Experiment Specification, Capture and Laboratory Automation Technology (ESCALATE): A software pipeline for automated chemical experimentation and data management

Ian M. Pendleton, Gary Cattabriga, Zhi Li, Mansoor Ani Najeeb,
Sorelle A. Friedler, Alexander J. Norquist, Emory Chan, Joshua Schrier

Supplementary Information

Project Aims Elaborated

We have developed ESCALATE to address specific needs related to our project goals. An ideal software platform should be general and reusable, providing an abstraction layer applicable to many different types of experiment types and chemical systems. The desired capabilities and challenges include:

(1) **A mechanism for specifying machine-readable experiment plans**, that enables "remote control" operation, e.g., over a web-based application programming interface (API). Most of the demonstrations described above are limited to modifying and optimizing a particular set of parameter values (e.g., reactant concentrations, pH), but in general, the system should be extensible to different types of experimental laboratory process workflows. Providing this layer of abstraction has a three-fold purpose. First, it unambiguously captures the intended experimental process description, facilitating reproduction studies. Second, it allows for domain experts to remotely operate high-throughput devices to test their hypotheses. Third, it facilitates participation by a broader range of scientists, e.g., theorists and computer scientists, who can use the remote operation to safely perform experiments that they would not otherwise be able to perform, and enables closed-loop control by computer algorithms.

(2) **Allow for hybrid human-robot laboratory operations.** The engineering efforts required to achieve complete automation are often expensive and time-consuming for exploratory scientific research. Inclusion of limited human intervention is typically the most expedient solution (e.g., preparing stock solutions, moving samples from one instrument to another). Therefore, in addition to converting the experiment plan into a set of equipment control files and capturing the outcomes of computer-monitored processes, it is also necessary to provide guidance to and collect observations from human operators. For example, a spreadsheet-based graphical user interface (GUI) that can be easily adapted for experimental workflow demands.

(3) **Comprehensive data and metadata capture.** As discussed above, a comprehensive description of an experiment includes a mixture of human observations and computer-generated output files. These data encompass a wide variety of formats such as scalar values of mass and concentration, time-series instrument data, spectra, photographic images, ambient environmental data. Each format has its own extraction-transformation-loading (ETL) requirements, and which may be associated with individual samples or batches of samples. An efficient way to handle this need is to create a semi-structured storage mechanism (e.g. data lake) for comprehensive data capture.

(4) **Facilitate machine learning on the captured data.** The semi-structured data captured in an experiment needs to be processed in order to make it useful for machine learning purposes. In addition to performing ETL on the experimental data, the inclusion of additional features describing the physicochemical properties of the chemicals and reagent mixtures is often desired. Finally, the data must be released in an easily manipulated format, with appropriate version control of the experimental process, data capture process, and data export process. A solution to this is the use of versioned repositories for both code and data to ensure backwards compatibility for all stages of the data pipeline.

Ontology

Categories - Elaborated

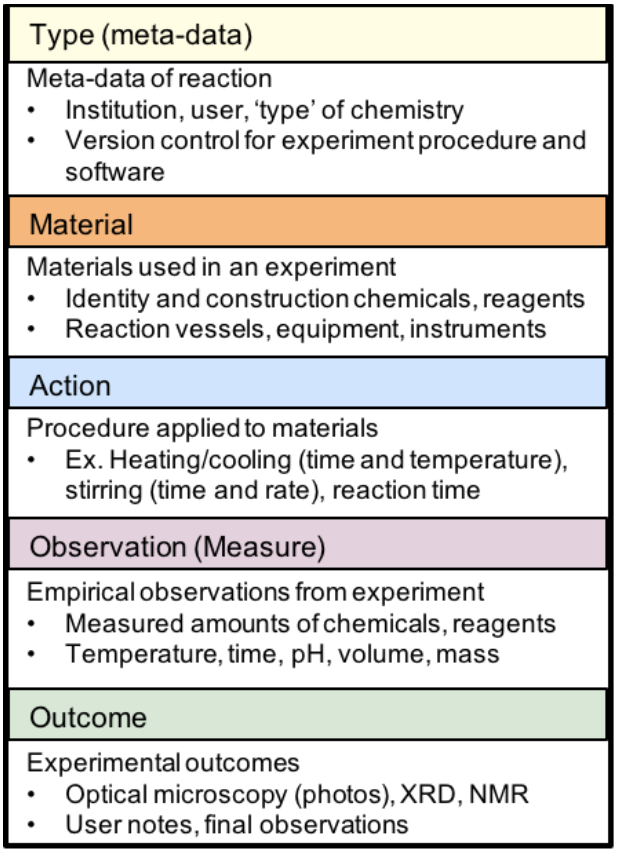

FIG. S1. Overview of categories for data sorting with examples from materials chemistry

