Systems & Control Transactions

**Research Article - Peer Reviewed Conference Proceeding**
ESCAPE 35 - European Symposium on Computer Aided Process Engineering
Ghent, Belgium. 6-9 July 2025
Jan F.M. Van Impe, Grégoire Léonard, Satyajeet S. Bhonsale,
Monika E. Polańska, Filip Logist (Eds.)

PSE PRESS

# The Smart HPLC Robot: Fully Autonomous Method Development Guided by A Mechanistic Model Framework

**Dian Ning Chia, Fanyi Duanmu, Luca Mazzei, Eva Sorensen, Maximilian O. Besenhard***

Department of Chemical Engineering, University College London, Torrington Place, London WC1E 7JE, United Kingdom
* Corresponding Author: m.besenhard@ucl.ac.uk

## ABSTRACT

Developing ultra- or high-performance liquid chromatography (HPLC) methods for analysis or purification requires significant amounts of material and manpower, and typically involves time-consuming iterative lab-based workflows. This work demonstrates in two case studies that an autonomous HPLC platform coupled with a mechanistic model that self-corrects itself by performing parameter estimation can efficiently develop an optimized HPLC method with minimal experiments (i.e., reduced experimental costs and burden) and manual intervention (i.e., reduced manpower). At the same time, this HPLC platform, referred to as Smart HPLC Robot, can deliver a calibrated mechanistic model that provides valuable insights into method robustness.

**Keywords**: Industry 4.0, Modelling and Simulations, Optimization, Genetic Algorithm, Batch Process, Self-driving, Autonomous, Digital Twin, Mechanistic Model, Chromatography

## INTRODUCTION

One of the most important analysis and purification methods in the fine chemical and pharmaceutical industry is liquid chromatography [1]. The degree of separation necessary for success is dictated by the affinity of the solutes (i.e., the dissolved molecules) towards the solid phase (i.e., the adsorbent). The greater the affinity of the solute for the solid phase, the slower its movement through the chromatography column, resulting in a later elution. Considering a column with a fixed solid phase, one of the most significant factors affecting the affinity of the solute is solvent composition, where the solvent in HPLC processes is typically a mixture of organic and inorganic solvents. The simplest way to operate an HPLC process is through isocratic elution, where the solvent composition remains constant throughout the operation. However, isocratic elution often has limitations, such as long process time (the time taken for the last component to elute) or the inability to satisfactorily resolve a complex mixture. An alternative mode of operation is gradient elution, where the solvent composition varies over time. This approach introduces additional degrees of freedom, such as the initial and final solvent composition, the duration of the composition change, and the number of gradient steps, adding complexity to the process. Even so, gradient elution is generally preferred for complex mixtures due to its enhanced ability to resolve components more efficiently and quickly [2].

Developing gradient based HPLC methods usually requires significant material and manpower, and typically involves time-consuming and iterative laboratory-based workflows. Recent technological advancements now enable fully computer-controlled workflows; these allow not only automated but also streamlined autonomous (that is, self-driven) HPLC method development, with minimal to no human intervention (employing robotic solutions for iterative workflows [3]). In this article, we summarize two recent applications of autonomous processes; readers are referred to Tom et al. [3] for a comprehensive review of the concepts of autonomous processes in laboratories (coined as self-driving laboratories in the review article). Boelrijk et al. [2] and Dixon et al. [4] demonstrated the potential of autonomy for developing HPLC methods. Both authors achieved autonomy of gradient method development via the following steps:

1. **Initial Small Experimental Campaign,** e.g. as in Boelrijk et al. via random input parameter selection [2] or Dixon et al. via Latin Hypercube Sampling (LHS)[4].
2. **Fit the data-based surrogate model** using experiments performed in step 1.

