. L. Prieto, T. Zepel, S. Grunert and J. E. Hein, *Acc. Chem. Res.*, 2021, **54**, 546–555.

53. J. Simm, L. Humbeck, A. Zalewski, N. Sturm, W. Heyndrickx, Y. Moreau, B. Beck and A. Schuffenhauer, *J. Cheminform.*, 2021, **13**, 96.

54. M. Garnelo and M. Shanahan, *Curr. Opin. Behav. Sci.*, 2019, **29**, 17–23.

55. R. K. Lindsay, B. G. Buchanan, E. A. Feigenbaum and J. Lederberg, *Artif. Intell.*, 1993, **61**, 209–261.

56. D. Goodman and R. Keene, *ICGA J.*, 1997, **20**, 186–187.

57. W. P. Wagner, in *Proceedings of the 1990 ACM SIGBDP conference on Trends and directions in expert systems*, Association for Computing Machinery, New York, NY, USA, 1990, pp. 247–261.

58. A. L. Fradkov, *IFAC-PapersOnLine*, 2020, **53**, 1385–1390.

59. J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S.

A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli and D. Hassabis, *Nature*, 2021, **596**, 583–589.

60. R. Bellman, *Science*, 1966, **153**, 34–37.

61. R.-C. Chen, C. Dewi, S.-W. Huang and R. E. Caraka, *J. Big Data*, 2020, **7**, 52.

62. D. E. Rumelhart, G. E. Hinton and R. J. Williams, *Nature*, 1986, **323**, 533–536.

63. S. Hochreiter and J. Schmidhuber, *Neural Comput.*, 1997, **9**, 1735–1780.

64. P. Branco, L. Torgo and R. P. Ribeiro, in *Proceedings of the First International Workshop on Learning with Imbalanced Domains: Theory and Applications*, PMLR, 2017, pp. 36–50.

65. L. McInnes, J. Healy, N. Saul and L. Großberger, *J. Open Source Softw.*, 2018, **3**, 861.

66. L. van der Maaten and G. Hinton, *J. Mach. Learn. Res.*, 2008, **9**, 2579–2605.

67. K. Pal and M. Sharma, in *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 2020, pp. 1106–1110.

68. K. Rajan, H. O. Brinkhaus, M. I. Agea, A. Zielesny and C. Steinbeck, *Nat. Commun.*, 2023, **14**, 5045.

69. K. Rajan, A. Zielesny and C. Steinbeck, *J. Cheminform.*, 2021, **13**, 34.

70. J. J. Vollmer, *J. Chem. Educ.*, 1983, **60**, 192.

71. A. Dalby, J. G. Nourse, W. D. Hounshell, A. K. I. Gushurst, D. L. Grier, B. A. Leland and J. Laufer, *J. Chem. Inf. Comput. Sci.*, 1992, **32**, 244–255.

72. D. Weininger, *J. Chem. Inf. Comput. Sci.*, 1988, **28**, 31–36.

73. G. Landrum, P. Tosco, B. Kelley, Ric, D. Cosgrove, sriniker, R. Vianello, gedeck, NadineSchneider, G. Jones, E. Kawashima, D. N, A. Dalke, B. Cole, M. Swain, S. Turk, A. Savelev, A. Vaucher, M. Wójcikowski, I. Take, V. F. Scalfani, D. Probst, K. Ujihara, guillaume godin, A. Pahl, R. Walker, J. Lehtivarjo, F. Berenger, strets123 and jasondbiggs, rdkit/rdkit (version Release_2023_09_5) Zenodo 2024.

74. S. Heller, A. McNaught, S. Stein, D. Tchekhovskoi and I. Pletnev, *J. Cheminform.*, 2013, **5**, 7.

75. J. M. Goodman, I. Pletnev, P. Thiessen, E. Bolton and S. R. Heller, *J. Cheminform.*, 2021, **13**, 40.

76. IUPAC-InChI/InChI: Main InChI repository, https://github.com/IUPAC-InChI/InChI, (accessed 6 September 2024).

77. J. Guo, A. S. Ibanez-Lopez, H. Gao, V. Quach, C. W. Coley, K. F. Jensen and R. Barzilay, *J. Chem. Inf. Model.*, 2022, **62**, 2035–2045.

78. A. C. Vaucher, F. Zipoli, J. Geluykens, V. H. Nair, P. Schwaller and T. Laino, *Nat. Commun.*, 2020, **11**, 3601.

79. G. Grethe, G. Blanke, H. Kraut and J. M. Goodman, *J. Cheminform.*, 2018, **10**, 22.

80. S. M. Kearnes, M. R. Maser, M. Wleklinski, A. Kast, A. G. Doyle, S. D. Dreher, J. M. Hawkins, K. F. Jensen and C. W. Coley, *J. Am. Chem. Soc.*, 2021, **143**, 18820–18826.

81. P. Tremouilhac, P.-C. Huang, C.-L. Lin, Y.-C. Huang, A. Nguyen, N. Jung, F. Bach and S. Bräse, *Chem. Methods*, 2021, **1**, 8–11.

82. P. Tremouilhac, A. Nguyen, Y.-C. Huang, S. Kotov, D. S. Lütjohann, F. Hübsch, N. Jung and S. Bräse, *J. Cheminform.*, 2017, **9**, 54.

83. W. Wang, N. H. Angello, D. J. Blair, T. Tyrikos-Ergas, W. H. Krueger, K. N. S. Medine, A. J. LaPorte, J. M. Berger and M. D. Burke, *Nat. Synth*, 2024, **3**, 1031–1038.

84. J. C. Davies, D. Pattison and J. D. Hirst, *J. Mol. Graph.*, 2023, **118**, 108356–108356.

85. Z. J. Baum, X. Yu, P. Y. Ayala, Y. Zhao, S. P. Watkins and Q. Zhou, *J. Chem. Inf. Model.*, 2021, **61**, 3197–3212.

86. A. Karthikeyan and U. D. Priyakumar, *J. Chem. Sci.*, 2022, **134**, 1–20.

87. L. B. Ayres, F. J. V. Gomez, J. R. Linton, M. F. Silva and C. D. Garcia, *Anal. Chim. Acta*, 2021, **1161**, 338403–338403.

88. S.-Y. Cho, Y. Lee, S. Lee, H. Kang, J. Kim, J. Choi, J. Ryu, H. Joo, H.-T. Jung and J. Kim, *Anal. Chem.*, 2020, **92**, 6529–6537.

89. B. Debus, H. Parastar, P. Harrington and D. Kirsanov, *TrAC, Trends Anal. Chem.*, 2021, **145**, 116459–116459.

90. S. Genheden, A. Thakkar, V. Chadimová, J.-L. Reymond, O. Engkvist and E. Bjerrum, *J. Cheminform.*, 2020, **12**, 70.

91. Y. Mo, Y. Guan, P. Verma, J. Guo, M. E. Fortunato, Z. Lu, C. W. Coley and K. F. Jensen, *Chem. Sci.*, 2021, **12**, 1469–1478.