The initial aim was to develop a systematic language to describe both the intention and outcome of a chemistry entity precisely and without ambiguity. An entity is the object of study in a chemical investigation such as an experiment, reagent, or chemical. Organizing the data into discrete categories aids initial design of data capture for chemists unfamiliar with curating datasets for statistical analysis. As such, the categories were developed to be easily interpreted by primary users, such as an experimental chemist or material scientist, as well as non-domain experts like computer scientists who might be participating in data analysis.

The '*type'* category describes general information about what the nature of the entity, who interacted with the entity, where the entity was created, as well as any associated entities and relevant relationships. Data grouped in the *type* category are used to record meta-data relating to an entity or to group similar entities together. The '*material'* category describes the physical components that are part of the entity, such as chemicals, reagents, equipment, and instruments. The '*action'* category describes operations performed by or to the entity. An '*observation'* consists of empirical data captured during the course of an experiment; this can

include both structured machine-generated data files as well as unstructured general comments from the human operator. An '*outcome'* consists of data that are the end result of an entity's lifecycle ***and*** are possible targets for machine learning models; isolating this subcategory from '*observation'* facilitates use by data scientists.

Entity Lifecycle - Elaborated

In the most general sense, any chemistry experiment can be fully characterized through a sufficient description of the lifecycle. We elected to use a specific example to describe relevant data capture for the complete lifecycle of a metal halide perovskite. The major aspects of a lifecycle are as follows: A general experimental lifecycle begins with a notional experimental *intent* that is used to guide an *actual* set of manipulations in the laboratory, which can result in the generation of a *particular* sample. To capture the distinction between the *intent* and the *actual* implementation, we describe an '*object'* as the sample prepared in the laboratory, e.g., this particular sample. A 'lifecycle' describes how the entity changes from a general class of descriptions to one or more unique objects through the course of typical laboratory operations.

Templates:

The lifecycle starts with a general '*template'*, which is a structured fill-in-the-blank form that provides a general pattern of information needed to specify an experiment and any relevant experimental limits.

Models:

Models may contain both user defined constraints as well as the inherent physical limitations of the chemistry. For instance, a model of a reagent solution must include the chemical contents, the nominal concentration of the reagents, and the final projected volume of the target solution. The model can also contain information obtained through previous experimentation such as empirical observation of solutions non-ideality.

Objects:

Executing the notional plan described by a *model* results in an *'object'* representing the particular physical laboratory instantiation. Each object is characterized by its particular type, material, action, observation, and outcome data, as well as the intended plan (model) and general experimental constraints (template). During the course of the experiment, an *object* acquires characteristics and interactions with other *objects*. As an example, all reagent objects have timestamps describing when a given reagent object was added to another.

Treating entity's in this fashion allows us to track the experimental intent, reproduce the particular experimental execution, and clarify the relationship between the nominal experiment and the empirical observations.

Lifecycle File Walkthrough

This section of the supporting information describes the contents and organization of the specific files captured in the current implementation of ESCALATE. The files discussed in this section have been renamed to better reflect the discussion in the main article. The file names would typically be prefixed with the associated UID of the plate.

The use of an UID ensures there are no duplicate entries in the final dataset and that the current iteration of the software/template can be successfully linked to all associated data. The creation of a UID naming scheme is left up to the convenience of the end user, but should be carefully selected before beginning data collection. For example, historical data entered from archived laboratory notebooks might use the combination of chemist's name, the notebook number and the notebook page to specify a unique instance of a particular experiment. New experiments specified through ESCALATE identify each run with a coordinated universal date-time label and a laboratory location identifier such that the location of the chemistry can be known. The lab location identifier also indicates specification from an alternative template as a difference in lab often corresponds to equipment changes. In general, a UID scheme should be developed which clearly differentiates unique entities for the experimental system of interest prior to data collection and reporting.

In the cases disclosed herein the prefix/UID was: 2018-11-09T01_43_18.720125+00_00_LBL (This and all other associated example files can be found in the 'Example ESCALATE Data.zip' github repo associate with: https://github.com/darkreactions/ESCALATE_report). The prefix has been removed for clarity.