3. **Optimize the HPLC process** using the surrogate model, where the optimizer **suggests the next experiment(s)** to perform. Both authors used Bayesian optimization aimed to optimize gradient methods regarding time, and number of resolved peaks.
4. **Run model-guided additional real experiments** and **analyze key indicators in-silico** (=computationally), e.g., peak height, peak width, retention time. The key indicators are analyzed by a surrogate model and are automatically sent to the optimizer (used in the next step); no manual interpretation is required.
5. **Repeat steps 2 and 3** until reaching stopping criteria.

Both groups demonstrated that satisfactory optimal results could be achieved autonomously with the help of in-silico surrogate models. However, surrogate models rely on data to be informative and effective, making them less efficient in the early stages. Due to their data-driven nature, they may not accurately reflect known physical phenomena, limiting their ability to provide reliable guidance, especially when only a few experiments are available for model training and development.

To address the above-mentioned issue, **this work proposes an autonomous process guided by a mechanistic model** for knowledge-driven (instead of purely data-driven) decision-making. One of the challenges when using a mechanistic model is to find a set of good parameters (e.g., isotherm parameters) that can represent the real experimental results (e.g., chromatogram) sufficiently well. Therefore, a parameter estimation step is added to the previously mentioned automation step (see discussion for Figure 2 in the methodology section). This framework, here called "Smart HPLC Robot", is an intelligent platform enabling the development of an optimized HPLC method with minimal experiments while simultaneously delivering a calibrated mechanistic model that provides valuable insights into method robustness. Importantly, since mechanistic models are physics-based and not tied to a specific objective function, they can be applied to optimization tasks with different objectives and provide reliable predictions beyond the range of experimental settings used for training (i.e., extrapolation). Once model parameters are properly determined and experimental conditions remain consistent (e.g., using the same mixture), mechanistic models offer better adaptability, require fewer calibration experiments, and do not need retraining, unlike the data based surrogate models used in previous works.

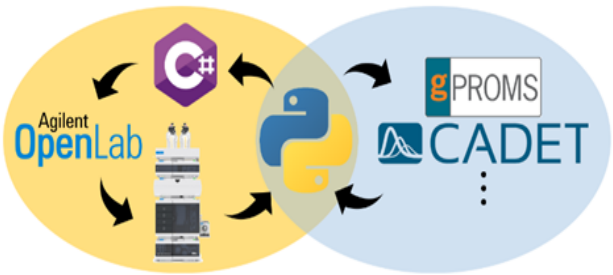



**Figure 1**. Communication protocol of Smart HPLC Robot.

## METHODOLOGY

The Smart HPLC Robot features mechanistic simulations and an autonomous process; thus, it requires efficient communication between laboratory equipment and chromatography simulators. The developed autonomous workflow, including the communication between model and HPLC system, is illustrated in Figure 1 and discussed in this section. In addition, the working principles (shown in Figure 2) will also be discussed.

### Communication Protocol

The user interface is developed in Python, which also bridges the communication between equipment and simulators. As shown in Figure 1, the current project is developed with Agilent equipment and simulators such as gPROMS and CADET (https://cadet-web.de/), but it is not limited to those and can be integrated with other systems, provided that users adhere to a standard communication framework. To integrate equipment from different manufacturers, users can leverage, for example, Application Programming Interfaces (APIs) to efficiently establish communication and exchange key values.

In this project, the communication protocol for Agilent OpenLab is written in C# (left part in Figure 1). Therefore the equipment control sequences (e.g., modifying experimental input sequence and executing sequence) are all implemented in C# for better compatibility and operation efficiency. A C# web API facilitates seamless data exchange by providing a standardized communication interface while isolating equipment control sequences. The right part of Figure 1 illustrates the communication between the Python framework and simulators. Currently, gPROMS and CADET are supported, and the communications with Python are via gO:Python and CADET-Python, respectively.