92. A. Cadeado, C. Machado, G. Oliveira, D. E Silva, R. Muñoz and S. Silva, *J. Braz. Chem. Soc.*

93. M. Mayer and A. J. Baeumner, *Chem. Rev.*, 2019, **119**, 7996–8027.

94. G. R. D. Prabhu, H. A. Witek and P. L. Urban, *React. Chem. Eng.*, 2019, **4**, 1616–1622.

95. R. A. Skilton, R. A. Bourne, Z. Amara, R. Horvath, J. Jin, M. J. Scully, E. Streng, S. L. Y. Tang, P. A. Summers, J. Wang, E. Pérez, N. Asfaw, G. L. P. Aydos, J. Dupont, G. Comak, M. W. George and M. Poliakoff, *Nat. Chem.*, 2015, **7**, 1–5.

96. I. S. Khan, M. O. Ahmad and J. Majava, *J. Clean. Prod.*, 2021, **297**, 126655.

97. T. Zonta, C. A. da Costa, R. da Rosa Righi, M. J. de Lima, E. S. da Trindade and G. P. Li, *Comput. Ind. Eng.*, 2020, **150**, 106889.

98. A. D. Clayton, J. A. Manson, C. J. Taylor, T. W. Chamberlain, B. A. Taylor, G. Clemens and R. A. Bourne, *React. Chem. Eng.*, 2019, **4**, 1545–1554.

99. J. Ke, C. Gao, A. A. Folgueiras-Amador, K. E. Jolley, O. de Frutos, C. Mateos, J. A. Rincón, R. C. D. Brown, M. Poliakoff and M. W. George, *Appl. Spectrosc.*, 2022, **76**, 38–50.

100. D. E. Fitzpatrick, C. Battilocchio and S. V. Ley, *Org. Process Res. Dev.*, 2016, **20**, 386–394.

101. D. Caramelli, J. M. Granda, S. H. M. Mehr, D. Cambié, A. B. Henson and L. Cronin, *ACS Cent. Sci.*, 2021, **7**, 1821–1830.

102. L. Wilbraham, S. H. M. Mehr and L. Cronin, *Acc. Chem. Res.*, 2021, **54**, 253–262.

103. K. Jorner, T. Brinck, P.-O. Norrby and D. Buttar, *Chem. Sci.*, 2021, **12**, 1163–1175.

104. Y. Guan, C. W. Coley, H. Wu, D. Ranasinghe, E. Heid, T. J. Struble, L. Pattanaik, W. H. Green and K. F. Jensen, *Chem. Sci.*, 2021, **12**, 2198–2208.

105. N. Collins, D. Stout, J.-P. Lim, J. P. Malerich, J. D. White, P. B. Madrid, M. Latendresse, D. Krieger, J. Szeto, V.-A. Vu, K. Rucker, M. Deleo, Y. Gorfu, M. Krummenacker, L. A. Hokama, P. Karp and S. Mallya, *Org. Process Res. Dev.*, 2020, **24**, 2064–2077.

106. T. Hardwick and N. Ahmed, *Chem. Sci.*, 2020, **11**, 11973–11988.

107. C. F. Carter, H. Lange, S. V. Ley, I. R. Baxendale, B. Wittkamp, J. G. Goode and N. L. Gaunt, *Org. Process Res. Dev.*, 2010, **14**, 393–404.

108. D. Angelone, A. J. S. Hammer, S. Rohrbach, S. Krambeck, J. M. Granda, J. Wolf, S. Zalesskiy, G. Chisholm and L. Cronin, *Nat. Chem.*, 2021, **13**, 63–69.

109. A. J. S. Hammer, A. I. Leonov, N. L. Bell and L. Cronin, *JACS Au*, 2021, **1**, 1572–1587.

110. S. Heron, M.-G. Maloumbi, M. Dreux, E. Verette and A. Tchapla, *J. Chromatogr. A*, 2007, **1161**, 152–156.

111. W. L. Fitch, A. K. Szardenings and E. M. Fujinari, *Tetrahedron Lett.*, 1997, **38**, 1689–1692.

112. R. Zeng, C. M. Mannaerts and Z. Shang, *Sensors*, 2021, **21**, 6699–6699.

113. S. V. Ley, R. J. Ingham, M. O'Brien and D. L. Browne, *Beilstein J. Org. Chem.*, 2013, **9**, 1051–1072.

114. T. Ahneman Derek, G. Estrada Jesús, S. Lin, D. Dreher Spencer and G. Doyle Abigail, *Science*, 2018, **360**, 186–190.

115. P. Schwaller, A. C. Vaucher, T. Laino and J.-L. Reymond, *Mach. learn.: sci. technol.*, 2021, **2**, 15016–15016.

116. A. L. Haywood, J. Redshaw, M. W. D. Hanson-Heine, A. Taylor, A. Brown, A. M. Mason, T. Gärtner and J. D. Hirst, *J. Chem. Inf. Mod.*, 2021, in press-in press.

117. F. Strieth-Kalthoff, F. Sandfort, M. Kühnemund, F. R. Schäfer, H. Kuchen and F. Glorius, *Angew. Chem. Int. Ed.*, 2022, **61**, e202204647.

118. C. Mauger and G. Mignani, *Synthetic Communications*, 2006, **36**, 1123–1129.

119. S. Siami-Namini, N. Tavakoli and A. Siami Namin, in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2018, pp. 1394–1401.

120. I.-K. Yeo, *Biometrika*, 2000, **87**, 954–959.

121. J. A. Bilbrey, C. O. Marrero, M. Sassi, A. M. Ritzmann, N. J. Henson and M. Schram, *Cite This: ACS Omega*, 2020, **5**, 4594–4594.

122. W. Bort, I. I. Baskin, T. Gimadiev, A. Mukanov, R. Nugmanov, P. Sidorov, G. Marcou, D. Horvath, O. Klimchuk, T. Madzhidov and A. Varnek, *Sci. Rep.*, 2021, **11**, 3178.

123. N. Kunz, nickkunz/smogn https://github.com/nickkunz/smogn 2024.

124. P. Naur and B. Randell, *Software Engineering: Report of a conference sponsored by the NATO Science Committee, Garmisch, Germany, 7-11 Oct. 1968, Brussels, Scientific Affairs Division, NATO*, 1969.

125. B. Fitzgerald, in *Software Technology: 10 Years of Innovation in IEEE Computer*, John Wiley & Sons, Ltd, 2018, pp. 1–16.

126. F. Strieth-Kalthoff, F. Sandfort, M. Kühnemund, F. R. Schäfer, H. Kuchen and F. Glorius, *Angew. Chem. Int. Ed.*, 2022, **61**, e202204647.

127. AWS Marketplace, https://aws.amazon.com/marketplace/pp/prodview-krmdep7ymtvae, (accessed 27 September 2024).

128. eLabFTW, https://www.deltablot.com/elabftw/, (accessed 27 September 2024).

129. S. G. Higgins, A. A. Nogiwa-Valdez and M. M. Stevens, *Nat. Protoc.*, 2022, **17**, 179–189.

130. N. CARPi, A. Minges and M. Piel, *J. Open Source Softw.*, 2017, **2**, 146.

131. ELN-Finder, https://eln-finder.ulb.tu-darmstadt.de/home, (accessed 27 September 2024).

132. S. Kanza, C. Willoughby, N. J. Knight, C. L. Bird, J. G. Frey and S. J. Coles, *Digit. Discov.*, 2023, **2**, 602–617.