Experiment Model

The experiment model used in the current implementation of ESCALATE has been designed for consumption by Hamilton NIMBUS automated liquid handling robots. The format of the experiment model is demonstrated in the supplementary information file: **Experiment-Model_RobotInput.xls**. This file dictates the dispensing volumes of various reagents, the identity of those reagents, the associated well (and thereby the experiment) to which each reagent is dispensed, plate level conditions which affect all of the associated experiments such as temperature, shake rate and duration of the reaction, along with the equipment used for operation of the robot. This file has also been used for manual dispensing with success. The user merely inputs the nominal (target) values for the reagent dispensing. In either case, robotic manipulation or manual dispensing, ESCALATE generates the final report through the same pipeline.

Reagent Model-Object

The reagent model-object interface is a key element in acquiring operator information about reagent preparation. In the main text, a simplified version of the form is presented for clarity. Fig. S2 demonstrates a full example of what an operator would see during use of ESCALATE. The form shown is also available as Reagent-Model-Object_Interface1_Entry.xlsx file and Reagent-Model-Object_Interface1_Entry.csv file.

	Run Data	Reagent Preparation Information					
Date Created	2018-11-09	Reagent	Temp (C)	Stir (RPM)	Duration (seconds)	Creation Date	Operator
Time Created _UTC	01_43_18	1				2018-11-09	Zhi Li
Laboratory	LBL	2	75	450	3600	2018-11-09	Zhi Li
Operator Name	Zhi Li	3	75	450	3600	2018-11-09	Zhi Li
Job Serial No.	2018-11-09T01_43_18.720125+00_00_LBL	4					
Exp Workflow Ver	1.10	5				2018-11-09	Zhi Li
Generator Workflow Ver	2.00	6				2018-11-09	Zhi Li
Challenge Problem	1.00	7				2018-11-09	Zhi Li
Note 1	null						
Note 2	null						
Experimental Summary:	null						

	Chemical Abbreviation (In order of addition)	Nominal Amount (target for robot deck or ideal behavior)	Actual Amount (total on robot deck or measured actual)	Unit	Nominal Concentration (M)	Actual Concentration (M)	Actual Reagent Temperature (Celsius - Immediately Prior to Robot Initiation)
Reagent1	Final Volume =	8.5	20.00				45
Chemical1	GBL	8.51	20	milliliter			
Chemical2							
Chemical3							
Reagent2	Final Volume =	15.8	22.00				45
Chemical1	PbI2	10.90	10.908	gram	1.50	1.48	
Chemical2	EtNH3I	8.18	8.19	gram	3.00	2.96	
Chemical3	GBL	15.76	16.00	milliliter			
Reagent3	Final Volume =	8.7	14.00				45
Chemical1	EtNH3I	9.06	9.064	gram	6.00	5.82	
Chemical2	GBL	8.73	9.00	milliliter			
Chemical3							
Reagent4	Final Volume =						
Chemical1							
Chemical2							
Chemical3							
Reagent5	Final Volume =	9.5	10.00				22
Chemical1	FAH	9.45	10	milliliter			
Chemical2							
Chemical3							
Reagent6	Final Volume =	6.1	10.00				22
Chemical1	FAH	6.13	10.00	milliliter			
Chemical2							
Chemical3							
Reagent7	Final Volume =		NA				
Chemical1							
Chemical2							
Chemical3							

FIG. S2. Full overview of the reagent interfacing form used for storing reagent model information and capturing operator input regarding the reagent object.

After completing the run, the data stored in this table are processed into a JSON format. An example of the output file from processing of the reagent-model-object interface is included as Reagent-Model-Object_Interface1_JSONRender.tsv. Note that in the current implementation of ESCALATE the reagent-model-object interface is designed for use by the operator as a web interface through google sheets.

Experiment Object

Significant data is captured regarding the experimental objects. Both the operator and the robot record information that is parsed by ESCALATE and used in the generation of the final report. A simplified example of the primary experimental object is shown in Table SI.

