Digital Discovery

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal | View Issue

Cite this: *Digital Discovery*, 2025, **4**, 3126

SEARS: a lightweight FAIR platform for multi-lab materials experiments and closed-loop optimization

Ronak Tali, [a] Ankush Kumar Mishra,[a] Devesh Lohia,[a] Jacob Paul Mauthe,[b] Justin Scott Neu, [c] Sung-Joo Kwon,[d] Yusuf Olanrewaju, [b] Aditya Balu,[a] Goce Trajcevski,[a] Franky So, [b] Wei You, [c] Aram Amassian *[b] and Baskar Ganapathysubramanian *[a]

The Shared Experiment Aggregation and Retrieval System (SEARS) is an open-source, lightweight, cloud-native platform that captures, versions, and exposes materials-experiment data and metadata *via* FAIR, programmatic interfaces. Designed for distributed, multi-lab workflows, SEARS provides configurable, ontology-driven data-entry screens backed by a public definitions registry (terms, units, provenance, versioning); automatic measurement capture and immutable audit trails; storage of arbitrary file types with JSON sidecars; real-time visualization for tabular data; and a documented REST API and Python SDK for closed-loop analysis (*e.g.*, adaptive design of experiments) and model building (*e.g.*, QSPR). We illustrate SEARS on doping studies of the high mobility conjugated polymer, pBTTT, with the dopant, F4TCNQ, where experimental and data-science teams iterated across sites using the API to propose and execute new processing conditions, enabling efficient exploration of ternary co-solvent composition and annealing temperature effects on sheet resistance of doped pBTTT films. SEARS does not claim novelty in these scientific methods; rather, it operationalizes them with rigorous provenance and interoperability, reducing handoff friction and improving reproducibility. Source code (MIT license), installation scripts, and a demonstration instance are provided. By making data findable, accessible, interoperable, and reusable across teams, SEARS lowers the barrier to collaborative materials research and accelerates the path from experiment to insight.

Received 30th April 2025
Accepted 23rd September 2025

DOI: 10.1039/d5dd00175g

rsc.li/digitaldiscovery

1 Introduction

Materials science research is undergoing a paradigm shift in how experiments are conceived and executed, enabled by the advancements in automation, data-centric methods, and the advent of "self-driving" laboratories.[1] Modern laboratories increasingly integrate robotic automation and high-throughput techniques, enabling experiments to be planned and run with minimal human intervention. In tandem, data-driven decision-making allows researchers to rely on mining prior experimental data and leveraging machine learning (ML) models to guide new experiments. An important aspect of these trends is that they are often accompanied by an explosion of data both in volume and complexity. Thus, it is extremely beneficial for scientists to have access to both large repositories of past experimental data and sophisticated software tools to search and extract relevant information for planning their next steps.

Advanced multiscale characterization techniques now yield rich multi-modal datasets[2] – for example, structural, thermal, and electrical properties of a material might be measured across different scales and saved in diverse file formats (*e.g.*, spectroscopy files, microscopy images). Organizing, storing, and retrieving such heterogeneous data streams pose significant challenges. Repeating measurements on the same specimen (common for reliability) further adds layers of data that must be tracked and versioned. To address these challenges and ensure that data can drive decision-making, the community is embracing FAIR data principles (Findable, Accessible, Interoperable, Reusable).[3] Adhering to FAIR standards means recording detailed metadata (instrument settings, sample preparation details, *etc.*) using well-defined ontologies[4] and making datasets available in standardized formats. Such practices not only improve reproducibility for independent researchers but also unlock new opportunities for integration with third-party tools. For instance, well-structured data can

[a]*Iowa State University, Ames, IA 50010, USA. E-mail: rtali@iastate.edu; akmishra@iastate.edu; devesh@iastate.edu; baditya@iastate.edu; gocet25@iastate.edu; baskarg@iastate.edu*

[b]*North Carolina State University, Raleigh, NC 27695, USA. E-mail: jpmauthe@ncsu.edu; yaolanre@ncsu.edu; fso@ncsu.edu; aamassi@ncsu.edu*

[c]*University of North Carolina, Chapel Hill, NC 27599, USA. E-mail: neuj@email.unc.edu; wyou@email.unc.edu*

[d]*University of Washington, Seattle, WA 98195, USA. E-mail: sjkwon12@uw.edu*

feed directly into ML algorithms or be queried by emerging large language model (LLM) assistants.[5] Interoperability is particularly crucial as standardized data formats enable different experimental workflows and database systems to connect and exchange information seamlessly. This allows the formation of a federated ecosystem of knowledge rather than isolated data silos.

Another key aspect of this evolving landscape is collaborative, cloud-enabled research. Given the increasing complexity of modern materials problems, no single laboratory can house all necessary expertise or instrumentation, and geographically distributed teams are becoming the norm.[6] To accelerate discovery, researchers need to collaborate across institutional boundaries, often in real time. This necessitates robust, cloud-based data infrastructure and electronic lab notebooks that multiple labs can access and contribute to concurrently. Essential features include customizability (to accommodate each lab's protocols and data types), reliability and version control (to track contributions and changes), and fine-grained access control (so that each team member can work on a shared experiment securely).[7] A cloud-centric approach ensures that data and analysis tools are available on-demand to all collaborators, eliminating traditional barriers of local data storage and allowing experiments to be monitored or even steered remotely. In this context, rather than classical trial-and-error approaches, researchers can harness ML as a tool to predict outcomes and suggest promising experimental conditions.[8] Once an ML model is (well) trained on existing data, it can rapidly screen hypothetical scenarios and recommend the most likely candidates for success, potentially narrowing down the experimental search space. This capability transforms the role of the experimenter – guiding experiments by computational insight – and accelerates the cycle of hypothesis to validation. Moreover, sharing such predictive models (in addition to raw data) between laboratories increases the beneficial impacts by enabling researchers to learn models and improve transfer learning in general.

We present SEARS (Shared Experiment Aggregation and Retrieval System), an open-source, cloud-native platform that operationalizes these trends by capturing, versioning, and exposing materials-experiment data and metadata through interoperable, programmatic interfaces. SEARS provides configurable, ontology-driven data-entry screens; a scalable document store with raw-file storage and JSON sidecars; search, tagging, and built-in version control; and a documented REST API with a Python SDK for analysis and closed-loop experimentation. Multiple laboratories can contribute to a shared record in real time, with provenance (owner, lab, timestamps) recorded consistently, and FAIR-compliant exports available for publication or downstream tools. By design, SEARS can be self-hosted and extended (MIT license), allowing teams to integrate existing workflows while preserving reproducibility and auditability.