### Working Principle

The working principle of the Smart HPLC Robot can be divided into two main steps as shown in Figure 2: (1) the preparation step (a one-time execution) and (2) the mechanistic loop (i.e., mechanistic model-guided autonomous method development). The former requires configuring the Smart HPLC Robot and performing initial experiments, and the latter involves parameter estimation for model calibration and optimization for the generation of new experiments. This section will detail each step together with associated challenges and solutions.

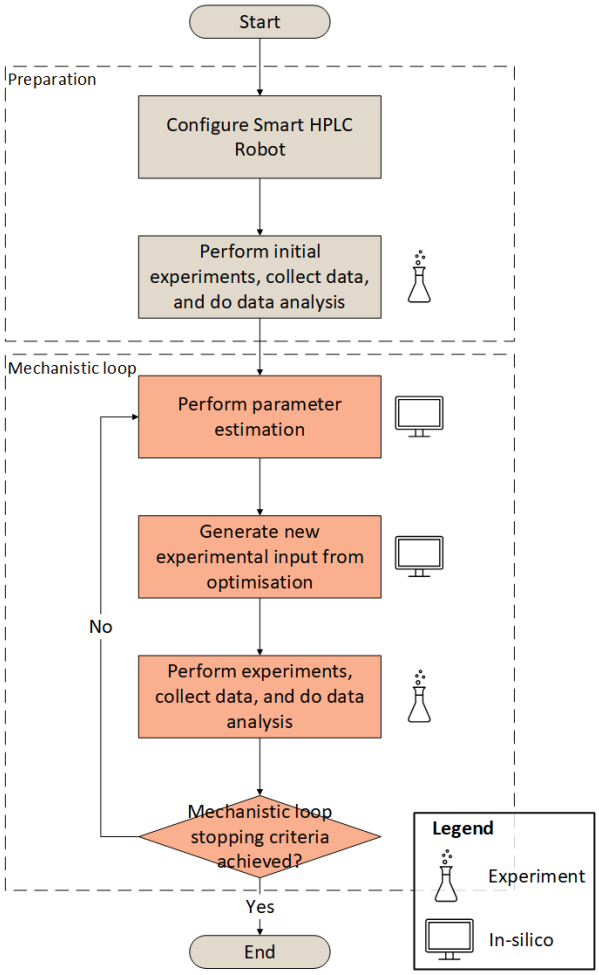



**Figure 2.** Smart HPLC Robot working principle. All steps are performed autonomously after the initial manual configuration of the robot (e.g., input column dimension, model selection, optimization settings).

## Preparation

In contrast to using machine learning [2,4], the mechanistic model framework used in this work requires prior knowledge of the number of components in the system for model development. This step is more straightforward with a mass spectrometer than an ultraviolet (UV) detector, as a UV detector cannot distinguish overlapping components, leading to inaccurate method development. When using a UV detector (as done here), it is recommended to perform preliminary experiments at low flow rates and a low initial organic solvent fraction and low gradients. This approach can, theoretically, maximize the separation of components with the current system, providing an initial understanding of the number of components that can be resolved from the (potentially) unknown mixture.

Besides the number of components, other information, e.g., column dimensions, selection of isotherms and mass transport model, and parameter estimation/optimization settings, including boundaries of parameter estimation/optimization variables and objective functions, is also required for configuring the mechanistic model of the Smart HPLC Robot. Most information can be obtained from equipment manufacturers or is based on the aim of the task. The boundaries of variables are crucial to the success of the HPLC method development, but they are often tricky to determine. For design variables, such as flow rates, percentage of organic solvent, and column temperature, the boundaries are often defined as the equipment physical limitations. Isotherm parameters, on the other hand, lack clear boundaries. The Smart HPLC Robot conducts a basic boundary analysis using Sobol sampling across an initial input search space (HPLC settings), typically with broad boundaries. This process helps identify regions with a higher likelihood of yielding optimal results, as indicated by the objective function's quantification of the HPLC method quality. Hence, shrinking the search space for the subsequent parameter estimation step, leading to an improved efficiency.