Table SI. Sample experimental object file for capturing observations and outcomes from a set of experiments including: material (ex. vial site), observations (reaction temperature), outcomes (outcome measured), and other type data (notes)

Vial Site	Empirical Reaction Temperature (C)	Outcome Measured	notes
A1	70	2	PbI_2 Precipitation
C1	70	3	Crystals on side of vial
E1	70	1	PbI_2 Precipitation
G1	70	4	Crystals on side of vial
		⋮	
H12	70	2	PbI_2 Precipitation

Log files from robotic output are parsed separately, but need to be correctly attributed to the appropriate experiment. This entails that the operator responsible for acquiring observational data has a means of storing data, accessing URLs or other pointers to the data, and correctly referencing those stored data in a useable form for post processing. A full example of the experimental output file can be found in the zip file included in the SI of this publication labeled **Experimental-Object_CrystalScoring.csv.** In the current implementation of ESCALATE the crystal scoring sheet is generated at the same time as the rest of the experimental models (during the generate step of the pipeline). The form is hosted on google drive as a google sheet where the operator can input the relevant workup information during the '*observe'* step of ESCALATE. In the provided example, the completed google sheet document (what amounts to a web hosted spreadsheet) has been exported as a CSV and as an XLSX. Note that the data collected in this step includes information about a particular experimental '*objects*' instantiation. Data include: bulk temperature analysis, crystal score, model name (for collaborators to identify the algorithms that they are using), predicted out (collaborators to identify model prediction), participant, and notes. The collaborators interact with the pipeline through the submission script (Experiment-Model_collaborator_submission_template.csv in the supplementary information). The submission template is parsed by ESCALATE which automatically completes the aforementioned participant information in the Experimental-Object_CrystalScoring form. Together with the Reagent-model-object interface, these data provide the relevant information to fully define an experimental object.

Other data is also associated with each run such as the **images** and **thermal_images** files which aim to record the visual state and temperature data of each well (and thereby each experiment) on a plate. Examples of these files are included as zipped folders in the supplementary information. All of the examples are associated with the denoted experiment and can be referenced against the Experimental-Object crystalscoring form (i.e. one can compare crystal score to the relevant image to better understand the outcome characterization).

Additional Captured Files

The following sections details ESCALATE **captured** files that do not yet contribute to the final **report**, but are captured during each run for later processing and development. Note that this is one of the strengths of the data lake approach: despite not having a complete framework for processing data and incorporating into a database (a more typical approach for machine learning applications), we are able to capture information that may be relevant to future analysis. The following list is a comprehensive overview of remaining portions of the ESCALATE's data capture.

Experiment-Object_Temperature_Humidity.txt outlines the ambient temperature data recorded for the laboratory for a given experiment. We intend to develop a processing pipeline for this data in the near future.

Experimental-Model_nominalMolarity.csv details the nominal concentration of each chemical in the final experimental model. This concentration is based on ideal solution mixing behavior and is not used in to calculate values in the final report.

Experimental-Model_mmolbreakout.csv is a fully elaborated nominal experiment plan which details the mmol of each chemical in each reagent that is targeted for dispensing by the nimbus robot. This file provides a high level of detail regarding the nominal experiment plan. This file is created as a result of the generate step of ESCALATE.

Experiment-Template_ESCALATELogFile.txt is an ESCALATE log file that records data associated with user defined template parameters. These data are generated from the Experiment-Template_Runme.py executable and track the operation of ESCALATE during run generation. Reagent information is included where relevant to capture the relationship between the experimental template, and the reagent model constructed for this particular experiment. This file is not used in the final report, but is designed to capture additional data regarding the intention of the experiment. We aim to parse this file for additional data capture in the near future.

Experiment-Object_ActivityDocument.med details the actions of the NIMBUS liquid handler, including information about hardware, dispensing equipment and general information related to typical equipment operation. These data are unused by the current version of ESCALATE as they do not capture variance in the systems of interest.

Experiment-Object__RoboticLayout.lay describes the specific configuration data relating to the NIMBUS liquid handler. All aspects of the current robotic equipment are included in this file. Unparsed, but potentially useful when comparing to other robotic environments or laboratories. These data are not parsed by the current version of ESCALATE.

Reports Detailed

Data Model Design

The ESCALATE pipeline generates data that can be organized into two general types: structured and unstructured. Structured data can be characterized by being classifiable and consistent. For example, the ontology entities experimental template, model or object; each composed of specified data elements (see Figure S3). Unstructured data are data not easily classified or predictable, for example images or robot log files, that can come in a variety of formats and structures.