We illustrate SEARS with several case studies (Sections 6.1 and 6.2) including adaptive design of experiments (ADoE) and quantitative structure–property relationship (QSPR) modeling. Our claim is not novelty in ADoE or QSPR, but that SEARS provides the enabling infrastructure (provenance-rich capture, multi-lab coordination, and API/SDK access) that makes such established methods practical for distributed collaborative teams. We demonstrate this on distributed studies of pBTTT:F4TCNQ, where experiment and data-science teams iterated across sites to refine ternary co-solvent composition and annealing temperature and to train reproducible QSPR models using consistently annotated data.

2 Background

A variety of digital tools and data platforms have recently been developed to support data-driven materials science. For example, the High-Throughput Experimental Materials (HTEM) database introduced a centralized repository for sharing materials synthesis and characterization data.[9] While HTEM and similar domain-specific databases provide valuable resources,[10–12] most are designed primarily for data dissemination (*i.e.*, users can upload or download datasets) and read-only exploration of published results. In many cases, external researchers can retrieve data but cannot actively contribute new experimental results or collaboratively build on the database's content. Other pain points identified by the community include inconsistent data quality, incomplete metadata, limited community adoption, and uncertain long-term sustainability of the platforms. In short, early materials databases laid important groundwork for open data, but their rigid schemas and single-project focus have impeded broader reuse of the data for diverse research questions.

Recognizing these issues, the community has argued for flexible and FAIR-centered data infrastructure in chemistry and materials. For instance, adopting FAIR principles in database design has been emphasized as crucial for enabling not only findability but also reusability of data across projects.[13] A key recommendation is to record all experimental outcomes – including failed or null results – alongside successful data. Logging negative results provides context for interpreting model predictions and helps quantify experimental error, ultimately improving the robustness of any data-driven analysis. Some initiatives have attempted to modernize legacy chemistry databases with more flexible data models: the Royal Society of Chemistry's ChemSpider platform was partially rebuilt on a NoSQL backend to accommodate new data types.[14] This change improved extensibility but resulted in a complex hybrid architecture (traditional relational databases coupled with NoSQL stores) that proved hard to maintain and scale. Other proposed solutions have approached data sharing by distributing the data directly to end-users. For example, the Materials Provenance Store (MPS)[15] concept involves researchers downloading an entire copy of a database to their local machine and using custom scripts or SQL queries to mine the data. While this ensures full access to the dataset, it creates formidable challenges: every user must overcome steep technical setup and update the data copy regularly, and querying large datasets locally becomes inefficient.

These examples underscore the difficulty of designing data systems that are both powerful and user-friendly. More recent

frameworks have capitalized on web technologies to broaden data accessibility. The NOMAD[16] repository is one prominent web-based platform that allows materials scientists to upload their results along with rich metadata, making those data publicly searchable and viewable through an online interface. Notably, NOMAD incorporates advanced tools like AI-driven analysis modules and interactive visualizers to help users derive insights from shared data. However, platforms like NOMAD remain after-the-fact repositories – they are geared toward publishing completed results rather than facilitating real-time collaboration during the experimentation process. Once data are uploaded, contributing laboratories typically relinquish some control over how those data are managed and updated. This approach can yield valuable datasets but leaves little opportunity for new laboratories to directly participate or continuously contribute fresh data in an open forum. Finally, there have been case studies demonstrating integrated data workflows within individual labs. One recent example from an electrochemistry laboratory showcased a custom pipeline that spanned from automated data acquisition to analysis and visualization, using a suite of open-source tools orchestrated for that lab's needs.[17] This "digital lab" case study highlighted the efficiency gains from linking instruments to data processing in a feedback loop. While this solution was highly specific to that lab's setup and was not released as a generalized tool for others, such solutions represent a powerful tool to accelerate acceleration. Such tools often do not address multi-lab collaboration, focusing only on internal data management.

Within this broader push toward interoperability and reuse, materials-science ontologies typically organize concepts into four classes: substance, process, property, and environment.[18] To make these classes actionable across collaborators and tools, it is useful to bind them to a single, public definitions registry that assigns each term a unique identifier and version; a human-readable label; a formal definition; units and, where applicable, permissible values; provenance (originating lab/person); creation and last-updated timestamps; and cross-references. Such a registry provides an unambiguous, searchable reference for every piece of information, improving consistency, reuse, and auditability across studies.

In this broad context, SEARS is designed to fill the existing gap by offering a unifying framework that builds on prior lessons – enabling broad participation, enforcing interoperability and FAIR standards, and empowering researchers to seamlessly share, retrieve, and act on data across laboratory boundaries. While SEARS is designed to be material agnostic, we illustrate the capabilities and utility of SEARS through several popular use cases and a detailed discussion of a multi-university case study to understand and increase the mobility of conjugated polymers through doping.

3 Goals

The design principles of SEARS are centered on flexibility and pluggability. SEARS achieves flexibility by fully leveraging the capabilities of NoSQL databases,[19] specifically MongoDB,[20] to store experimental data as isolated JSON-like objects. This structure allows each experiment to have a unique ontology while maintaining the adaptability to modify the ontology for future experiments. Such room for a flexible ontology classifies SEARS as an ontology-driven platform as opposed to an ontology-enforced platform. In terms of pluggability, SEARS is built as a loosely coupled collection of components, with synchronization managed entirely through REST[21] APIs. This architecture enables independent deployment of components across diverse cloud platforms while allowing seamless integration with machine learning, visualization, and FAIR-compliant data management tools. With these foundational principles in place, we outline the key goals of SEARS below.

3.1 Storage

Experimental data typically consists of three primary forms: (i) metadata describing the experiment (*e.g.*, instrument details, polymer information), (ii) numerical experimental results (*e.g.*, thickness measurements), and (iii) raw experimental files (*e.g.*, UV-visible spectra). SEARS is designed to support the storage of all three data types efficiently. From a security perspective, all data entries are redundantly backed up to ensure robustness. Additionally, SEARS allows for metadata enrichment, tracking the entire experimental lifecycle as data flows between collaborating research laboratories. As part of our standard operating procedure, SEARS runs an automated pre-ingest validation script on uploaded files (*e.g.*, checks for missing columns/values, row-count anomalies, and basic format sanity). Because SEARS is intentionally domain-agnostic, these validations are designed to be configurable per deployment.