133. F. Rudolphi and L. J. Goossen, *J. Chem. Inf. Model.*, 2012, **52**, 293–301.

134. J. Heeley, S. Boobier and J. D. Hirst, *J. Cheminform.*, 2024, **16**, 60.

135. TIOBE Index, https://www.tiobe.com/tiobe-index/, (accessed 10 October 2024).

136. PEP 8 – Style Guide for Python Code | peps.python.org, https://peps.python.org/pep-0008/, (accessed 11 October 2024).

137. D. F. Nippa, A. T. Müller, K. Atz, D. B. Konrad, U. Grether, R. E. Martin and G. Schneider, *ChemRxiv*, 2024, DOI: 10.26434/chemrxiv-2023-nfq7h-v2.

138. TheELNConsortium/TheELNFileFormat The ELN Consortium 2024.

139. Research Object Crate (RO-Crate), https://www.researchobject.org/ro-crate/specification/1.1/, (accessed 18 October 2024).

140. Schema.org - Schema.org, https://schema.org/, (accessed 18 October 2024).

141. AI4Green, https://ai4green.app/home, (accessed 4 November 2024).

142. S. Boobier, J. C. Davies, I. N. Derbenev, C. M. Handley and J. D. Hirst, *J. Chem. Inf. Model.*, 2023, **63**, 2895–2901.

143. E. J. Corey and W. T. Wipke, *Science*, 1969, **166**, 178–192.

144. S. Szymkuć, E. P. Gajewska, T. Klucznik, K. Molga, P. Dittwald, M. Startek, M. Bajczyk and B. A. Grzybowski, *Angew. Chem. Int. Ed. Engl.*, 2016, **55**, 5904–5937.

145. Y. Jiang, Y. Yu, M. Kong, Y. Mei, L. Yuan, Z. Huang, K. Kuang, Z. Wang, H. Yao, J. Zou, C. W. Coley and Y. Wei, *Engineering*, 2023, **25**, 32–50.

146. T. Klucznik, B. Mikulak-Klucznik, M. P. McCormack, H. Lima, S. Szymkuć, M. Bhowmick, K. Molga, Y. Zhou, L. Rickershauser, E. P. Gajewska, A. Toutchkine, P. Dittwald, M. P. Startek, G. J. Kirkovits, R. Roszak, A. Adamski, B. Sieredzińska, M. Mrksich, S. L. J. Trice and B. A. Grzybowski, *Chem*, 2018, **4**, 522–532.

147. B. Mikulak-Klucznik, P. Gołębiowska, A. A. Bayly, O. Popik, T. Klucznik, S. Szymkuć, E. P. Gajewska, P. Dittwald, O. Staszewska-Krajewska, W. Beker, T. Badowski, K. A. Scheidt, K. Molga, J. Mlynarski, M. Mrksich and B. A. Grzybowski, *Nature*, 2020, **588**, 83–88.

148. Y. Lin, R. Zhang, D. Wang and T. Cernak, *Science*, 2023, **379**, 453–457.

149. K. Zeng, B. Yang, X. Zhao, Y. Zhang, F. Nie, X. Yang, Y. Jin and Y. Xu, *J. Cheminform.*, 2024, **16**, 80.

150. B. Liu, B. Ramsundar, P. Kawthekar, J. Shi, J. Gomes, Q. Luu Nguyen, S. Ho, J. Sloane, P. Wender and V. Pande, *ACS Cent. Sci.*, 2017, **3**, 1103–1113.

151. L. Yao, W. Guo, Z. Wang, S. Xiang, W. Liu and G. Ke, *JACS Au*, 2024, **4**, 992–1003.

152. W. Zhong, Z. Yang and C. Y.-C. Chen, *Nat. Commun.*, 2023, **14**, 3009.

153. M. Krenn, Q. Ai, S. Barthel, N. Carson, A. Frei, N. C. Frey, P. Friederich, T. Gaudin, A. A. Gayle, K. M. Jablonka, R. F. Lameiro, D. Lemm, A. Lo, S. M. Moosavi, J. M. Nápoles-Duarte, A. Nigam, R. Pollice, K. Rajan, U. Schatzschneider, P. Schwaller, M. Skreta, B. Smit, F. Strieth-Kalthoff, C. Sun, G. Tom, G. Falk von Rudorff, A. Wang, A. D. White, A. Young, R. Yu and A. Aspuru-Guzik, *Patterns*, 2022, **3**, 100588.

154. Z. Tu, T. Stuyver and C. W. Coley, *Chem. Sci.*, 2023, **14**, 226–244.

155. M. E. Fortunato, C. W. Coley, B. C. Barnes and K. F. Jensen, *J. Chem. Inf. Model.*, 2020, **60**, 3398–3407.

156. Y. Shee, H. Li, P. Zhang, A. M. Nikolic, W. Lu, H. R. Kelly, V. Manee, S. Sreekumar, F. G. Buono, J. J. Song, T. R. Newhouse and V. S. Batista, *Nat. Commun.*, 2024, **15**, 7818.

157. A. Thakkar, T. Kogej, J.-L. Reymond, O. Engkvist and E. Jannik Bjerrum, *Chem. Sci.*, 2020, **11**, 154–168.

158. M. E. Maier, *Org. Biomol. Chem.*, 2015, **13**, 5302–5343.

159. F. Lovering, J. Bikker and C. Humblet, *J. Med. Chem.*, 2009, **52**, 6752–6756.

160. D. G. Brown and J. Boström, *J. Med. Chem.*, 2016, **59**, 4443–4458.

161. A. M. Westerlund, B. Barge, L. Mervin and S. Genheden, *Mol. Inform.*, 2023, **42**, e202300128.

162. V. Delannée and M. C. Nicklaus, *J. Cheminform.*, 2020, **12**, 72.

163. N. Schneider, D. M. Lowe, R. A. Sayle and G. A. Landrum, *J. Chem. Inf. Model.*, 2015, **55**, 39–53.

164. D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel and D. Hassabis, *Nature*, 2016, **529**, 484–489.

165. T. J. Struble, J. C. Alvarez, S. P. Brown, M. Chytil, J. Cisar, R. L. DesJarlais, O. Engkvist, S. A. Frank, D. R. Greve, D. J. Griffin, X. Hou, J. W. Johannes, C. Kreatsoulas, B. Lahue, M. Mathea, G. Mogk, C. A. Nicolaou, A. D. Palmer, D. J. Price, R. I. Robinson, S. Salentin, L. Xing, T. Jaakkola, William. H. Green, R. Barzilay, C. W. Coley and K. F. Jensen, *J. Med. Chem.*, 2020, **63**, 8667–8682.

166. L. Saigiridharan, A. K. Hassen, H. Lai, P. Torren-Peraire, O. Engkvist and S. Genheden, *J. Cheminform.*, 2024, **16**, 57.

167. S. Genheden and E. Bjerrum, *Digit. Discov.*, 2022, **1**, 527–539.

168. K. Maziarz, A. Tripp, G. Liu, M. Stanley, S. Xie, P. Gaiński, P. Seidl and M. H. S. Segler, *Faraday Discuss.*, 2025, Advance Article.