Typically, there is an upfront cost (effort and resources) to structured [disambiguated] data. Yet for this pipeline the structured data is a by-product of a pre-defined ontology, resulting in data element, structure and element relationship specifications. Overall, this disambiguation delivers value to the data collection, storage and reporting process. The value of the structured (disambiguated) data can be summarized as providing:

- consistent, valid and relatable data from the pipeline,
- that can be automated,
- improving efficiencies in:
    - data retrieval, and
    - comprehension, analysis, and reporting
- containers for flexible data elements (e.g. metadata)

Examples of ESCALATE pipeline structured data are:

- MATERIAL container: data describing material type, quantity (value), and [meta-data] description
    - Duration requires the date and time are captured, point to log files in the additional documentation
    - URL mapping to the files that describe the operation of instruments
- OUTCOME container: data representing the outcome type, value, and units

The ESCALATE pipeline also creates unstructured data. Though the structure of this data may not be known or defined, it can still provide value. Current technology provides unstructured data to be easily stored, as a document in its original form, on a variety of platforms, as a

'document'. All that is needed is to provide a data link or reference (e.g. URL) from a structured data container (e.g. Actions) to the document(s).

Examples of ESCALATE pipeline unstructured data are:

- ACTION container: URL mapping to the files that describe the operation of instruments (log files)
- OBSERVATION container: Pointers to file URLs or naming scheme for image (.jpg) files

Regardless how the data is represented, structured or unstructured, the Report form remains the same; in the current implementation, a 2D data frame representation. In the future, more complex data structures could be implemented to provide higher efficiencies in reporting or analysis (e.g. the highly normalized relational structure in DRP [1]. Or the data could be stored in a technology better suited for a structured / unstructured information (e.g. MongoDB as a data lake) as is being pursued in the SD2 Collaboration.