3.2 Search

SEARS supports two primary search functionalities. First, laboratory members can search for experiments using intuitive keyword-based queries through a front-end interface. Second, advanced search *via* the REST API and Python SDK lets analysts retrieve metadata, numerical readings, and raw files (*e.g.*, UV-visible spectra) using filters such as owner, lab, tags, and date ranges.

3.3 Collaboration

Facilitating seamless collaboration is a core objective of SEARS. Many existing frameworks, as discussed in Section 2, operate under an individual ownership model, where experimenters retain control over their respective datasets. SEARS, in contrast, introduces a joint ownership model, wherein collaborating research laboratories collectively manage and share experimental data. This approach fosters a continuous data flow across research teams, allowing multiple group to create, update, and analyze shared datasets within a unified system.

3.4 User interface (UI)

The SEARS user experience (UX) is designed to fulfill three key objectives. First, it ensures data security and controlled access for all users. Second, it provides a comprehensive command-and-control dashboard that centralizes all experimental data

management tasks—allowing users to create experiments, update records, log measurements, upload/download files, and share data *via* QR codes, all from a single interface. Finally, the UI prioritizes ease of onboarding, ensuring that new users can quickly navigate and utilize SEARS with minimal learning overhead.

3.5 Versioning

In the natural progression of an experiment, both within and across research teams, it is essential to maintain a detailed record of the experiment's history. This history may include new experimental readings or additional files, such as updated UV-visible spectra at different points of an experimental process. SEARS is designed to support and store a complete version history of all experimental events. Additionally, users can conveniently download all relevant data through the command-and-control dashboard.

3.6 FAIR and ML

SEARS provides FAIR-by-design access to experiment data with minimal effort. From the primary interface, users can obtain FAIR-compliant exports; a central registry of ontological definitions offers built-in search and direct downloads of JSON sidecars. Every data record in SEARS is linked to a definition in this registry. For machine learning, independent users can pull data directly into an IDE *via* a REST API (portal). Access is administered through the portal: once approved, users receive a unique API key (default validity 90 days) to query, filter, and retrieve data in JSON. The REST API connects directly to the underlying MongoDB database; step-by-step usage instructions are provided here. (Access policies and key extensions are managed by local SEARS administrators.) Our design goal is to surface intuitive access controls in the main UI so FAIR exports remain straightforward.

In the case studies reported here, we exercised these same capabilities end-to-end: data were captured under SEARS ontology terms, exported as FAIR packages, and programmatically consumed to support ADoE iterations and QSPR model training. To illustrate this workflow for readers, we provide resources comprising (i) a de-identified exemplar FAIR dataset with a corresponding CSV representation and (ii) a notebook demonstrating the analysis applied to one case study. The case-study summaries included here are on independent publication tracks; full datasets will be released with their respective publications. Comprising (i) a de-identified exemplar FAIR dataset with its corresponding CSV representation, and (ii) a notebook demonstrating the data analysis workflow applied to one of our case studies. The case study summaries included in this work are on independent publication tracks; full datasets for these studies will be released in conjunction with their respective publications.

4 Database schema

To achieve the goals outlined in the previous section, SEARS employs a NoSQL database. NoSQL databases differ fundamentally from traditional SQL databases, which enforce a rigid schema requiring users to define *a priori* the structure of each data entry. In SQL databases, every unit of data must adhere to a predefined schema, preventing the storage of heterogeneous data structures within a single dataset. In contrast, NoSQL databases, such as MongoDB (used in SEARS), store data as nested key-value pairs, referred to as objects, rather than linear rows. Fig. 1 presents a conceptual layout of our database schema built on our cloud-based MongoDB instance.

Unlike SQL databases, which rely on tables, columns, and rows, NoSQL databases organize objects within flexible constructs known as documents. Importantly, these documents do not require uniform structures, allowing the storage of objects with varying attributes within the same collection. Additionally, NoSQL databases do not rely on traditional concepts like joins and normalization, prioritizing schema flexibility over strict relational consistency. However, this flexibility comes at a trade-off: NoSQL databases do not inherently provide ACID[22] guarantees, which ensure transactional reliability in SQL databases. This is not a limitation in the context of SEARS, as ACID compliance is typically crucial for online transaction processing (OLTP),[23] which is not the primary focus of our system.

Fig. 1 illustrates key features of our database schema. First, each experiment, along with its metadata and associated measurements, is stored as a self-contained, nested object, mirroring the native object structure of JavaScript.† Second, SEARS does not impose constraints such as predefined column names or data types, eliminating the need for rigid headers. Third, each experiment is treated as an independent unit, allowing documents with different structures to coexist within the same database collection. Fourth, this inherent flexibility enables modifications to experiment schemas dynamically, without affecting previously stored data. This adaptability extends to other system components, such as the application front end, allowing iterative improvements without disrupting historical records. Finally, NoSQL databases offer significantly faster read and write operations compared to SQL databases,

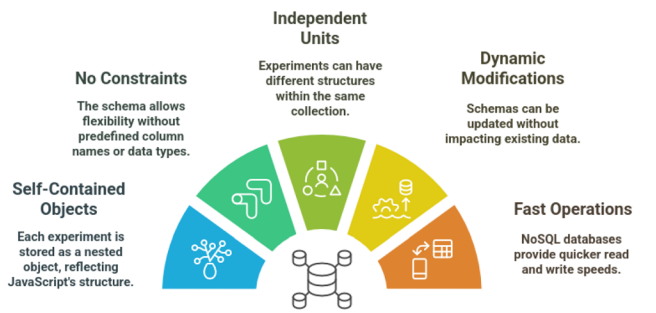

Fig. 1 SEARS: features of the database schema design.

† This aligns naturally with how JavaScript defines objects.

making them well-suited for handling large-scale experimental datasets efficiently.

5 System components

In this section, we present the complete architecture of SEARS. Fig. 2 shows the layout and wiring of all the components implemented in SEARS. This layout ingrains several forward-looking properties. Chief among these is the need to support extensive visualization and complex machine learning use cases such as the storage of trained models and using these saved models to generate real-time predictions (inference).