Once all necessary information is provided, users can execute the program and the robot will do all other steps without further human interference. The first automated step is to generate and perform a user-defined number of initial experiments generated from sampling methods (e.g., LHS or factorial design). The number of initial experiments is a trade-off between accuracy of prediction and experimental burden. For the low-parameter mechanistic models used thus far, four experiments provided a good foundation

## Model Calibration and Validation Loop

The mechanistic model's initial training (parameter estimation) will begin once the Smart HPLC Robot completes the initial experiments and sends the results to Python. The first step in the loop is to estimate the parameters for the selected isotherm(s) (e.g., linear isotherm or Langmuir isotherm) based on the user-defined objective functions (e.g., maximum likelihood method or least square method). In this step, all experiments, including initial experiments and those generated in the mechanistic loop, will be taken into account.

Once calibrated with the best available parameter estimates, the mechanistic model computationally optimizes the HPLC method settings (i.e., the inputs such as flow rate, initial and final mobile phase composition, and gradient time) based on the selected method objectives (such as achieving sufficient resolution for all or selected components in the shortest possible time). Therefore, selecting an appropriate objective function is crucial to yield the desired HPLC method.

Once computational optimization is complete, the HPLC system will automatically execute the updated

method (using the optimum inputs) and return key data (including UV absorbance, retention time, and peak width, i.e., the method results) to Python. The latter aims to validate whether the computed optimum has been achieved and to determine if a sufficiently robust HPLC method now meets the stopping criteria. If not, it serves as a new experiment to refine the mechanistic model parameters and rerun the computational optimization. The stopping criteria may include factors like reaching the maximum iterations or no further method improvement. Once met, the Smart HPLC Robot reports the validated model and optimal HPLC method settings. In most cases, the mechanistic loop requires multiple iterations. During the parameter estimation step, both stochastic and deterministic optimizers carry the risk of yielding worse isotherm parameters than the previous iteration (stochastic optimizers due to inherent randomness and deterministic ones due to sensitivity to initial values).Therefore, the newly estimated model parameters will be compared with the previous best values, and only those yielding an improved estimation of parameters (reduced error between experiment and model calculation) will be used to update the model.

## RESULTS AND DISCUSSIONS

Two case studies were performed in this work: (1) an in-silico system case study to quantify the performance of the Smart HPLC Robot, and (2) a real world case study for initial showcase. The simulations for both case studies were performed using the CADET-Core simulator. The mass transfer model used is the native Equilibrium Dispersive Model, and the isotherm model is a non-native linear isotherm coupled with the Linear Solvent Strength (LSS) theory:

$$\text{Linear isotherm: } q_i = a_i\, C_i \qquad \forall i = 1, \dots, n \qquad (1)$$

$$\text{LSS theory: } a_i = a_{0.i} \exp\left(-S_{s,i}\, \phi\right) \quad \forall i = 1, \dots, n \qquad (2)$$

where $n$ is the number of components in the mixture, $q$ is the equilibrium concentration in the stationary phase, $C$ is the concentration in the liquid phase, $a$ is the Henry coefficient, $a_0$ and $S_s$ are the numerical constants in LSS theory, and $\phi$ is the organic solvent composition (i.e., volumetric fraction of the organic modifier in the solvent).

In both case studies, initial experimental conditions are generated via LHS, which are then performed either in-silico using the aforementioned model or sent to the instruments (see setup in later section) to perform real experiments. A virtual column is used in the in-silico system case study, while the Agilent Poroshell 120 EC-C18 column (G7104C096) is used in the real world case study. The column information is summarized in Table 1.