169. C. W. Coley, D. A. Thomas, J. A. M. Lummiss, J. N. Jaworski, C. P. Breen, V. Schultz, T. Hart, J. S. Fishman, L. Rogers, H. Gao, R. W. Hicklin, P. P. Plehiers, J. Byington, J. S. Piotti, W. H. Green, A. J. Hart, T. F. Jamison and K. F. Jensen, *Science*, 2019, **365**, eaax1566.

170. IBM RXN for Chemistry, https://rxn.app.accelerate.science/rxn/subscription/plans, (accessed 30 November 2024).

171. A. Cardinale, A. Castrogiovanni, T. Gaudin, J. Geluykens, T. Laino, M. Manica, D. Probst, P. Schwaller, A. Sobczyk, A. Toniato, A. C. Vaucher, H. Wolf and F. Zipoli, *CHIMIA*, 2023, **77**, 484–488.

172. S. Sinha, M. Obkircher, M. Yaragani, R. Deokar, S. Kumar and E. Gardener, *Computer-Assisted Synthetic Route Optimization using SYNTHIA™ Retrosynthesis Software*, Merck Life Science, 2024.

173. C. J. Taylor, K. C. Felton, D. Wigh, M. I. Jeraal, R. Grainger, G. Chessari, C. N. Johnson and A. A. Lapkin, *ACS Cent. Sci.*, 2023, **9**, 957–968.

174. D. J. C. Constable, C. Jimenez-Gonzalez and R. K. Henderson, *Org. Process Res. Dev.*, 2007, **11**, 133–137.

175. B. W. Cue and J. Zhang, *Green Chem. Lett. Rev.*, 2009, **2**, 193–211.

176. United States, US3385886A, 1968.

177. United States, US4981995A, 1991.

178. T. Laird, *Org. Process Res. Dev.*, 2012, **16**, 1–2.

Appendix

Chapter 3: Machine Learning for Yield Prediction for Chemical Reactions Using in situ Sensors

Code and data for replicating the work reported here can be found on GitHub. https://github.com/JoeDavies-6/ML-for-product-prediction

Sensor data

1) **DX 1.5 – reaction probe (unused sensors omitted)**
    - Temperature of glass (reaction) at immersed depth, in Celsius
    - Stir rate – stir rate of the stir plate, in revolutions per minute.
    - uvaE – UV A response at exposed level, in $\mu W/cm^2$
    - uvaI – UV A response at immersed level in $\mu W/cm^2$
    - uvbE – UV B response at exposed level in $\mu W/cm^2$
    - uvbI – UV B response at immersed level in $\mu W/cm^2$
2) **Camera – on end of reaction probe**
    - Blue – blue component of average colour capture by camera, 0-255 pixels
    - Green – green component of average colour capture by camera, 0-255 pixels
    - Red– red component of average colour capture by camera, 0-255 pixels
3) **Environmental Sensor (ES) – in reaction hood but not in the reaction mixture.**
    - Luxometer, environmental light in $\mu W/cm^2$.
    - Humidity – environmental humidity in %
    - Temperature – environmental temperature in Celsius
    - Pressure – environmental pressure in hPa

4) <u>Vernier Thermometer – thermometer inside reaction mixture</u>

1. Temperature – temperature of Vernier contact thermometer in Celsiu

Table A1: All experiment runs, results, and permutations.

Experiment title	Solvent	Experiment duration (hours)	Batches of hydrazine addition	Camera present	Vernier thermometer	Environmental Sensor	HPLC UV Conversion (%)	Used in modelling?
Crocus-001	TAA	2	13	No	No	No	92.8	Not used
Crocus-009	TAA	2	13	No	No	Yes	89.8	Not used
Crocus-010	TAA	2	13	No	No	Yes	88.0	Not used
Crocus-011	TAA	2	13	No	No	Yes	76.6	Not used
Crocus-014	IPA	5	13	Yes	No	Yes	37.8	Used
Crocus-015	IPA	5	13	Yes	No	Yes	39.7	Used
Crocus-016	IPA	5	13	Yes	No	Yes	20.2	Used
Crocus-017	IPA	5	13	Yes	No	Yes	8.8	Used
Crocus-018	IPA	5	13	Yes	Yes	Yes	47.2	Used
Crocus-019	IPA	5	13	Yes	Yes	Yes	46.1	Used
Crocus-020	IPA	5	13	Yes	Yes	Yes	45.1	Used
Crocus-022	IPA	5	3	Yes	Yes	Yes	45.4	Used
Crocus-023	IPA	8	3	Yes	Yes	Yes	38.9	Used
Crocus-024	IPA	8	3	No	Yes	Yes	39.8	Not used
Crocus-025	IPA	8	3	Yes	Yes	Yes	49.8	Used