We ensured our components were loosely coupled and interacted with each other purely *via* REST APIs.[21] This means that as long as the API messaging interfaces remain consistent, we can deploy an updated version of any single component with no impact whatsoever on other components, with just a momentary downtime. This also means plugging in newer components can be achieved without impacting other running components and with no downtime. Second, we adopted a cloud-based deployment model, stitching together services from different cloud providers. This heterogeneity allows us to utilize best-in-class services from different providers and, importantly, prevents the chances of a single point of failure. Finally, we focused on building a joint ownership model of the experiment data. We do this by providing a complete set of features for research laboratories to seamlessly work together on a single experiment such that they have full visibility on every new experimental data point or file entered into SEARS. We also make it easy to track the entire history of the experiment for all members.

5.1 Secure front-end

The SEARS front-end is designed to provide a seamless and intuitive user experience. As an application accessible over the public internet, SEARS incorporates a secure authentication layer, managed through a third-party service, ensuring that only authorized users can access the system. The front end is built entirely using the open-source React.js JavaScript framework,[24] complemented by various third-party open-source React components to enhance functionality. The SEARS front end offers several key features:

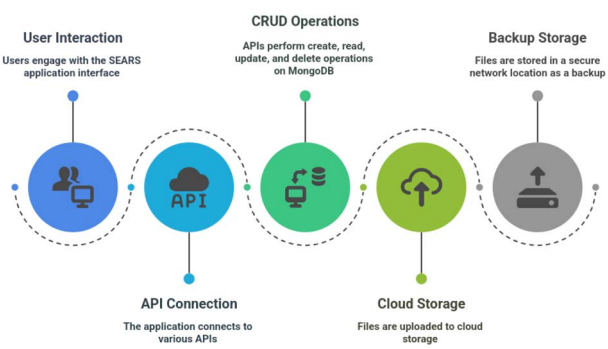

Fig. 2 SEARS: layout and wiring of the system components.

• Central dashboard: the central dashboard serves as the primary interface for users upon logging into SEARS. It provides a comprehensive control panel where users can view, create, update, upload, download, search, and share experimental data. Each experiment entry displays essential identifiers such as experiment name, creation date, associated laboratory, and owner. To facilitate rapid access, each experiment is assigned a unique Quick Response (QR) code, which can be scanned to quickly retrieve experiment details. Additionally, the dashboard features a search bar for users to efficiently locate experiments by name. The dashboard includes filters for Owner and Lab to scope views and downloads to specific contributors or groups.

• New experiment creation: this feature enables users to enter all relevant metadata associated with an experiment on a single screen. The interface organizes metadata into clearly defined sections for ease of use. A key characteristic of this functionality is that once an experiment is created, its metadata remains immutable, ensuring consistency and integrity.

• Viewing and editing an existing experiment: SEARS provides a structured interface for managing experimental data, organized into multiple tabs representing different measurement categories, such as sample thickness. Within each tab, users can input measurement values for different batches as needed. If a category requires the upload of raw data files (*e.g.*, current–voltage characteristics), users can drag and drop the relevant CSV file into the designated area. Once uploaded, these files are available for download and visualization. By default, the first two columns of a CSV file are automatically assigned as the x and y values for data plotting, streamlining the visualization process. It is important to note that while we are able to plot data only from uploaded csv files in real time, we support uploading of files of any type.

5.2 Metadata and measurements storage

Each experiment in SEARS consists of three primary components: metadata, measurements, and files. This section describes how metadata and measurements are stored within the system. Fig. 1 illustrates how these data elements are structured in our MongoDB database.

NoSQL databases are designed with the principle that data accessed together should be stored together. This contrasts with SQL databases, which prioritize data sparsity and schema normalization to minimize redundancy. NoSQL databases follow the document object model, where collections of objects (analogous to rows in SQL databases) are aggregated into documents (similar to tables in SQL). A key advantage of this approach is its inherent flexibility: unlike SQL databases that enforce rigid column structures, NoSQL databases allow dynamic and adaptive data storage. Objects store information as key-value pairs, provided they are serializable and can be transmitted over a TCP network. Additionally, key-value pairs can be encapsulated within arrays, allowing for hierarchical and structured data storage.

An inspection of Fig. 1 reveals that these objects closely resemble JSON structures, which are widely used as the *de facto* standard for internet data exchange. SEARS' front-end

application converts all incoming metadata and measurements into a single JSON payload, which is then transmitted to the MongoDB database. MongoDB processes this payload and stores it as a BSON document, a format that closely mirrors JSON but with additional optimizations for storage efficiency. Once stored, MongoDB assigns a unique experiment ID, which allows the front-end application to retrieve and organize experiment data efficiently. Furthermore, all stored data undergo automatic backups to ensure protection against potential data loss, maintaining the integrity and reliability of the system.

We would like to make a special note of the material-agnostic nature of SEARS. This means that the way data is stored and ultimately viewed with SEARS can be completely customized at the time of deployment. This includes both the metadata and the measurements. While we have pre-configured SEARS for illustrative purposes, we provide a detailed SEARS customization guide with examples in our software repo. Section 5.7 provides more details on customization.

5.3 Data types and FAIR exports

The upload sections in SEARS support all kinds of file uploads. Which means, users can choose to upload practically any file including csv files, flat files, images to SEARS. In case the data contained in these files have not already been defined in our central registry, we strongly encourage the user to upload a json sidecar along with the main data file. SEARS also allows users to export experimental data and uploaded measurement files from the main dashboard. Two conveniently placed buttons are positioned near the name of the experiment to facilitate downloads. The downloaded data is in FAIR format and the associated JSON sidecar can be downloaded from the SEARS FAIR portal.

5.4 File stores

All files uploaded by users through the SEARS front-end application are stored in parallel across two independent storage locations to ensure redundancy and reliability. The first copy of each uploaded file is sent to a cloud-based object storage system, where it is organized into a directory structure based on the unique experiment ID assigned by MongoDB. This structured organization facilitates easy retrieval and management of experimental files.

To enhance data durability, a replica of each file is simultaneously stored in an in-house long-term storage system. To further safeguard against data loss, each storage location employs triple replication, meaning that every file is backed up three times per location. As a result, each file in the SEARS framework is maintained in six independent copies, distributed across both cloud and local storage infrastructures. This comprehensive replication strategy ensures that all high-value experimental data remain secure and resilient against potential system failures or accidental deletions.

5.5 API ecosystem

The SEARS framework is built on a network of REST APIs, which facilitate seamless communication between the front-end application, databases, and file storage systems. REST APIs ensure that all components remain modular and interact through well-defined interfaces, where each component both sends and receives messages in a predefined format. This structured approach enhances system stability, security, and scalability by maintaining clear boundaries between different services. The REST API exposes filter parameters (*e.g.*, owner, lab, date ranges) for programmatic queries and FAIR exports. We also provide a Python SDK and an OpenAPI specification to streamline scripted access and integration with notebooks and pipelines.