Parameter estimation is then performed in-silico by minimizing the mean of the relative errors ($EE_{PE}$) [5] via genetic algorithm (using Python package "geneticalgorithm2"):

**Table 1**. Information of the columns in the case studies.

| Parameters | In-silico | Real |
|---|---|---|
| Col. length (mm) | 150 | 100 |
| Col. radius (mm) | 5 | 3 |
| Total porosity (-) | 0.645 | 0.840 |
| Internal porosity (-) | 0.189 | 0.160 |
| Particle radius (μm) | 2.50 | 1.35 |

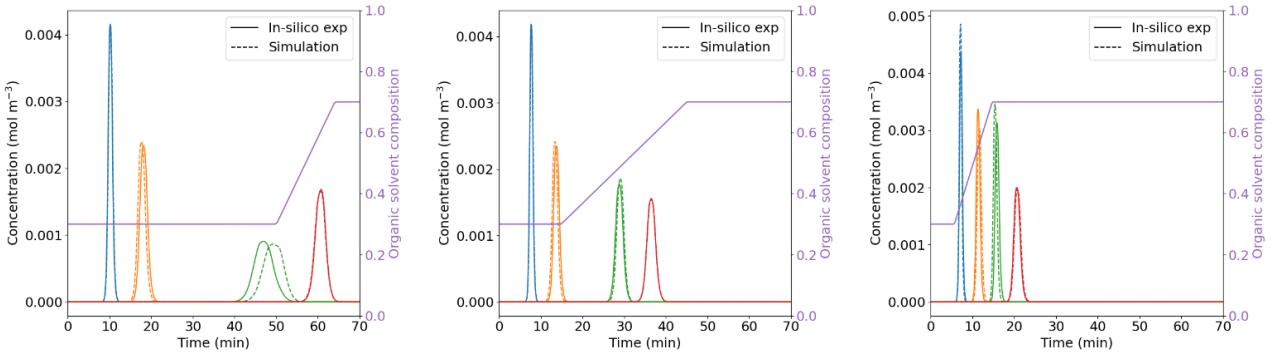
$$EE_{PE} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\left|t_{R,i}^{\text{sim}} - t_{R,i}^{\text{exp}}\right|}{t_{R,i}^{\text{sim}}} \times 100\right) \quad \forall i = 1, \dots, n \qquad (3)$$

where $t_R^{\text{sim/exp}}$ are the simulation (using the mechanistic model)/experimental retention times. The parameters that are varied to minimize the model error $EE_{PE}$ are $a_{0,i}$ and $S_{s,i}$.

Next, the isotherm parameters in the simulation model are updated with those obtained from the parameter estimation step before performing optimization. The optimizer used is also a genetic algorithm. During optimization, the objective function (note that this is not the same as the objective function in parameter estimation) is to minimize the method time (i.e., the retention time of the last eluting component) subject to a critical resolution constraint (i.e., the smallest resolution of all component pairs) of at least 1.5. The critical resolution is calculated considering the peak asymmetry and peak width at 0.135 peak height as suggested by Boelrijk et al. [2]. The optimization variables (and model inputs) are the flow rate, gradient start time, and gradient change duration. The gradient starting and ending volume fractions are fixed at 0.3 and 0.7 for the in-silico system case study, and 0.5 and 1.0 for the real world case study (but could also be optimized). The "optimal" experimental conditions are then performed in the HPLC platform considered, and the model parameter estimation and HPLC method optimization steps are repeated until the stopping criteria are satisfied. Here, a simple stopping criterion of a maximum of four mechanistic loops was used in both case studies.

### (1) In-silico System Case Study

An in-silico system is used as a case study so that the performance of the Smart HPLC Robot can be analyzed quantitatively as the true values of the parameters involved are all known. A four-component mixture is considered. In this case study, simulations and in-silico (i.e., virtual) experiments are carried out using the simulation model described above without considering any uncertainties in the model parameters. Note that for the in-silico experiments, the isotherm parameters are pre-defined and fixed. Therefore, theoretically, the parameter




*(a) Worst parameter estimation case* *(b) Best parameter estimation case* *(c) Final optimized condition*