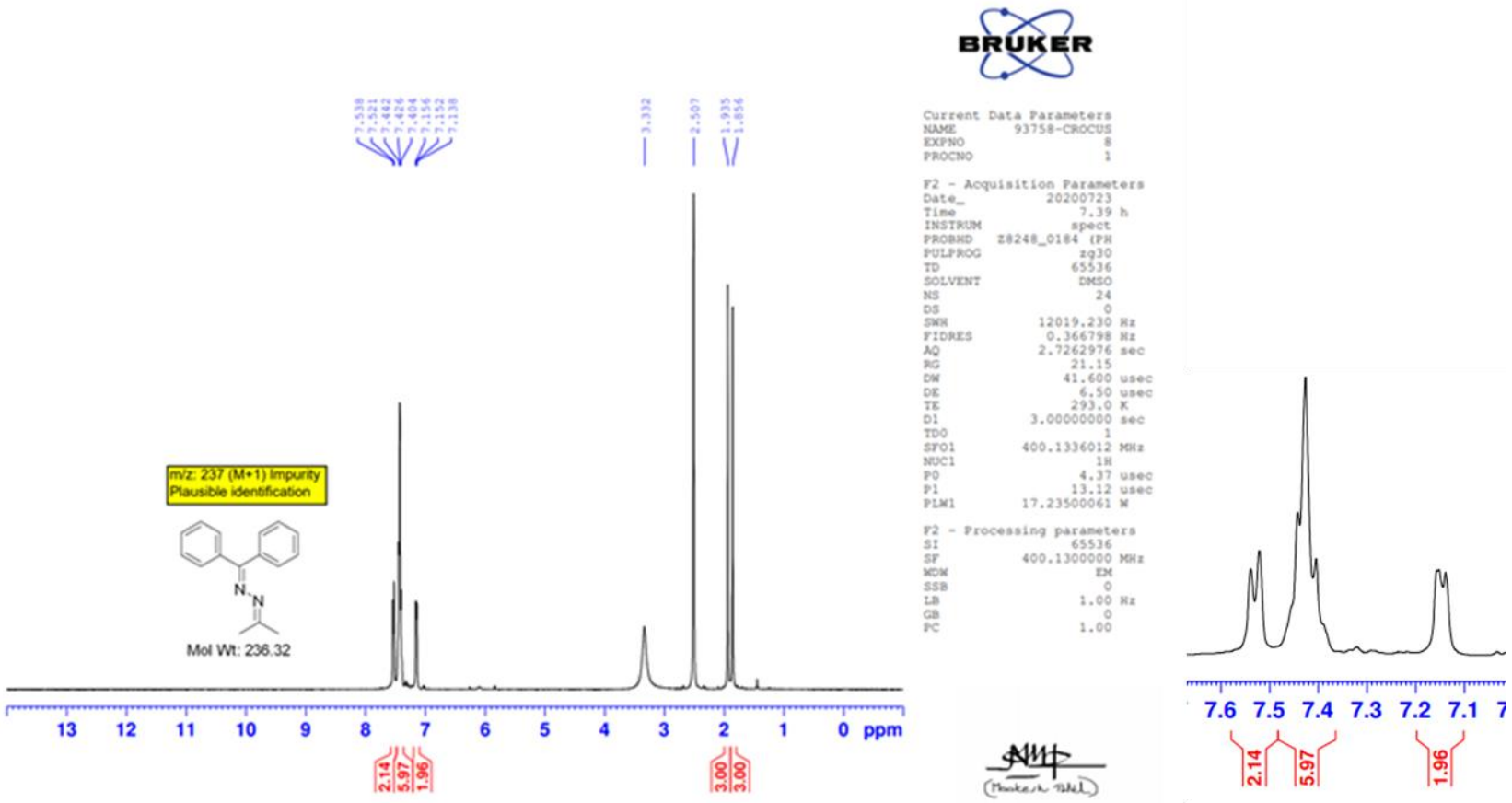


Figure A1 1H NMR of impurity from Buchwald reaction.

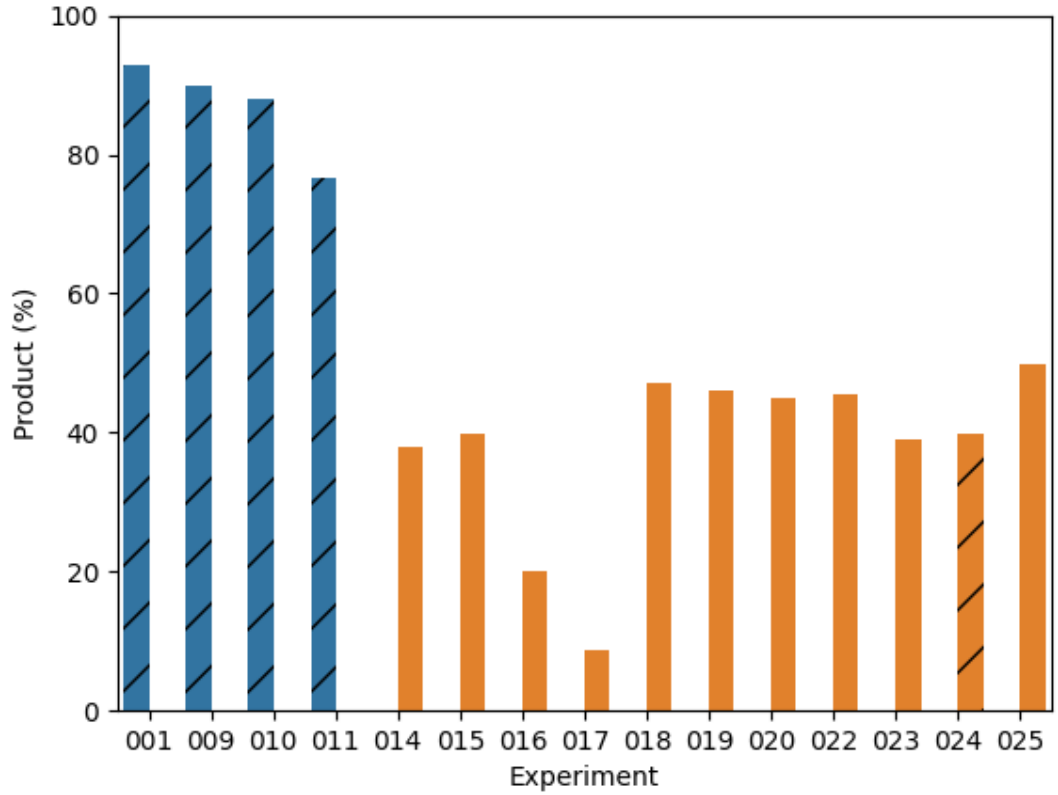


Figure A2: Percentage product at reaction completion measured by UV trace from HPLC data. The blue bars represent the reactions performed in tert-amyl alcohol and the orange bars represent the reactions performed in iso-propyl alcohol. The bars with diagonal lines represent runs which were excluded from modelling.

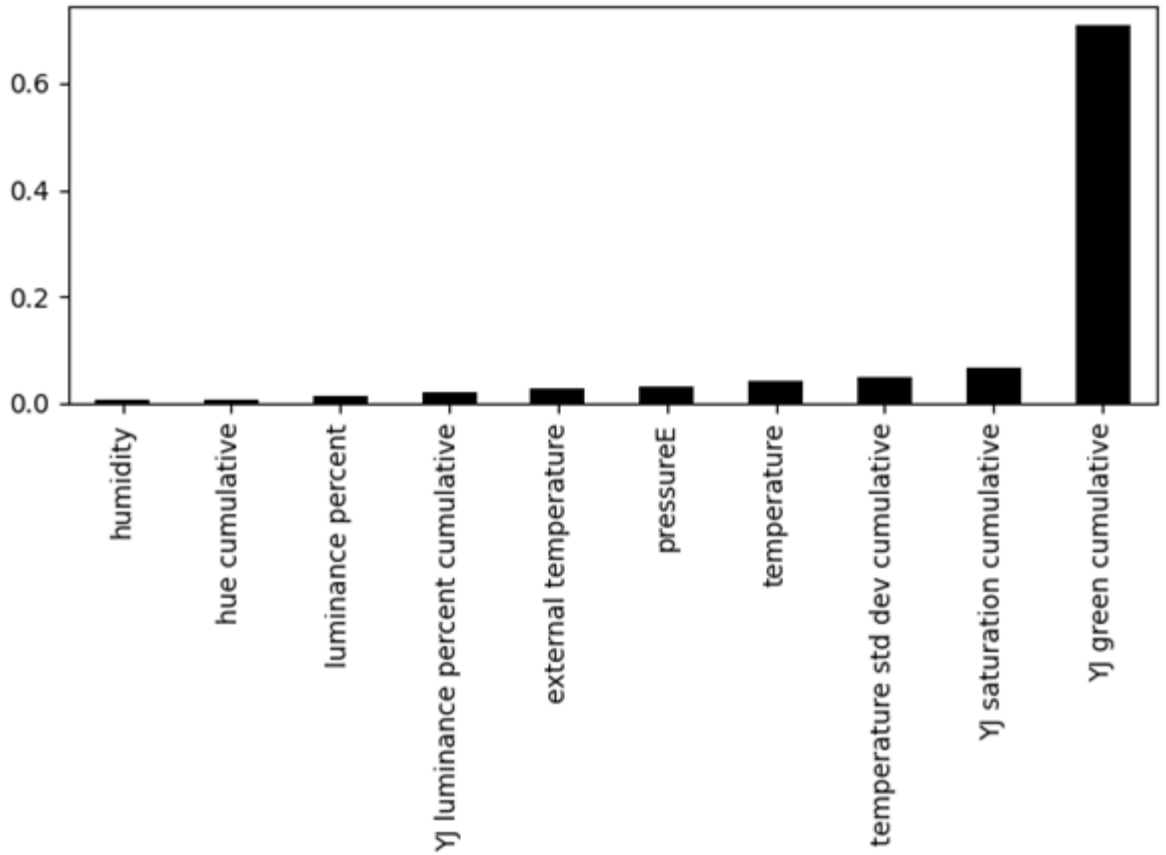


Figure A3: The ten highest relative feature importance values from a gradient boosted regression model. Data is from run 025 and is representative of all runs and the random forest model trained on all features. Yeo-Johnson has been shorted to YJ.