Fig. 3 illustrates the SEARS API ecosystem, which is encapsulated within a secure authentication layer. Within this framework, we implement/utilize several specialized API categories:

- Database APIs: these APIs support CRUD (Create, Read, Update, and Delete) operations for managing experiment data in MongoDB. The front-end application interacts with these APIs to execute all database-related transactions. To optimize cost and performance, SEARS leverages MongoDB Atlas cloud functions for executing database operations efficiently.
- Storage APIs: these APIs handle file storage and retrieval by interacting directly with both cloud-based and in-house storage solutions. To maintain consistency, all write operations are executed in parallel across both storage locations, ensuring redundancy and data integrity. The storage APIs are implemented using the FASTAPI framework, selected for its performance and robust security features.
- Internal APIs: in addition to core database and storage functions, SEARS includes internal APIs dedicated to logging and system health monitoring. These APIs are restricted to framework administrators and are not accessible to standard users.

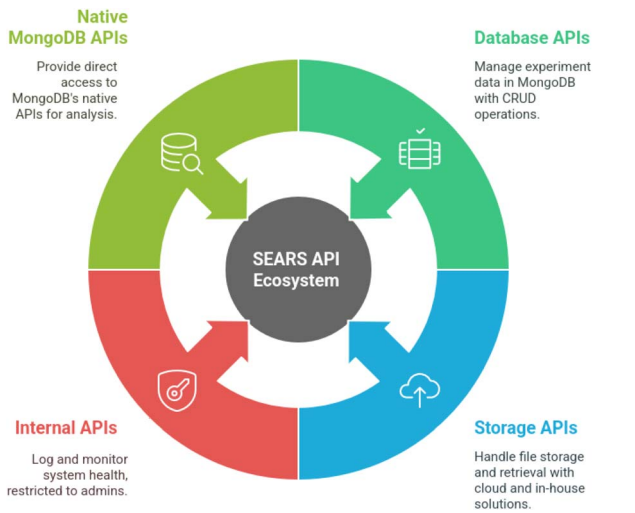

Fig. 3 SEARS: API Ecosystem.

• Native MongoDB APIs: SEARS also provides direct access to MongoDB's native APIs, allowing users to retrieve structured experimental data for advanced analysis, machine learning workflows, and data-driven decision-making.

This API-driven architecture ensures that SEARS remains modular, scalable, and efficient, enabling simple integration with external tools and emerging data science applications.

5.6 Security and provenance

SEARS implements a trust-circle plus immutability model, ensuring controlled data intake within a multi-lab setting, which means we assign equal access to every lab(s) member but do not allow anyone to delete any data. This ensures maximum visibility of data among collaborating labs while preventing any possibility of inadvertent data loss. The authentication framework is built around MongoDB Atlas, which manages the issuance and validation of authentication tokens. By leveraging MongoDB Atlas's authentication system, SEARS seamlessly integrates secure token-based access control within its API ecosystem. This eliminates the need for a custom authentication mechanism, reducing security risks while ensuring an industry-standard authentication process. This architecture provides a secure and efficient method for managing user access. For deployments that require additional oversight, SEARS includes an administrative audit view summarizing write operations by user and time window, leveraging the system's immutable history. We note that in our future releases, we aim to provide for more granular role-based authentication (RBAC) wherein a lab manager assigns explicit visibility of experiments to different lab members. In addition, our REST API (portal) now includes a UX for users to write queries with various filters.

5.7 Discussion

In order to ensure the convenience and potential benefits of SEARS to the broader material science research community, in addition to conducting several use-cases (detailed in the next Section) we also conducted different groups of testing throughout its development. Specifically, each of the main components (*cf.* Fig. 2) has been subjected to a unit-testing by carefully selecting samples of inputs and observing the correctness of the outputs. In addition, we also conducted interface and integration testing to ensure that the flow of data corresponding to users' requests towards the processing modules and back (*i.e.*, the outcome of executing models, visualization, *etc.*) is performing correctly, based on the scenarios enacted through the use-cases. Lastly, we performed an overall system/acceptance testing regarding the steps involved in its enactment (*e.g.*, accounts creation/verification, data and model accesses, *etc.*).

We close this section with a note that the implementation of SEARS is publicly available at GitHub. This repository contains complete explanations to download, set up and customize SEARS to your needs. In addition we provide a demonstration video of SEARS here. We also want to make a special note of the role AI-based code editors potentially play in quickly customizing SEARS. Fully customizing SEARS involves edits to one or two files at the maximum; see detailed instructions in our repo. Please refer in particular to our customization guide that provides illustrative examples of how to customize SEARS using AI prompts. We anticipate AI editors to easily comprehend the repeatability in our logic and quickly make edits to suit the end user's needs.

We anticipate SEARS to evolve over time as we incorporate an increasing number of features in our future iterations. Accordingly, our change management policy will entail the following; first, our data schema will remain incremental, that is, we will never delete any data elements, we will only make additions as needed, second, we will provide migration scripts that will allow bulk transfer of documents from an existing schema to a new schema and third, new versions of SEARS will be distributed as new code repositories without touching those for older versions. These safety features will allow SEARS users to remain confident that in case of any issue during version migration, they will always be able to switch back to an older version.

6 Case studies

To illustrate the capabilities of SEARS, we present four case studies to demonstrate its role in experiment-to-analysis pipelines across different laboratory environments. For each researcher participating in these studies, we provided interactive sessions, which introduced them to SEARS' key features and provided hands-on experience in uploading, managing, and analyzing experimental data. The case studies in Sections 6.1 and 6.2 are intended to demonstrate how SEARS supports distributed, closed-loop workflows, not to propose new ADoE or QSPR algorithms. We explicitly separate platform from technique: ADoE and QSPR are standard; SEARS provides the infrastructure (definitions registry, immutable audit, REST/SDK) that makes these techniques reproducible and scalable across labs.

6.1 Case study 1: adaptive design of experiment using SEARS platform

In this case study, SEARS was employed for an Adaptive Design of Experiment (ADoE) campaign aimed at optimizing the doping of a common conjugated polymer, pBTTT, with the molecular dopant, F4TCNQ. The experiment systematically varied the concentrations of three ternary polymer co-solvents — chlorobenzene (CB), dichlorobenzene (DCB), and toluene (Tol) — alongside polymer annealing temperatures to identify conditions influencing sheet resistance of dip-doped pBTTT. The range of concentration of co-solvents varied from 0–100% with a least count of 5%, while following the constraint that together they sum to 100%. The polymer annealing temperature varied from room temperature to 270 °C with a least count of 30 °C.