**Figure 3.** Chromatograms of simulation and in-silico experiments obtained from (a,b) parameter estimation and (c) optimization. All chromatograms are obtained at the last mechanistic loop (loop 4).

estimation should be capable of yielding the true isotherm values, but the optimizer cannot guarantee global optimality, so there is still a small deviation of about 4% absolute average difference in this case study. The parameter estimation (see Equation 3 for the objective function) is performed on all previously executed experiments, and Figures 3a and 3b show the bad and good cases, respectively. It is clearly seen that the simulation and in-silico experiment chromatograms for "fast" (i.e., short method time) cases have greater similarities (i.e., indicating better predictions of the isotherm parameters), which benefits the objective function as this is to minimize the method time.

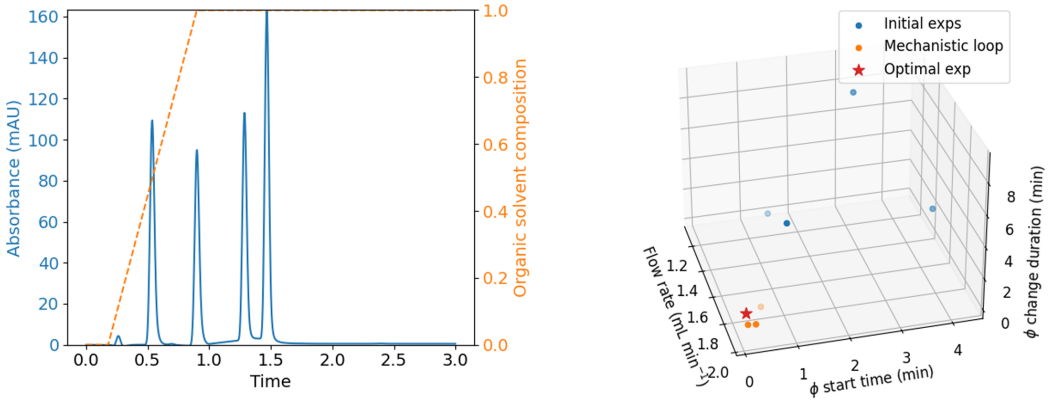
Figure 3c shows the optimized chromatogram for the in-silico experiment and the associated simulation. The results deliver two main takeaways: (1) the chromatograms nearly overlapped, indicating a fairly accurate prediction of isotherm values; (2) all components are completely resolved with a short/minimum gap between adjacent components, illustrating a good quality of the optimized experiment condition.

## (2) Real World Case Study

Having validated the performance of the Smart HPLC Robot, a real world case study is carried out to demonstrate its applicability to laboratory-based experiments. The simulations are performed using the same mechanistic model as for the in-silico system case study, and the real experiments are carried out with an Agilent 1260 Infinity II system. The system is equipped with a quaternary pump (G7104C), an online sample manager set (G3167AA) that consists of an auto-sampler and an online valve configured for off-line sampling, a multicolumn compartment with thermostat (G7116A), and a variable wavelength detector (G7114A). The UV absorbance is recorded at a wavelength of 254 nm and a frequency of 5 Hz. The needle is set to draw the sample at 100 µL min$^{-1}$, eject at 400 µL min$^{-1}$, and an equilibration time of 1.2 s. A four-component sample is used as model sample, with an injection volume of 1 µL.

Compared to the in-silico case study, two new challenges arise for real system. Firstly, the raw experimental data from Agilent provides UV absorbance, and calibration is required to convert absorbance to concentration, but this requires prior knowledge of each component which is sometimes unavailable. Therefore, as done in Case Study 1, component retention times are used as the sole performance indicator for parameter estimation (see Equation 3). The second challenge is that numerical diffusion may contribute significantly to the band broadening in the simulation if the discretization grid of the column is not fine enough. Conversely, a fine discretization may increase the computational burden, leading to a long simulation time. To obtain accurate retention times and peak widths within a short time, this work uses a loose discretization grid to estimate only the retention time, and the peak widths are extracted directly from the experiments. In the first iteration, the peak width of each component is based on the average from initial experiments, and the chromatogram is approximated by a Gaussian curve centered on the retention time. In subsequent iterations, the previous iteration's peak width is used, enabling progressive self-correction and refinement of the method.