Table A2: Standard Error values from the 10-fold cross-validation of the non-neural network methodologies used.

	All features	No colour features	YJ green cumulative	Green cumulative
Linear regression	1.27	2.36	0.52	0.51
Random forest	0.65	1.72	0.60	0.60
Polynomial regression	x	x	x	0.61
Gradient boosting regressor	0.39	1.39	0.63	0.63

Table A3: Individual results and t-test comparison with baseline from the 10-fold cross-validation of the non-neural network methodologies.

Feature set - model combination	Baseline t-tests p value	Significant difference vs baseline
Green cumulative-polynomial regression	0.014	Yes
Green cumulative-random forest	0.048	Yes
Green cumulative-gradient boosting regressor	0.032	Yes
Green cumulative-linear regression	0.030	Yes
All features-gradient boosting regressor	0.015	Yes
Yeo-Johnson Green cumulative-linear regression	0.015	Yes
Yeo-Johnson Green cumulative-gradient boosting regressor	0.032	Yes
Yeo-Johnson Green cumulative-random forest	0.048	Yes
All features-linear regression	0.454	No
All features-random forest	0.285	No
No colour features-linear regression	0.616	No
No colour features-random forest	0.595	No
No colour features-gradient boosting regressor	0.987	No
baseline	1.000	No

Table A4: Analysis of variance of non-neural network methodologies

	ANOVA p value	Significant differences within group
All Methodologies (excluding baseline)	0.0056	Yes
Eight methodologies with a significant difference to the baseline	0.97	No

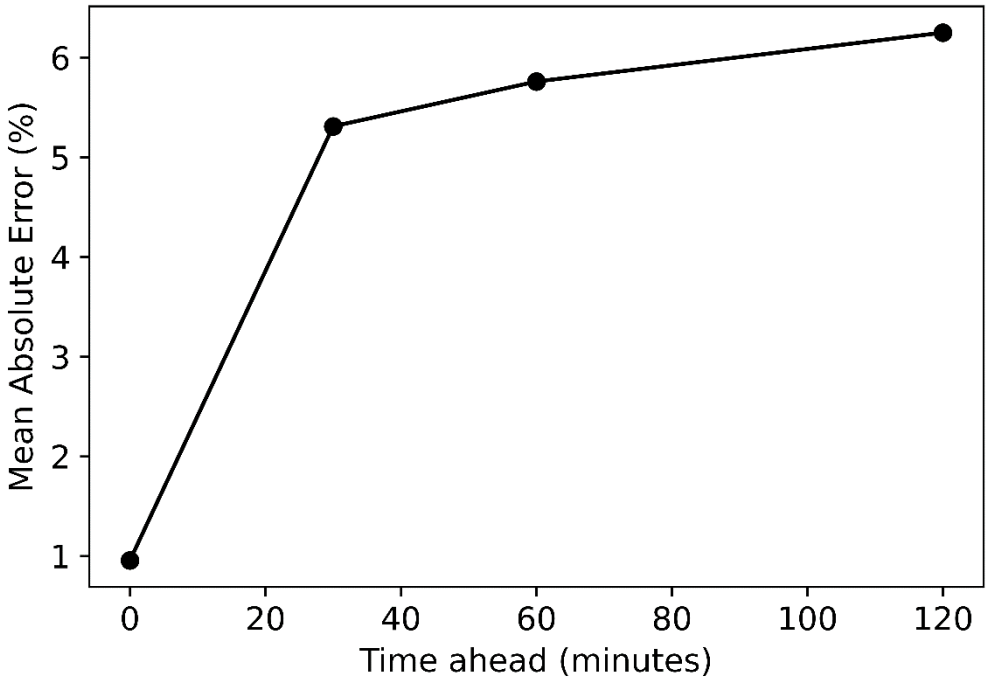


Figure A4: The mean absolute error for product formation predictions from an LSTM model predicting 0, 30, 60 or 120 minutes ahead using a 45-fold cross-validation with an 8:2 train-test split.

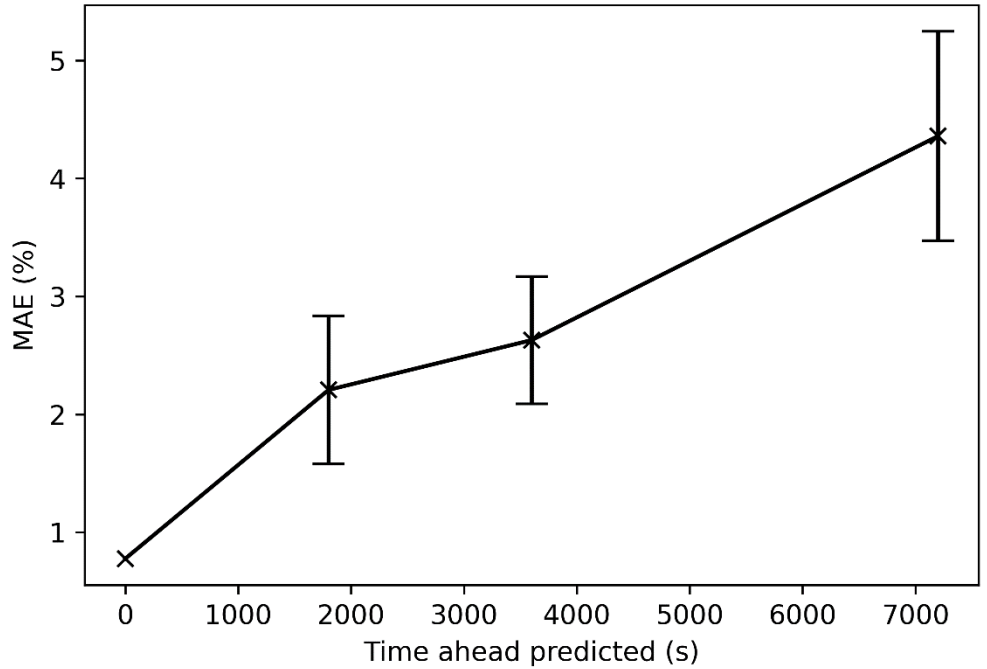


Figure A5: Error values and standard error bars from predicting results for run 25. The standard error bars are calculated from ten repeats.