While ADoE is a well-established approach in materials science[25,26] for optimizing experimental conditions,[27] the use of SEARS enables a collaborative, distributed workflow that would be impractical with traditional methods. In particular, SEARS allows experimental data generated at one or more laboratories to be uploaded, accessed, and analyzed in real time by geographically distributed research teams. This streamlines the

iterative process of model updating and experimental planning: multiple research groups can simultaneously contribute new experimental results, which are immediately available for updating the regression models that drive ADoE optimization. This shared, cloud-based infrastructure removes barriers to collaboration, reduces data silos, and accelerates convergence towards optimal processing conditions. Thus, SEARS facilitates a more efficient, collaborative, and reproducible ADoE workflow, particularly when dealing with large, heterogeneous datasets from multiple contributors.

In our use case, the experimental team uploaded their data to SEARS, where the data science group, at a different geographical location, accessed it *via* a Python script. Using this data, an ADoE framework was applied to propose the next set of processing conditions, refining ternary co-solvent concentrations and polymer annealing temperatures based on prior results. Over three iterative campaigns, SEARS facilitated seamless data exchange, enabling the identification of regions associated with both high and low sheet resistance. The initial processing conditions were selected *via* Latin hypercube sampling, while subsequent adjustments were determined through ADoE optimizations, as shown in Fig. 4. The SEARS platform functioned as a central intermediary between experimentalists and data scientists, ensuring smooth integration. The ADoE samples were further used to understand important features that influence the sheet resistance. A detailed analysis of the features is part of a separate publication.[28]

6.2 Case study 2: quantitative structure–property relationship model using machine learning

In this case study, SEARS was employed as a centralized data management and retrieval platform for developing Quantitative Structure–Property Relationship (QSPR) models. The dataset, consisting of results from LHS and ADoE campaigns, was utilized to establish a predictive mapping between ternary co-solvent concentrations and polymer annealing temperatures on the sheet resistance of doped pBTTT.

To establish a robust predictive mapping, we trained several machine learning models using the aggregated data; the Random Forest algorithm yielded the best predictive performance. The dataset was split into an 80 : 20 train–test ratio, with model results summarized in Table 1. Once trained, the QSPR model enabled rapid prediction of sheet resistance throughout the design space, as illustrated in Fig. 5.

SEARS played a crucial role by providing a unified repository where experimental data from various sources could be uploaded, curated, and accessed in real time. This centralized infrastructure supports continuous and collaborative model development: as new experimental results are contributed by any participating lab, the QSPR models can be instantly retrained and updated, enabling near real-time feedback on experimental progress and facilitating data-driven decision-making. The platform's architecture thus streamlines the otherwise labor-intensive processes of data collation and version control, ensuring data integrity and accessibility for all collaborators.

Table 1 Machine learning model between processing conditions and sheet resistance

Model	Input	Output	R^2
Random Forest	% CB, % DCB % Tol, anneal temp (°C)	Sheet resistance (Ω/□)	0.41

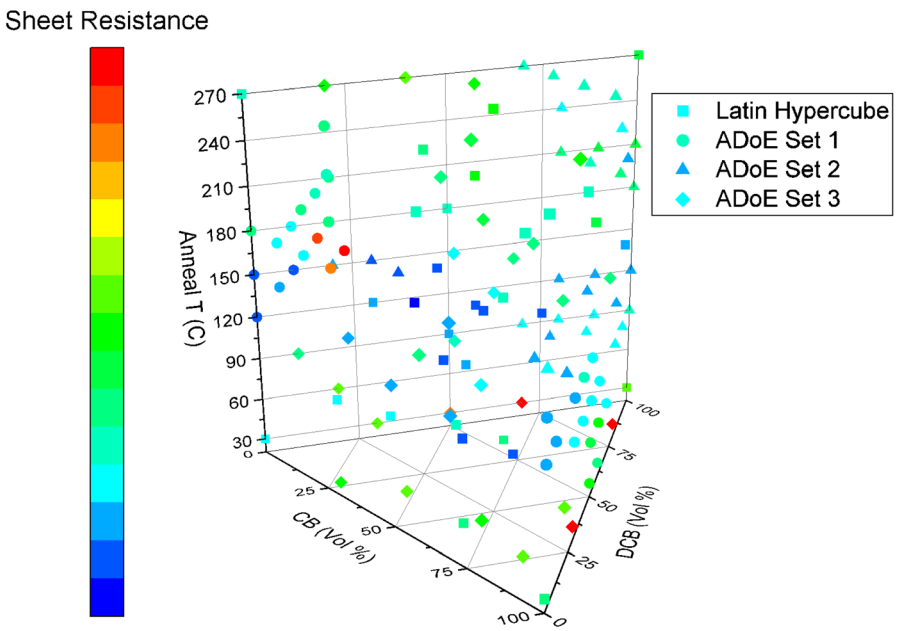

Fig. 4 ADoE batches to determine regions of higher(red points) and lower sheet resistance (blue points) using the SEARS portal as an intermediary between the experimentalist and data scientist.

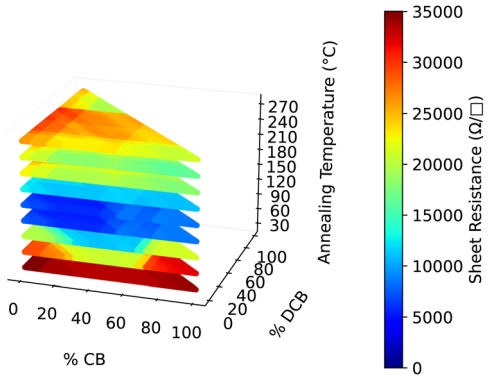

Fig. 5 Quantitative structure–property relationship prediction using random forest.