As the simulation and experiment generate different profiles (concentration vs absorbance), only the final optimal experimental results are shown in Figure 4a. It is noted that the first small peak is the solvent front, which is not considered a component during parameter estimation and optimization. After four iterations of the mechanistic loop, the method time is minimized to about 1.49 min while achieving complete separation (critical resolution $\geq 1.5$) of all components, which is 49% to 78% faster compared to the initial experiments. In Figure 4b, the distribution of all experiments performed is plotted together. It shows that the initial experiments are spread across the search space and are away from the final optimal experiment, showing how simulations helped eliminate the real




*(a) Optimal chromatogram* *(b) Experiment propagation*

**Figure 4.** Key indicators of the real system case study: (a) The optimal chromatogram obtained from real experiment; (b)The propagation of the experiments throughout the whole automation process.

experiments that are otherwise needed to finally arrive at the optimal point. Also, note that the optimized experiments are close to each other, indicating that they have similar performances. Therefore, the goal is not necessarily to develop a perfect model, but rather a robust method that achieves reliable results. Ideally, this would be complemented by a well-calibrated mechanistic model, which can provide added value, such as enabling sensitivity studies and further refining the process.

## CONCLUSIONS

This work has presented an autonomous HPLC method development program coupled with a mechanistic model, here termed the Smart HPLC Robot. Unlike literature methods, which rely solely on a data-driven approach, incorporating a mechanistic model retains the physics of the system, providing valuable insights and enabling easy model reuse. Throughout the process, the model automatically calibrates itself using real-time data feedback, making it a digital HPLC twin.

The Smart HPLC Robot's performance was validated through an in-silico case study, demonstrating its ability to accurately predict isotherm model parameters and optimize HPLC methods. In a real-world case study, it efficiently explored the search space, improving the HPLC method (e.g., minimizing method time). Notably, the goal of the Smart HPLC Robot is to deliver an optimal method, not a perfect model.

To conclude, the Smart HPLC Robot has two main benefits: (1) reduction in experimental costs and burden through efficient search space exploration, and (2) reduced manpower by enabling self-driven method development until “satisfaction” compared to traditional iterative experimental workflow through automation.

## ACKNOWLEDGEMENTS

We gratefully acknowledge funding from UCL's Impact Acceleration Account, which supported this project, as well as Eli Lilly for their specific support and Agilent for their valuable contributions, all of which were vital to the success of this work

## REFERENCES

1. Besenhard MO, Tsatse A, Mazzei L, Sorensen E. Recent advances in modelling and control of liquid chromatography. *Curr Opin Chem Eng* 32:100685 (2021). 10.1016/j.coche.2021.100685
2. Boelrijk J, Ensing B, Forré P, Pirok BWJ. Closed-loop automatic gradient design for liquid chromatography using Bayesian optimization. *Anal Chim Acta* 1242:340789 (2023). 10.1016/j.aca.2023.340789
3. Tom G, Schmid SP, Baird SG, Cao Y, Darvish K, Hao H, et al. Self-Driving Laboratories for Chemistry and Materials Science. *Chem Rev* 124(16):9633–732 (2024). 10.1021/acs.chemrev.4c00055
4. Dixon TM, Williams J, Besenhard M, Howard RM, MacGregor J, Peach P, et al. Operator-free HPLC automated method development guided by Bayesian optimization. *Digit Discov* 3(8):1591–601 (2024). 10.1039/D4DD00062E
5. Tirapelle M, Besenhard MO, Mazzei L, Zhou J, Hartzell SA, Sorensen E. Towards A Digital Twin for Analytic HPLC. In Texas, US; 2023.