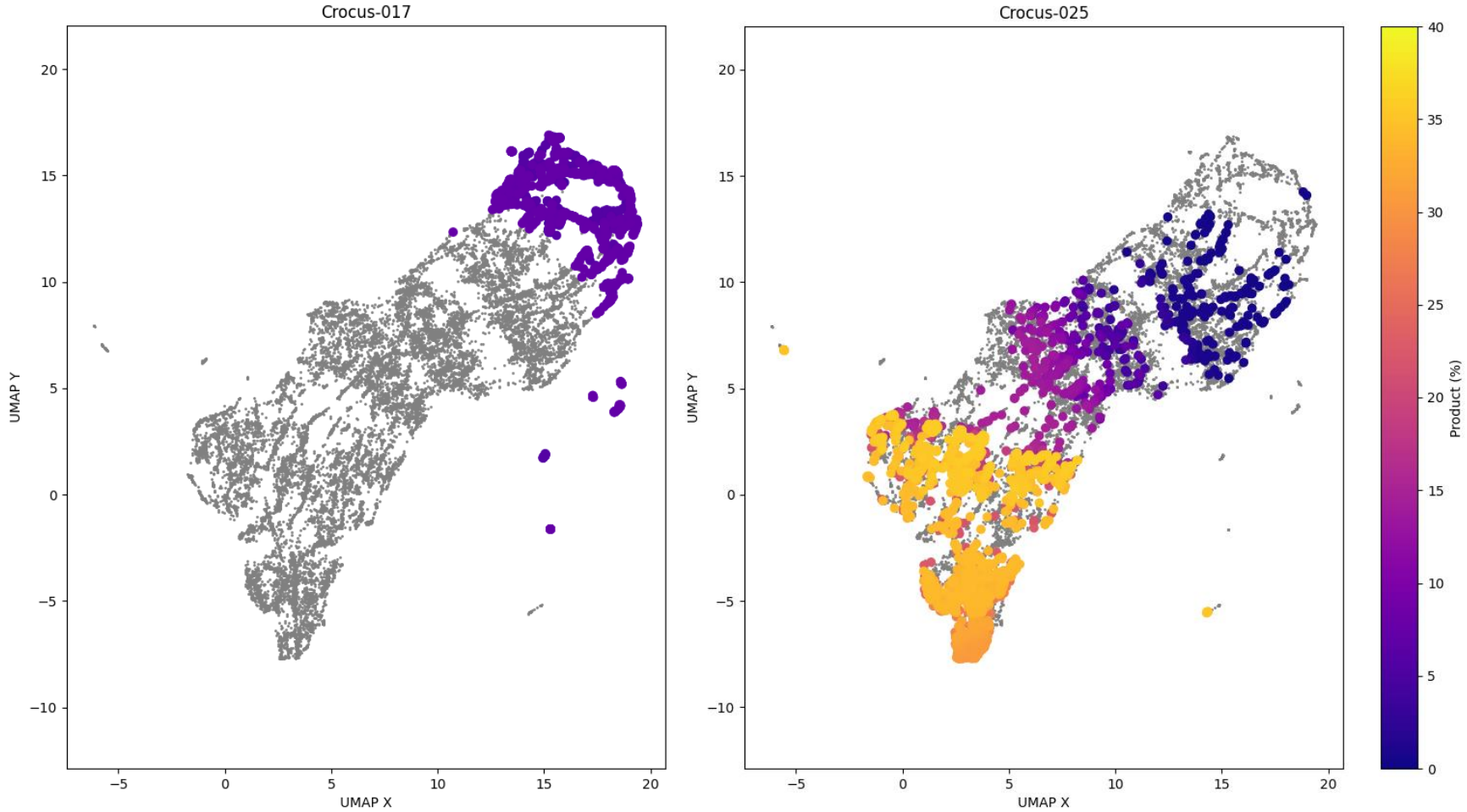


Figure A6:Left panel: Run 16. Right panel: Run 25. UMAP projection of all run data with the datapoints for a specific run highlighted according to product formation.

Chapter 4: Development of the AI4Green Electronic Laboratory Notebook for Green and Sustainable Chemistry

Code for the work reported here can be found on GitHub: https://github.com/AI4Green/AI4Green

The hosted application for the work reported here can be found at: https://ai4green.app/

Screenshots of the interface:

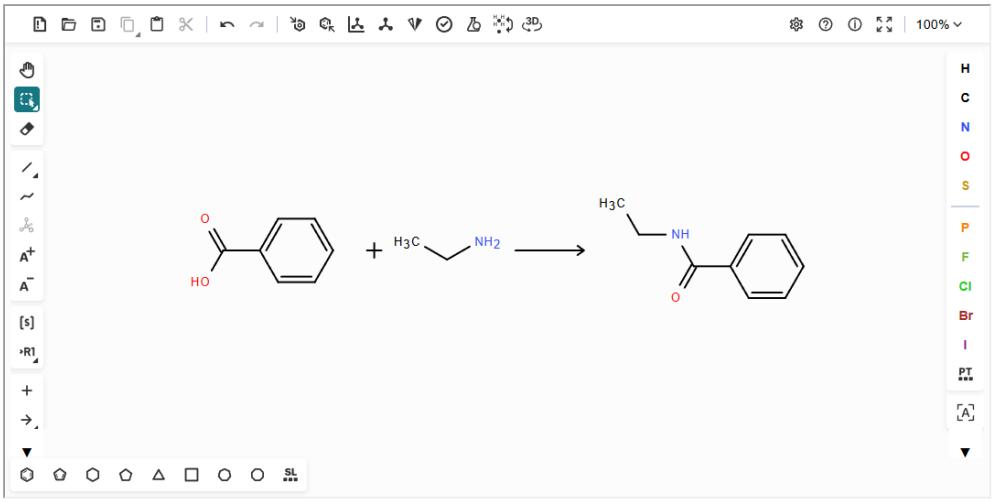


Figure A7: The integrated Ketcher chemical sketcher with an amide reaction drawn.

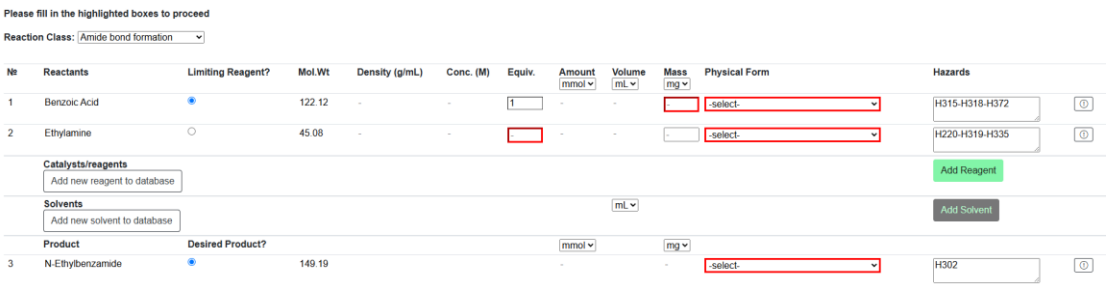


Figure A8: The empty reaction table immediately after submitting the sketcher contents, with required fields highlighted in red and the option to add reagents and solvents.