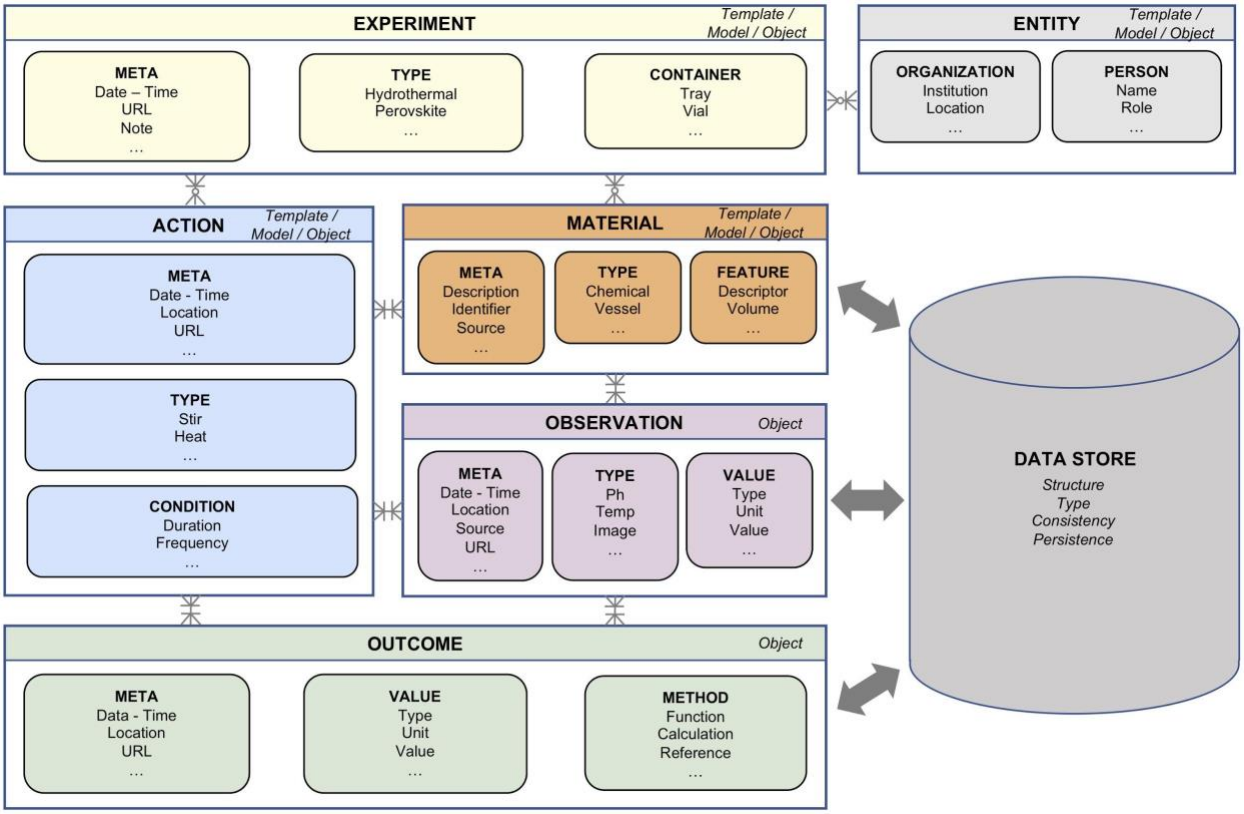

FIG S3. Logic Data Model - Overview of the relationships between entities in the final data model

Chemical Tracking

Chemical and Reagent UIDs / Indexes

We have included examples of the chemical tracking with this supplementary information: ChemicalTracking_Inventory.csv

Chemicals and reagents Reagents and chemicals have UIDs serve as a compliment to the experiment UID. In general, all templates should have a UID to act as a pointer for other entities. We further specify reagent and chemical UIDs in this section to elaborate on the number of options available for storing data related to these entities. We have found that unique identifiers linked to an external data storage mechanism is sufficient for describing reagents and chemicals. that are resolved locally. For example, a URL accessible tabulated data entry form (such as those provided by Google Sheets) can store the list of the chemical components of a given reagent model along with an associated unique identifier. In our ontology, 'chemicals' describe pure compounds. We use InChIKeys; other identifiers (e.g., SMILES, InChI strings, CAS numbers, PubChem ID) can be used by ESCALATE, but in practice these have problems with canonical order and copyright.

The most fundamental description of reagents is that which references a set of instructions from only chemicals. Chemicals are entities which have unique chemical reference labels or are "pure" compounds. For most purposes chemicals can be treated as entities which can be commercially obtained. A number of unique naming schemes exist for describing chemicals including SMILES, InChI, InChIKey. Note that the commonly used system of CAS numbers, often referenced in commercial catalogs, are not UIDs and should be avoided. While a detailed inventory tracking system would provide further information about chemical provenance, but adds too much complexity to the composition of chemicals, simplicity in the early stages of data capture is desirable. A set of chemicals along with preparation instructions constructed as a set of lists is sufficient for describing a reagent. Such an implementation avoids the complexity of integrating more complicated data structures into the definition of reagents early in development.

State Space Examples

The file size for the state space varies as a function of the grid density. At the limit of the robotic precision, the total file size can be as large as 8tb. The reduced density grid used in the most recent iterations of the DARPA-SD2 challenge problem have been approximately 500mb in size. Due to these constraints we elected not to include a demonstration of the state space. However, this can be provided upon request.

Technology Stack Elaborated

Current Implementation

ESCALATE is written in Python 3.6 using Google Apps for Education to provide persistent shared storage and graphical interfaces. The data lake (i.e. the storage system for each plate of experiments) is organized as a set of directories on Google Drive. A complete example of a directory is included with this supporting information. The naming scheme for the example has altered to increase clarity and consistency with the language used in this article. The recording of the experimental template is currently instantiated through a Python 3.6 executable. The experimental template file along with the ESCALATE generated experimental model (a .XLS formatted file) are uploaded to the data lake via the Google Drive API. Google Sheets was used as the spreadsheet interface used to capture reagent data from the operator (i.e. the reagent *model* and reagent *particular/object* file) and rendered to the data lake as a JSON type file after the experiment is complete. Chemical and reagent identifiers are stored as API accessible Google Sheets. The final report is generated by accessing the data lake through the Google Drive API to generate "_raw_", "_rxn_", and "_out_" followed by concatenating the physicochemical data access through Chemaxon for the additional "_feat_" and "_calc_" columns. Though these technologies use a Google Drive environment, the open framework of the code allows for alternative databases, interfaces, and formats that what have been demonstrated here. For example, work is underway to migrate the final report to a version controlled data repository as well a migrating chemical and reagent UID linked information to a Synbiohub hosted database. An up to date version of the ESCALATE 'capture' and 'report' codes can be found on github at the following links:
https://github.com/darkreactions/ESCALATE_Capture
https://github.com/darkreactions/ESCALATE_report

References

1 Dark Reactions Project, https://darkreactions.haverford.edu, (accessed 15 January 2019).