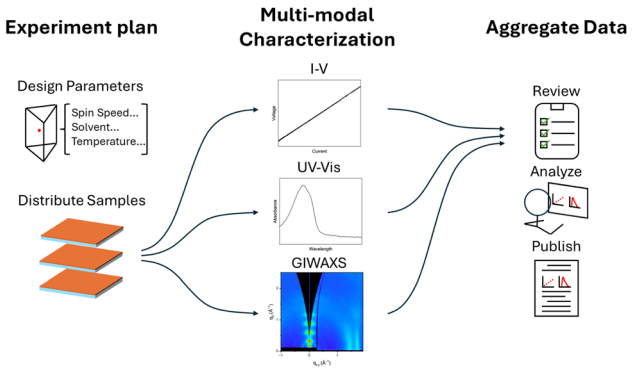

Fig. 6 SEARS: multi-facility, multi-modal data aggregation.

In addition to processing parameters, we extended our QSPR modeling efforts to incorporate spectral data uploaded to SEARS. This enabled a systematic, data-driven identification of spectral features most relevant for explaining variations in sheet resistance and conductivity, and allowed us to directly compare these features with those identified by expert judgment. A detailed analysis of featurization techniques, QSPR models, and feature importance ranking is presented in a separate publication.[29]

While machine learning approaches to QSPR modeling are well-established,[30] the integration with SEARS unlocks several practical advantages. Most notably, it enables interactive and adaptive experimentation: researchers can monitor QSPR model performance in real time, guiding decisions on when sufficient data has been collected or when further experiments are warranted. Furthermore, by aggregating data from multiple laboratories, SEARS supports the construction of more generalizable and robust predictive QSPR models, accelerating both discovery and validation processes in materials science.

6.3 Case study 3: making SEARS data available in FAIR format

In this case study, we illustrate SEARS' capabilities to ensure compliance with FAIR data principles.[31] Our approach is straightforward: for each experiment, SEARS provides an interface allowing researchers to download experiment data in JSON format, making it accessible to the broader scientific community.

Consistent with FAIR principles,[32] we ensure that each JSON key is fully defined and documented *via* our FAIR metadata server, accessible here. The use of JSON format ensures platform agnosticism and serialization, making the data easily accessible and interoperable across most programming environments. By aligning SEARS with FAIR standards, we enhance data reusability, interoperability, and integration with external computational tools, further strengthening its utility in scientific research.

6.4 Case study 4: multi-facility, multi-modal data aggregation demonstration with SEARS

Using a shared, aggregated database for collaborative research enables seamless integration of data from multiple measurement locations, including various instruments, universities, and national labs. This approach enhances data consistency, facilitates cross-comparison, and improves reproducibility by centralizing experimental metadata and standardizing analysis protocols. By leveraging these tools, researchers can efficiently aggregate and analyze large multi-modal datasets.

Additionally, such a database fosters transparency, minimizes redundancy in data collection, and promotes interdisciplinary collaboration. In this case study, we combined complementary four-point probe current–voltage measurements with UV-visible absorption spectra and X-ray scattering data from instruments at NC State University and Brookhaven National Lab (NSLS-II). SEARS facilitated data hosting and visualization, enabling direct comparison of duplicate samples analyzed at both facilities, as illustrated in Fig. 6.

7 Conclusion

We presented SEARS, a novel integrated framework which provides unique opportunities for data collection, management, dissemination, and analysis in materials science research. Through consultations with stakeholders and iterative development, we have refined SEARS to align with the needs of researchers. Currently, SEARS has been successfully deployed in a multi-university research collaboration, where members actively create, update, and share experimental data in real time. The platform's user interface has minimized on boarding time, enabling new users to become proficient within a few hours. Most features are designed to be self-explanatory and easily accessible, ensuring a simple user experience with enhanced control and efficiency.

As discussed in Section 2, there have been efforts to create systems that can improve collaboration among researchers; however, given the generality of SEARS, we anticipate broader adoption of SEARS (and SEARS like tools) across the research community, expanding its role as a scalable and adaptable solution for experimental data management. As part of future work, we intend to extend the capability of SEARS to enable seamless inclusion of novel experimental templates/use cases

and to be adaptable with respect to FAIR regulations in the future. We also intend to include an optional notification trigger in future versions of SEARS. This feature will allow lab members to be notified each time there is any new activity in SEARS. We understand that this could be beneficial in certain low experiment frequency scenarios.

Author contributions

Ronak Tali: methodology, software, data curation, writing – original draft, visualization. Ankush Kumar Mishra: formal analysis, investigation, visualization, writing – review and editing. Devesh Lohia: software. Jacob Paul Mauthe: investigation, data curation, formal analysis, validation. Justin Scott Neu: validation. Sung-Joo Kwon: validation. Yusuf Olanrewaju: validation. Aditya Balu: conceptualization, methodology. Writing – review and editing Goce Trajcevski: methodology, writing – review and editing Franky So: supervision. Wei You: supervision. Aram Amassian: conceptualization, supervision, writing – review and editing. Baskar Ganapathysubramanian: conceptualization, supervision, project administration, writing – review and editing.

Conflicts of interest

There are no conflicts to declare.

Data availability

The data and code for SEARS can be found at https://github.com/baskargroup/SEARS/tree/master and https://doi.org/10.5281/zenodo.16299279.

Acknowledgements

We acknowledge support from the Office of Naval Research (MURI award N00014-23-1-2001), and National Science Foundation (NSF DMR 2323716).