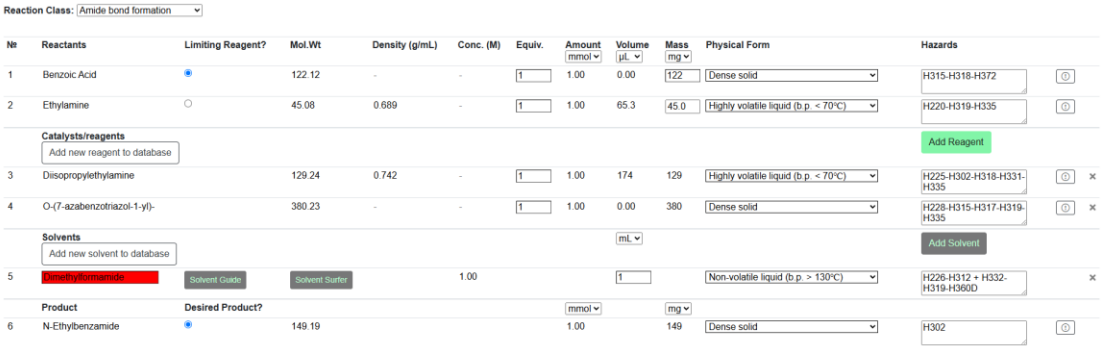


Figure A9: The completed reaction table for an experiment with the reaction class being autodetected from the reaction sketcher contents.

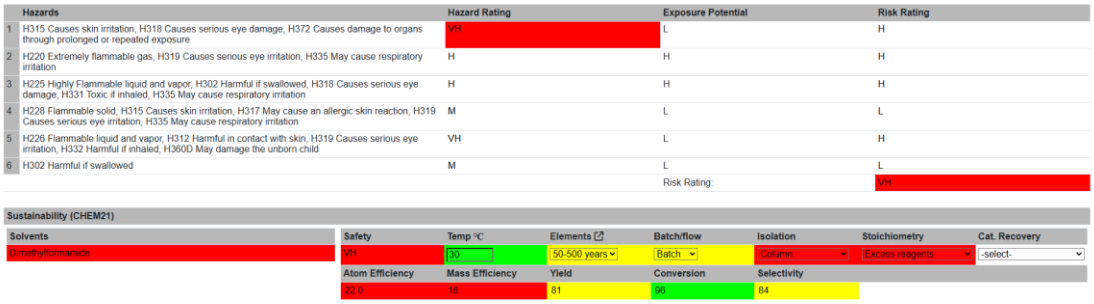


Figure A10: The hazard matrix and sustainability assessment for the reaction in the summary table

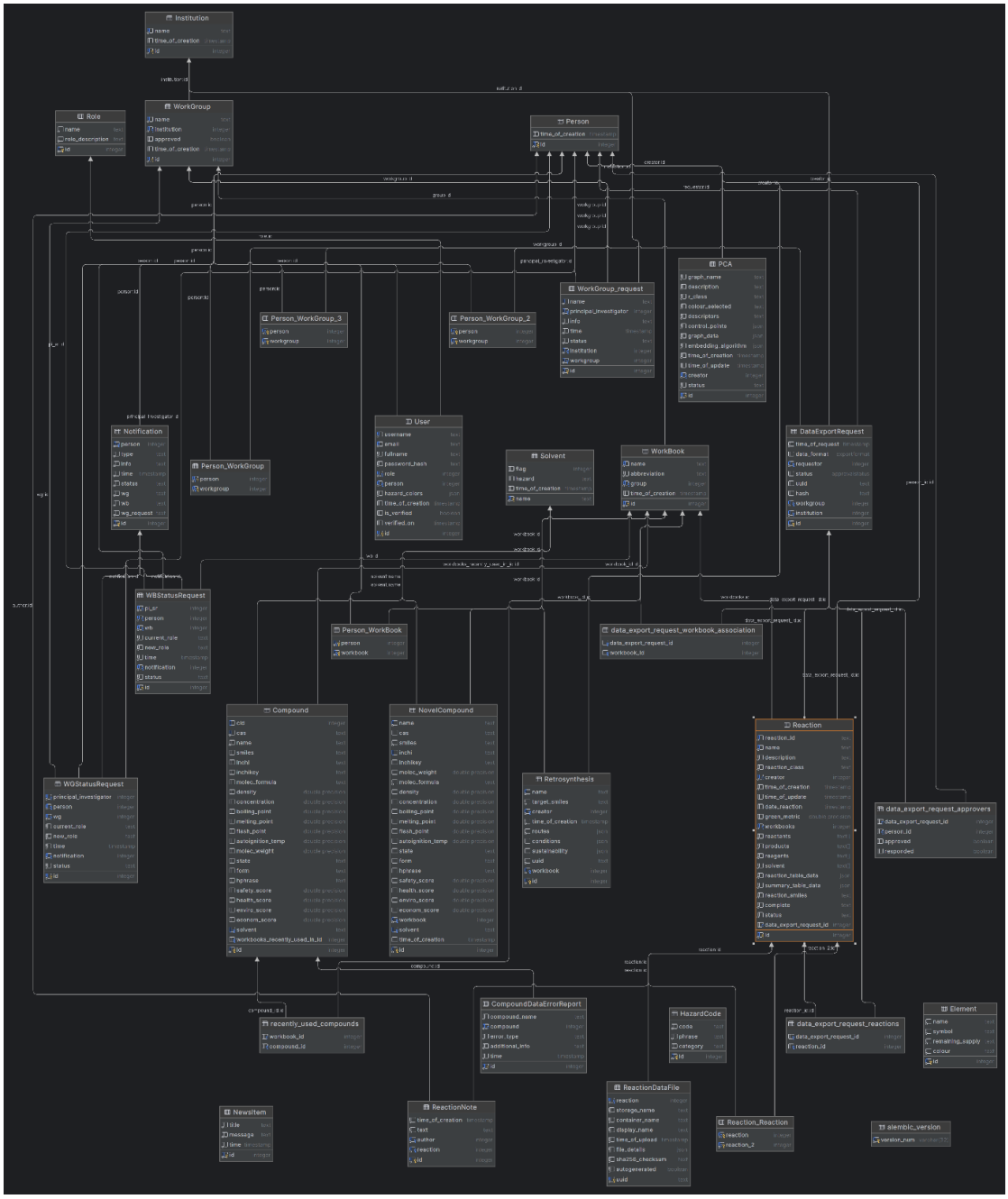


Figure A11: AI4Green database schema

Chapter 5: Integration of a Retrosynthesis Tool with Sustainability Metrics into the AI4Green Electronic Laboratory Notebook

The code for the work reported here can be found on GitHub:

https://github.com/AI4Green/retrosynthesis-api

https://github.com/AI4Green/conditions-api

This work built upon the code and models provided by ASKCOS and AiZynthFinder.

The code for the ASKCOS condition prediction tool can found at:

https://gitlab.com/mlpds_mit/askcosv2/context_recommender

The code for the AiZynthFinder retrosynthetic planner can be at:

https://github.com/MolecularAI/aizynthfinder