References

1 G. Tom, S. P. Schmid, S. G. Baird, Y. Cao, K. Darvish, H. Hao, S. Lo, S. Pablo-García, E. M. Rajaonson, M. Skreta, N. Yoshikawa, S. Corapi, G. D. Akkoc, F. Strieth-Kalthoff, M. Seifrid and A. Aspuru-Guzik, *Chem. Rev.*, 2024, **124**, 9633–9732.
2 D. P. Finegan, I. Squires, A. Dahari, S. Kench, K. L. Jungjohann and S. J. Cooper, *ACS Energy Lett.*, 2022, **7**, 4368–4378.
3 M. Scheffler, M. Aeschlimann, M. Albrecht, T. Bereau, H.-J. Bungartz, C. Felser, M. Greiner, A. Groß, C. T. Koch, K. Kremer, W. E. Nagel, M. Scheidgen, C. Wöll and C. Draxl, *Nature*, 2022, **604**, 635–642.
4 E. Norouzi, J. Waitelonis and H. Sack, The landscape of ontologies in materials science and engineering: A survey and evaluation, *arXiv*, 2024, preprint, 2408.06034, DOI: 10.48550/arXiv.2408.06034.
5 S. Yu, N. Ran and J. Liu, *Artif. Intell. Chem.*, 2024, **2**, 100076.
6 L. Himanen, A. Geurts, A. S. Foster and P. Rinke, *Advanced Science*, 2020, **7**, 1903667.
7 L. M. Ghiringhelli, C. Baldauf, T. Bereau, S. Brockhauser, C. Carbogno, J. Chamanara, S. Cozzini, S. Curtarolo, C. Draxl, S. Dwaraknath, Ä. Fekete, J. Kermode, C. T. Koch, M. Kühbach, A. N. Ladines, P. Lambrix, M.-O. Himmer, S. V. Levchenko, M. Oliveira, A. Michalchuk, R. E. Miller, B. Onat, P. Pavone, G. Pizzi, B. Regler, G.-M. Rignanese, J. Schaarschmidt, M. Scheidgen, A. Schneidewind, T. Sheveleva, C. Su, D. Usvyat, O. Valsson, C. Wöll and M. Scheffler, *Sci. Data*, 2023, **10**, 626.
8 C. Zuccarini, K. Ramachandran and D. D. Jayaseelan, *APL Mater.*, 2024, **12**, 090601.
9 A. Zakutayev, J. Perkins, M. Schwarting, R. White, K. Munch, W. Tumas, N. Wunder and C. Phillips, High Throughput Experimental Materials Database, *NREL Data Catalog*, 2017, https://htem.nrel.gov/, last updated: December 18, 2024.
10 C. Draxl and M. Scheffler, *J. Phys.: Mater.*, 2019, **2**, 036001.
11 K. Choudhary, K. F. Garrity, A. C. E. Reid, B. DeCost, A. J. Biacchi, A. R. Hight Walker, Z. Trautt, J. Hattrick-Simpers, A. G. Kusne, A. Centrone, A. Davydov, J. Jiang, R. Pachter, G. Cheon, E. Reed, A. Agrawal, X. Qian, V. Sharma, H. Zhuang, S. V. Kalinin, B. G. Sumpter, G. Pilania, P. Acar, S. Mandal, K. Haule, D. Vanderbilt, K. Rabe and F. Tavazza, *npj Comput. Mater.*, 2020, **6**, 173.
12 A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder and K. A. Persson, *APL Mater.*, 2013, **1**, 011002.
13 M. Scheffler, M. Aeschlimann, M. Albrecht, T. Bereau, H.-J. Bungartz, C. Felser, M. Greiner, A. Groß, C. T. Koch, K. Kremer, W. E. Nagel, M. Scheidgen, C. Wöll and C. Draxl, *Nature*, 2022, **604**, 635–642.
14 A. Williams and V. Tkachenko, *J. Comput. Aided Mol. Des.*, 2014, **28**, 1023–1030.
15 M. J. Statt, B. A. Rohr, D. Guevarra, S. K. Suram, T. E. Morrell and J. M. Gregoire, *Sci. Data*, 2023, **10**, 184.
16 M. Scheidgen, L. Himanen, A. N. Ladines, D. Sikter, M. Nakhaee, Á. Fekete, T. Chang, A. Golparvar, J. A. Márquez, S. Brockhauser, *et al.*, *J. Open Source Softw.*, 2023, **8**, 5388.
17 N. C. Röttcher, G. D. Akkoc, S. Finger, B. Fritsch, J. Möller, K. J. J. Mayrhofer and D. Dworschak, *J. Mater. Chem. A*, 2024, **12**, 3933–3942.
18 T. Ashino, *J. Data Sci.*, 2010, **9**, 54–61.
19 M. Borad, T. Chauhan and H. Mehta, *Int. J. Agric. Sci. Res.*, 2017, **6**, 26–28.
20 S. Scherzinger and S. Sidortschuck, *Conceptual Modeling*, Cham, 2020, pp. 441–455.
21 R. T. Fielding, *Architectural styles and the design of network-based software architectures*, 2000, https://ics.uci.edu/fielding/pubs/dissertation/rest_arch_style.htm.
22 T. Haerder and A. Reuter, *ACM Comput. Surv.*, 1983, **15**, 287–317.
23 N. El-Sayed, Z. Sun, K. Sun and R. Mayerhofer, *2021 29th International Symposium on Modeling, Analysis, and*

Simulation of Computer and Telecommunication Systems (MASCOTS), 2021, pp. 1–8.

24 P. Dwivedi, Kshamta and A. Joshi, *2022 International Conference on Cyber Resilience (ICCR)*, 2022, pp. 01–07.

25 J. Wu, L. Torresi, M. Hu, P. Reiser, J. Zhang, J. S. Rocha-Ortiz, L. Wang, Z. Xie, K. Zhang, B.-W. Park, *et al.*, *Science*, 2024, **386**, 1256–1264.

26 N. Baishnab, A. K. Mishra, O. Wodo and B. Ganapathysubramanian, *Comput. Mater. Sci.*, 2024, **244**, 113193.

27 T. Wu, S. Kheiri, R. J. Hickman, H. Tao, T. C. Wu, Z.-B. Yang, X. Ge, W. Zhang, M. Abolhasani, K. Liu, *et al.*, *Nat. Commun.*, 2025, **16**, 1473.

28 J. P. Mauthe, A. K. Mishra, A. Sarkar, B. Guo, G. J. Thapa, J. Schroedl, N. Luke, J. S. Neu, S.-J. Kwon, M. Chauhan, T. Wang, T. Trapier, H. Ade, D. Ginger, W. You, R. Ghosh, B. Ganapathysubramanian and A. Amassian, *Matter*, 2026, **9**, 102477.

29 A. K. Mishra, J. P. Mauthe, N. Luke, A. Amassian and B. Ganapathysubramanian, Interpretable Spectral Features Predict Conductivity in Self-Driving Doped Conjugated Polymer Labs, *arXiv*, 2025, preprint, 2509.21330, DOI: **10.48550/arXiv.2509.21330**.

30 S. Chinta and R. Rengaswamy, *Ind. Eng. Chem. Res.*, 2019, **58**, 3082–3092.

31 M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. Gonzalez-Beltran, A. J. Gray, P. Groth, C. Goble, J. S. Grethe, J. Heringa, P. A. 't Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S. J. Lusher, M. E. Martone, A. Mons, A. L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. van Schaik, S.-A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M. A. Swertz, M. Thompson, J. van der Lei, E. van Mulligen, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao and B. Mons, *Sci. Data*, 2016, **3**, 160018.

32 C. D. Services, Preparing FAIR data for reuse and reproducibility, **https://data.research.cornell.edu/data-management/sharing/fair/**.